

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky

System pro interní firemní komunikaci

Bc. Tomáš Langer

Diplomová práce

2020

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2019/2020

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

|                   |  |
|-------------------|--|
| Jméno a příjmení: | <b>Bc. Tomáš Langer</b>                      |
| Osobní číslo:     | <b>I17209</b>                                |
| Studijní program: | <b>N2646 Informační technologie</b>          |
| Studijní obor:    | <b>Informační technologie</b>                |
| Téma práce:       | <b>Systém pro interní firemní komunikaci</b> |
| Zadávací katedra: | <b>Katedra softwarových technologií</b>      |

### Zásady pro vypracování

Cílem této práce je návrh systému a vytvoření funkční aplikace, která bude umožňovat interní firemní komunikaci. Aplikace bude umožňovat tyto funkcionality: registrace uživatelů a přístup do systému podle jejich práv, evidence vnitropodnikových příkazů, směrnic a dalších dokumentů, zakládání a správa diskusních vláken, evidence firemních akcí, porad a pracovních úkolů včetně termínů a zodpovědných pracovníků, generování sestav dle zadaných kritérií pro potřeby zaměstnanců a komunikace s řešiteli a uživateli systému prostřednictvím mailů, SMS zpráv a dalších komunikačních nástrojů.

Práce bude obsahovat: rešerši systémů, které se zabývají touto problematikou, analýzu procesů, které budou znázorněny v procesním BPMN diagramu s cílem dosažení maximální bezpečnosti aplikace, popis použitých technologií, návrh databáze a ER diagram s využitím „Crow's Foot“ notace entity-relationship.

Pro vytvoření aplikace bude využita databáze MySQL nebo Oracle a objektově orientovaný programovací jazyk Java s možností využití technologií pro podporu tvorby aplikací (např. JavaServer Pages (JSP)).

Rozsah pracovní zprávy: **50-60 stran**  
Rozsah grafických prací: **-**  
Forma zpracování diplomové práce: **tištěná**

Seznam doporučené literatury:

ŘEPA, Václav. Analýza a návrh informačních systémů. 1. vyd. Praha : Ekopress, 1999. 403 s. ISBN 80-86119-13-0.  
GROFF, James R. a Paul N. WEINBERG. SQL kompletní průvodce. Brno: Computer Press a.s., 2005. 936 s. ISBN 80-251-0369-2.  
BRYLA, Bob a Kevin LONEY. Mistrovství v Oracle Database 10g. Brno: Computer Press a.s., 2006. 700 s. ISBN 80-251-1277-2.  
SCHILD, Herbert. Mistrovství Java – Kompletní průvodce vývojáře. Brno: Computer Press a.s., 2014. 1224 s. EAN: 9788025141458.

Vedoucí diplomové práce: **Ing. Miloslav Macháček, Ph.D.**  
Katedra informačních technologií

Datum zadání diplomové práce: **5. listopadu 2019**  
Termín odevzdání diplomové práce: **15. května 2020**



---

**Ing. Zdeněk Němec, Ph.D.**  
děkan

---

**prof. Ing. Antonín Kavička, Ph.D.**  
vedoucí katedry

V Pardubicích dne 15. listopadu 2019

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na mou práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, a zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 15. 5. 2020

Bc. Tomáš Langer

# PODĚKOVÁNÍ

Rád bych poděkoval Ing. Miloslavu Macháčkovi, Ph.D., za jeho rady a připomínky, které mi pomohly při zpracování mé diplomové práce. Velkou měrou bych rád poděkoval své rodině, která mě podporovala a byla mi oporou po celou dobu mého studia.

## **ANOTACE**

Diplomová práce byla zaměřena na interní komunikační procesy malé firmy a jejím cílem bylo vypracovat systém pro zlepšení interní komunikace.

## **KLÍČOVÁ SLOVA**

Interní komunikační systém, JPS

## **TITLE**

System for internal company communication

## **ANNOTATION**

Dissertation focused on internal communication processes in small companies.

## **KEYWORDS**

Internal communication system, JSP

# OBSAH

|  |           |
|--|-----------|
| <b>Seznam obrázků.....</b>                                       | <b>10</b> |
| <b>Seznam tabulek .....</b>                                      | <b>11</b> |
| <b>Seznam zkratk .....</b>                                       | <b>12</b> |
| <b>Úvod .....</b>  | <b>14</b> |
| <b>1 Analýza komunikačních softwarů.....</b>                     | <b>15</b> |
| 1.1 Rešerše.....   | 15        |
| 1.1.1 Jive.....  | 15        |
| 1.1.2 Deskalerts.....  | 16        |
| 1.1.3 SnapComms .....  | 18        |
| 1.1.4 Jira.....  | 20        |
| 1.1.5 WordPress.....   | 21        |
| 1.1.6 IBM Notes/Domino .....                                     | 22        |
| 1.2 Vyhodnocení systémů:.....                                    | 22        |
| 1.3 Komunikační systém.....                                      | 23        |
| <b>2 Technologie a Nástroje použité pro tvorbu aplikace.....</b> | <b>24</b> |
| 2.1 UML.....   | 24        |
| 2.2 UP .....   | 24        |
| 2.2.1 Požadavky .....  | 25        |
| 2.2.2 Analýza .....  | 26        |
| 2.2.3 Návrh .....  | 29        |
| 2.2.4 Implementace.....  | 30        |
| 2.2.5 Testování.....   | 30        |
| 2.3 Enterprise Architect .....                                   | 30        |
| 2.4 Netbeans.....  | 30        |
| 2.5 ArisExpres .....   | 31        |
| 2.6 Java .....   | 31        |
| 2.7 HTML .....   | 31        |
| 2.8 JSP .....  | 31        |

|          |                                       |           |
|----------|---------------------------------------|-----------|
| 2.9      | Historie JSP .....                    | 32        |
| 2.10     | Architektura JSP .....                | 33        |
| 2.10.1   | JSP Direktivy .....                   | 34        |
| 2.10.2   | Sevlet .....                          | 36        |
| 2.10.3   | web.xml.....                          | 36        |
| 2.10.4   | doGet.....                            | 37        |
| 2.10.5   | doPost.....                           | 37        |
| 2.11     | CSS .....                             | 38        |
| 2.12     | Regulární výrazy .....                | 38        |
| 2.13     | Apache Tomcat .....                   | 39        |
| <b>3</b> | <b>Datový model.....</b>              | <b>40</b> |
| <b>4</b> | <b>Uživatelská příručka.....</b>      | <b>53</b> |
| 4.1      | Šifrování hesla .....                 | 53        |
| 4.2      | Část pro registraci a přihlášení..... | 53        |
| 4.2.1    | Přihlášení .....                      | 53        |
| 4.2.2    | Registrace.....                       | 53        |
| 4.3      | Úvodní stránka.....                   | 54        |
| 4.4      | Stránka účtů .....                    | 55        |
| 4.5      | Stránka událostí.....                 | 55        |
| 4.6      | Stránka dokumenty .....               | 56        |
| 4.7      | Stránka Taskboard .....               | 57        |
| 4.7.1    | Stránka přidání úkolu.....            | 58        |
| 4.8      | Stránka kalendář .....                | 58        |
| 4.9      | Část komunikace.....                  | 59        |
| 4.9.1    | Mail .....                            | 59        |
| 4.9.2    | SMS .....                             | 59        |
| 4.9.3    | Chat.....                             | 59        |
| 4.10     | Část účet.....                        | 60        |
| 4.10.1   | Stránka podrobnosti .....             | 60        |
| 4.10.2   | Stránka změna údajů.....              | 60        |



|          |  |           |
|----------|--|-----------|
| 4.10.3   | Stránka Kvíz .....                             | 61        |
| <b>5</b> | <b>Programová příručka .....</b>               | <b>62</b> |
| 5.1      | Instalační příručka .....                      | 62        |
| 5.2      | Ukázka zdrojového kódu .....                   | 65        |
| <b>6</b> | <b>Externí knihovny A TEMPLATY .....</b>       | <b>66</b> |
| 6.1      | mail.jar .....                                 | 66        |
| 6.2      | activation.jar .....                           | 66        |
| 6.3      | xssflt.jar .....                               | 66        |
| 6.4      | Twilio-7.9.1 -jar -with-dependencies.jar ..... | 66        |
| 6.5      | mysql-connector-java-8.0.16.jar .....          | 66        |
| 6.6      | Externí template .....                         | 67        |
|          | <b>Závěr .....</b>                             | <b>68</b> |
|          | <b>Použitá literatura .....</b>                | <b>69</b> |
|          | <b>Přílohy .....</b>                           | <b>70</b> |

## SEZNAM OBRÁZKŮ

|   |    |
|---|----|
| Obrázek 1 Síťový diagram aplikace Deskalerts .....  | 16 |
| Obrázek 2 ACL aplikace Deskalerts .....             | 17 |
| Obrázek 3 SnapComms rozeslání zprávy .....          | 19 |
| Obrázek 4 Kroky návrhu software .....               | 25 |
| Obrázek 5 Ukázka Use Case diagramu .....            | 27 |
| Obrázek 6 Ukázka komunikačního diagramu .....       | 28 |
| Obrázek 7 Ukázka sekvenčního diagramu .....         | 28 |
| Obrázek 8 Ukázka BPMN diagramu .....                | 29 |
| Obrázek 9 Architektura JSP .....                    | 33 |
| Obrázek 10 Datový model část 1 .....                | 40 |
| Obrázek 11 Datový model část 2 .....                | 41 |
| Obrázek 12 Úvodní stránka .....                     | 55 |
| Obrázek 13 Stránka dokumentů .....                  | 56 |
| Obrázek 14 Ukázka úkolů .....                       | 58 |
| Obrázek 15 Seznam chatovacích místností .....       | 59 |
| Obrázek 16 Chatovací místnost .....                 | 60 |
| Obrázek 17 Stránka kvízů .....                      | 61 |
| Obrázek 18 Nastavení zabezpečení mailu krok 1 ..... | 63 |
| Obrázek 19 Nastavení zabezpečení mailu krok 2 ..... | 64 |
| Obrázek 20 Nastavení zabezpečení mailu krok 3 ..... | 64 |

## SEZNAM TABULEK

|  |    |
|--|----|
| Tabulka 1 Uživatelé .....                                      | 42 |
| Tabulka 2 Role .....   | 43 |
| Tabulka 3 Taskboard.....                                       | 43 |
| Tabulka 4 Stav tasku .....                                     | 44 |
| Tabulka 5 Taskboard uživatelů .....                            | 45 |
| Tabulka 6 Vysvětlivky vztahů v tabulce uzivatel_taskboard..... | 46 |
| Tabulka 7 Kalendář .....                                       | 46 |
| Tabulka 8 Kalendářní události uživatelů.....                   | 47 |
| Tabulka 9 Události v systému .....                             | 47 |
| Tabulka 10 Datastore .....                                     | 48 |
| Tabulka 11 Chatovací místnosti.....                            | 49 |
| Tabulka 12 Stav chatovacích místností.....                     | 50 |
| Tabulka 13 Chatové zprávy .....                                | 50 |
| Tabulka 14 Archivované zprávy .....                            | 51 |
| Tabulka 15 Tabulka kvízů.....                                  | 51 |
| Tabulka 16 Tabulka odpovědí na kvízové otázky.....             | 52 |
| Tabulka 17 Stav odpovědi.....                                  | 52 |
| Tabulka 18 Konkrétní hodnoty pro tabulku rolí .....            | 62 |

## SEZNAM ZKRATEK

|       |                                    |
|-------|------------------------------------|
| JSP   | Java Server Pages                  |
| XML   | eXtensible Markup Language         |
| HTML  | Hypertext Markup Language          |
| CSS   | Cascading Style Sheets             |
| URL   | Uniform Resource Locator           |
| SQL   | Structured Query Language          |
| PK    | Primary Key                        |
| FK    | Foreign Key                        |
| IDE   | Integrated development environment |
| JAR   | Java Archive                       |
| UML   | Unified Modeling Language          |
| UP    | Unified Process                    |
| Regex | Regular expression                 |
| Tag   | štítek, značka                     |
| SMTP  | Simple Mail Transfer Protocol      |
| IMAP  | Internet Message Access Protocol   |

|      |                                     |
|------|-------------------------------------|
| POP  | Post Office Protocol                |
| BPMN | Business Process Model and Notation |
| SMS  | Short Message Service               |
| API  | Application Programming Interface   |
| ACL  | Access Control List                 |
| IP   | Internet Protocol.                  |

# ÚVOD

V dnešní době potřebují velké i malé podniky snadný, rychlý a efektivní přístup k informacím všeho druhu a tyto informace dále použít, ke zlepšování svých služeb, financí i pracovního prostředí. Pro přístup k datům nám v dnešní době v pracovním prostředí slouží především notebooky. Jsou relativně malé, rychlé a připojení k internetu je dnes dostupné takřka odkudkoliv. V notebookech mohou být nainstalovány různé aplikace pro usnadnění přístupu k datům nebo jejich zprostředkování ostatním. Proto jsou vhodné jako médium pro zprostředkování informací.

Notebooky a podobná zařízení pokrývají řekněme hardwarovou část problému. Ale k získání a využití informací nestačí jen hardware je nutný i software. A cílem této diplomové práce je vytvořit takovou aplikaci, která by usnadňovala přístup k datům a komunikaci v rámci menší firmy. V každém oboru dnes platí, že čas jsou peníze. Uvedu příklad: v době, kdy byla veškerá data na papírech uložených v kartotékách, někdo musel jít a fyzicky vyhledat příslušná data, ať se jednalo o zápis z porady, kalendář akcí z minulého roku, seznam obchodních partnerů atd... Dnes je tato práce značně ulehčena díky komputelizaci. V podstatě se nemusíte hnout z místa a počítač za vás může prohledat celou databázi. Tímto se šetří čas, a tedy i peníze. A právě takového usnadnění bych chtěl dosáhnout pomocí své aplikace.

Návrh a analýza aplikace byla provedena v aplikaci Enterprise Architect a BPMN diagram byl vytvořen pomocí nástroje ARIS Express. Webová aplikace byla vytvořena v IDE NetBeans za použití technologie JSP (Java + HTML) a CSS. V aplikaci bylo využito externích knihoven a template. Pro uchovávání a řízení dat byl zvolen databázový systém MySQL.

# 1 ANALÝZA KOMUNIKAČNÍCH SOFTWAREŮ

## 1.1 Rešerše

V této části se budu zabývat jak aplikacemi pro firemní komunikaci, tak i aplikacemi, které mají podobné nebo jinak užitečné vlastnosti pro firemní komunikaci, ale nejsou určeny konkrétně pro tento účel. Příkladem budou softwary Jira a WordPress, které jako takové nejsou určeny přímo pro komunikaci, ale pro organizaci týmu nebo publikování článků. Jelikož podle mého názoru tyto systémy obsahují užitečné funkce, které by měla mít dobrá aplikace pro interní firemní komunikaci, rozhodl jsem se je zahrnout do této části, konkrétní vlastnosti budou popsány dále v textu.

### 1.1.1 Jive

Jive je označení pro platformu sociálních sítí sloužící pro obchodní aplikace. Jive je označován za „předního poskytovatele moderních komunikačních a kooperačních řešení pro podnikání“.

Jive nabízí více verzí svých produktů, každá verze je určena pro různé potřeby. V závislosti na velikosti a způsobu použití se Jive dělí následovně:

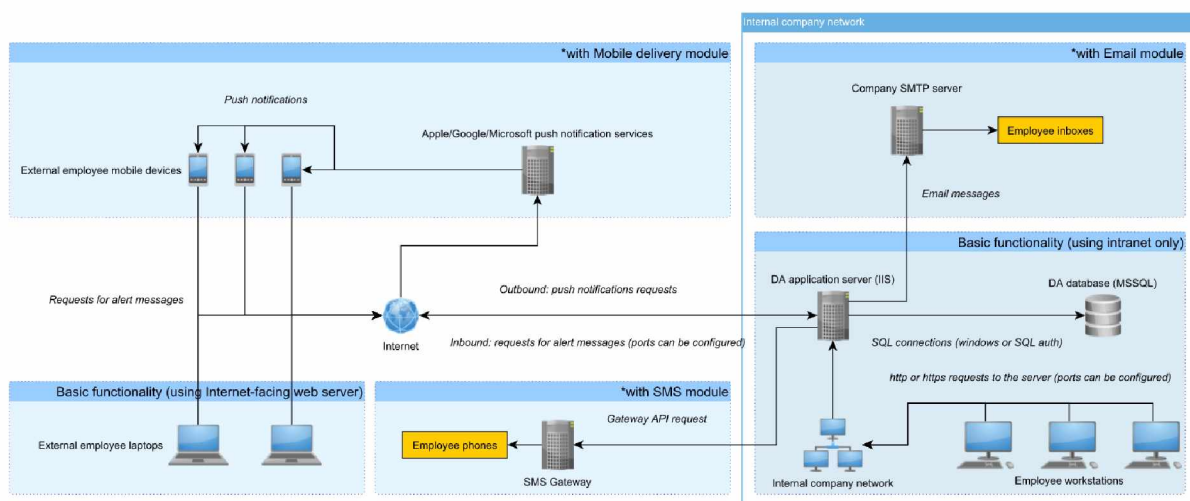
- Jive-n: je základní platformou Jive softwaru pro zaměstnance, slouží k tomu, aby se mohli zaměstnanci navzájem propojit na zabezpečené síti.
- Jive-x: je software pro kolaboraci tedy pro spolupráci zúčastněných osob, umožňuje obchodním partnerům a zákazníkům komunikovat se svými protějšky v ostatních firmách.
- Jive Daily: je mobilní aplikace, která zprostředkovává přístup k Jive pomocí mobilního zařízení a tím pomáhá zaměstnancům zůstat v obraze ohledně důležitých akcí a novinek.

Jak již bylo řečeno Jive má spoustu funkcí, které teď proberu podrobněji. Pro naléhavé zprávy obsahuje software Jive systém pro naléhavá oznámení. Každé takové naléhavé oznámení se zobrazí ve formě pop-up zprávy. Dále jsou veškeré zprávy z různých platform shromažďovány na jednom místě. Poskytuje také možnost nahrávání videí, textů, zvukových a dalších různých typů souborů. K takto vloženým souborům umožňuje ostatním uživatelům vložit komentáře. Díky této funkci mohou uživatelé sdílet své názory a návrhy k danému souboru.

Další z funkcí je tzv. kolaborační systém, tedy systém, který umožňuje efektivní spolupráci zaměstnanců. Pro tento účel Jive poskytuje různé nástroje, kterými lze projekt spravovat. Jsou jimi například taskboard, jehož prostřednictvím lze zadávat úkoly. Dále nástroj pro sdílení dokumentů, plánování porad a jiných akcí.

### 1.1.2 Deskalerts

DeskAlerts je webový výstražný (upozorňující) systém. Umožňuje odesílat upozornění vybraným příjemcům v různých formách. Například pomocí vyskakovacího okna na ploše, tapety, spořiče obrazovky, e-mailu nebo SMS zprávy. U odeslaného upozornění je možné sledovat, zda zaměstnanec zprávu obdržel či nikoli.



Obrázek 1 Síťový diagram aplikace Deskalerts

Aplikace DeskAlerts je složena ze dvou hlavních částí:

- webového serveru aplikace
- a klientské aplikace.



Webový server běží na platformě Windows a svá data ukládá do databáze MSSQL. Databáze může být umístěna na stejném serveru jako webová aplikace anebo může být umístěna na jiný databázový server. Klientská aplikace je instalována lokálně na počítačích příjemců zpráv.

Zde uvádím obrázek, který zobrazuje konfigurovatelnost celé aplikace Deskalerts. Na něm jsou naznačena práva pro různé funkce aplikace. Práva se dělí do šesti skupin, a to na právo vytvoření, úpravy, smazání, odeslání, prohlížení a zrušení. Pro běžného uživatele jsou podstatnější funkce, jako je například vytváření upozornění, rozesílání mailů, vytváření skupin uživatelů atd... Pro pokročilejší uživatele nebo pro administrátory aplikace nabízí pokročilejší funkce jako je například funkce IP group. Tato funkce je zajímavá z toho pohledu, že v podstatě připomíná klasický email. Ale místo emailových adres se uvádí rozsah IP adres. Takže, zpráva není určena přímo pro osobu ale spíše pro konkrétní zařízení.

|                                     |                   | Create                              | Edit                     | Delete                   | Send                                | View                                | Stop                     |
|-------------------------------------|-------------------|-------------------------------------|--------------------------|--------------------------|-------------------------------------|-------------------------------------|--------------------------|
| <input type="checkbox"/>            | Desktop alerts    | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/>            | Campaign          | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |
| <input type="checkbox"/>            | E-mails           |                                     |                          |                          | <input type="checkbox"/>            |                                     |                          |
| <input type="checkbox"/>            | Surveys           | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |
| <input type="checkbox"/>            | Users             | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> |                                     |                                     |                          |
| <input type="checkbox"/>            | Groups            | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> |                                     | <input type="checkbox"/>            |                          |
| <input type="checkbox"/>            | Message Templates | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |                          |
| <input checked="" type="checkbox"/> | Reports           |                                     |                          |                          |                                     | <input checked="" type="checkbox"/> |                          |
| <input type="checkbox"/>            | IP groups         | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            |                          |
| <input type="checkbox"/>            | RSS/news feed     | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |
| <input type="checkbox"/>            | Screensavers      | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |
| <input type="checkbox"/>            | Wallpapers        | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input type="checkbox"/> |
| <input type="checkbox"/>            | Emergency Alerts  | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            |                          |
| <input type="checkbox"/>            | Color codes       | <input type="checkbox"/>            | <input type="checkbox"/> | <input type="checkbox"/> |                                     | <input type="checkbox"/>            |                          |

Obrázek 2 ACL aplikace Deskalerts

### 1.1.3 SnapComms

SnapComms je software určený pro včasné upozornění zaměstnanců na důležité zprávy. Zprávy v SnapComms je možné personalizovat a zvolit různé způsoby zobrazení. Zprávy se mohou zobrazit jako upozornění ve vyskakovacím okně, tapetě pracovní plochy, spořiči obrazovky, ve formě videa, nebo dokonce i kvízu či zpravodaje. Za zmínku stojí hlavně možnost kvízu, která je jakousi zpětnou vazbou na předešlá upozornění a komunikaci. Zaměstnanec musí odpovědět na kvízovou otázku, týkající se nějaké předešlé zprávy. Pokud většina zaměstnanců, jímž byl kvíz poslán neodpoví správně, znamená to, že je třeba upravit nebo zcela změnit systém upozornění.

Aplikace SnapComm upoutá pozornost zaměstnance k určitému problému, vždy když je třeba. Na kterou zprávu je třeba upozornit hned a která zpráva nemusí být zachycena okamžitě? Toto určuje takzvaná priorita zpráv. Zprávy s vyšší prioritou mohou být zaslány zaměstnanci jinou formou než zprávy s nižší prioritou. Samozřejmostí je i kategorizace zpráv, tedy jakým subjektům budou zprávy zaslány. Subjektem, kterému je zpráva určena může být celá organizace, nebo jen její část, to je jeden nebo více týmu, jen vedoucí pracovníci, jednotlivý zaměstnanec nebo jakákoliv kombinace předchozích. Přesná specifikace příjemce zajistí, že důležitá zpráva bude doručena pouze oprávněným a relevantním osobám, a tím se zlepší efektivita práce.

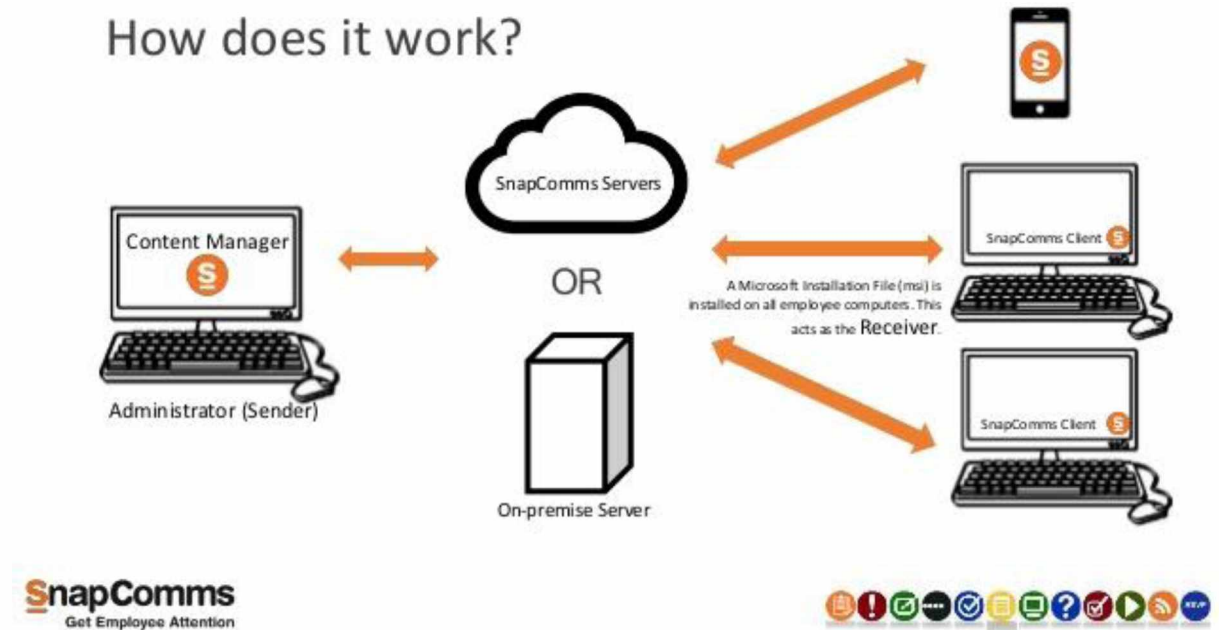
U odesílaných zpráv je možné nastavit libovolný čas a datum doručení. Může to být pro pevně stanovený čas, nebo i pro různé časy, a navíc i pro časy v různých časových pásmech. Takovéto zprávy jsou určeny například pro pobočky v jiných zemích. Tento systém umožňuje odesílatelům zajistit, aby jejich zprávy byly čteny v požadovaném čase nebo v době. Například ředitel celé společnosti chce upozornit na nový předpis a chce, aby si ho každý přečetl při příchodu do práce, pracovní doba začíná řekněme v sedm hodin a firma má pobočku v jiném časovém pásmu, které je o hodinu napřed. Tak ředitel nastaví odeslání pro zprávy pro toto časové pásmo na sedm hodin a pro druhé časové pásmo na šest. Tím se dosáhne toho, že přichozí zaměstnanci obdrží zprávu ve stejný čas tedy v sedm hodin.

Další funkcionalitou systému je opakované plánování. To umožňuje pravidelně zveřejňovat opakující se zprávy. Tyto zprávy budou zobrazeny vždy v určený den a čas. Zprávy mohou být nastaveny tak, aby se opakovaly v určitých intervalech, do té doby, dokud si je zaměstnanec nepřečte. To znamená, že se bude oznámení opakovat tak dlouho, dokud

zaměstnanec nepotvrdí přečtení, nebo nezašle odpověď. Tato funkce je nepostradatelná hlavně pro důležitou komunikaci, zabraňuje se tím ignorování zpráv ať už úmyslnému či nikoli.

Ke zlepšení účinnosti komunikace SnapComms využívá statistiky. SnapComms má možnost Reportingu, tedy aplikace shromažďuje data a na jejich základě vytváří statistiky, které zobrazuje přehledně v rámci reportů. Reporty mohou obsahovat statistiky výkonosti doručení zpráv, tedy jak dlouho trvá jejich doručení, jaká je reakční doba uživatelů na zprávy a na jakém zařízení si zprávu přečetli. Díky těmto statistickým údajům může management zlepšit doručování zpráv a tím zlepšit i komunikaci ve firmě.

Jak již bylo uvedeno, uživatelé mohou zprávy přijímat na různých zařízeních. Každé takové zařízení má ale jinou velikost obrazovky, a proto SnapComms umožňuje konfigurovat zprávy tak, aby se jejich obsah zobrazoval korektně na přijímacím zařízení (např. na mobilu).



Obrázek 3 SnapComms rozeslání zprávy

### 1.1.4 Jira

Jira je free software používaný se pro řízení projektů, úkolů, zaměstnanců, odhalování chyb a jejich řešení a pro zvýšení efektivity při vývoji software.

Jira poskytuje nástroje pomocí nichž je možné projekt rozdělit na menší dílčí sub problémy a ty dále dělit dále, až na úroveň jednotlivých úkolů, které pak mohou být přiřazeny konkrétním zaměstnancům. Každý úkol má svůj termín řešení, zodpovědného pracovníka a stav v jakém se nachází. Úkoly je možné i kategorizovat, například jestli se bude jednat o nové chování systému, nebo o pomocnou funkci, ošetření chyby atd.

Po vyřešení skupiny dílčích úkolů je možné tyto úkoly shlukovat do celků, tedy do projektů, projekty dále slučovat s jinými projekty. Výsledkem tohoto slučování je finální produkt, ať už se jedná o program, metodiku, dokumentaci, vše je možné pomocí nástroje Jira řídit.

U každého projektu je možné nastavit prioritu s jakou musí být projekt vyřešen, oprávnění, tedy kdo má právo projekt prohlížet, upravovat, či smazat. Projekty mohou být seskupeny na základě týmů, lokace, kde bude probíhat vývoj, obchodní skupiny, pod kterou spadají atd.

Po vytvoření plánu může vedoucí pracovník stanovovat množství práce, které se musí za určitý čas vykonat, a po dokončení stanovit skutečnou sumu práce, která se na projektu odvedla. Časovým intervalům vývoje projektu se říká sprint. Pro sledování pokroků vývoji neboli ve sprintu Jira nabízí tzv. graf vyhoření neboli burndown graf, který ukazuje skutečné a odhadované množství práce, kterou je třeba ve sprintu udělat.

### 1.1.5 WordPress

WordPress je systém CMS, tedy systém správy obsahu (**C**ontent **M**anagement **S**ystem). Jedná se o open sourceový projekt, který uživatelům umožňuje vytvářet webové stránky a blogy. Umožňuje ukládání dat, jako jsou například filmy, hudba, texty atd.

Důvodem, proč byl WordPress zařazen do části analýza v mé diplomové práci jsou hlavně tyto vlastnosti, které má WordPress, a má aplikace je jimi inspirována:

- Správa uživatelů – každý uživatel, má jiná práva pro prohlížení webu. Administrátoři mají práva na vše, uživatele s nižším oprávněním označujeme je jako redaktory. Ti mohou vkládat a upravovat články. A uživatele s nejnižším oprávněním tedy běžní uživatele mohou v podstatě jen prohlížet příspěvky. Rozdělením uživatelů do těchto kategorií se zaručuje, že nedojde k neoprávněné manipulaci s obsahem webu.
- Správa souborů – jak již bylo řečeno WordPress umožňuje uchovávání a správu dokumentů.
- Standardizace – Kód generovaný pomocí WordPress je zcela v souladu se standardy stanovenými W3C. Díky tomu bude takto vytvořený web fungovat ve všech dnešních prohlížečích.
- Vkládání komentářů – komentáře poskytují prostor pro ostatní uživatele, aby se mohli vyjádřit k obsahu blogu. Komentáře mohou být jakou si formou chatu.

### **1.1.6 IBM Notes/Domino**

Jedná se o balíček programů, který obsahuje klientskou a serverovou část. IBM Notes poskytuje interface pro připojení na služby, které jsou uloženy na serveru Domino. Na tomto serveru běží základní služby, které balíček nabízí. Program IBM Notes je samostatně funkční i bez serveru Domino, ale jeho použití je značně omezeno.

Bez serveru domino jej lze využít jako emailového klienta pro odesílání a přijímání zpráv, ale s ním jsou možnosti využití daleko širší. V první řadě nabízí program Notes pracovní plochu rozdělenou na více částí, do kterých může kategorizovat jednotlivé služby, ať už předinstalované nebo externí služby. Mezi základní služby Notes patří pošta, kalendář, kontakty, úkol a zápisník.

Aplikace pošta slouží, jak název naznačuje pro odesílání a přijímání emailů. Maily je možné seskupovat do složek, pro snadnější orientaci v příchozí poště. Komponenta kalendář slouží pro evidenci akcí, v kontaktech je možné udržovat adresní knihy různých subjektů. Úkol slouží k definici a rozeslání úloh pro různé uživatele a zápisník funguje jako poznámkový blok pro konkrétního uživatele. Umožňuje také ukládání internetových odkazů a souborů a uchovává historii akcí uživatele. Toto jsou však pouze základní funkce, které tento balíček nabízí. Velkou výhodou balíčku Notes je jeho rozšiřitelnost o další moduly, které byly vyvinuty jinými společnostmi, jako jsou například Google.

## **1.2 Vyhodnocení systémů:**

Při zhodnocení všech uvedených systému, jsem došel k závěru, že nejlepší systémy jsou první tři uvedené, tedy Jive, SnapComms a Deskaler. Tyto tři aplikace jsou to nejmodernější a jedná se o vedoucí systémy v oboru interní firemní komunikace. Tyto systémy poskytují takřka vše, co by mohl uživatel potřebovat pro práci ve skupině.

Hlavní nevýhodou vyhodnocených systémů, kromě WordPress, který je free, je jejich cena. Cena používání těchto software se liší podle počtu uživatelů, čím více uživatelů systém využívá tím vyšší cena a dále je u některých nutno obnovovat pravidelně licence např. u Lotus Notes, což také zvyšuje náklady na používání toho software.

### 1.3 Komunikační systém

Kvalitní interní firemní komunikace je hnací silou úspěšného podniku a šťastných a spokojených zaměstnanců, přičemž nezáleží na tom, zda se jedná o společnost s osmi nebo osmi sty zaměstnanci, efektivní komunikace je jedním z rozdílů mezi prosperující společností s motivovanými a spokojenými zaměstnanci a neúspěšnou společností, která tento rozdíl přehlíží. Nespokojenost zaměstnanců má přímý a negativní dopad na běh firmy. Když je komunikace na pracovišti efektivní a cílená na zodpovědné zaměstnance, jsou zaměstnanci více inspirovaní, angažovanější a produktivnější. Pokud zaměstnanci denně přijdou desítky nebo stovky zpráv, které se jich přímo netýkají nebo nemají vliv na jejich práci, zaměstnanci tráví část pracovní doby čtením nerelevantních zpráv, čímž klesá jak jejich spokojenost, tak i výkon.

Interní firemní komunikační systémy nabízí různé funkce pro předání zpráv mezi zaměstnanci a vedením nebo pro komunikaci s vnějšími subjekty. Průběh takovéto komunikace, ať už se jedná o komunikaci písemnou, ústní, vizuální nebo elektronickou má podstatný dopad na způsob podnikání. Cituji Herta A. Murphyho a Herberta W. Hildebranda, kteří ve své knize *Effective Business Communications* uvádějí: „Dobrá komunikace by měla být úplná, stručná, jasná, konkrétní, správná, ohleduplná a zdvořilá. Konkrétně to znamená, že komunikace by měla odpovídat na základní otázky jako: kdo, co, kdy, kde, být relevantní, a neměla by být příliš odborná, a měla by být zaměřena na příjemce a jeho zájmy, používat konkrétní fakta a čísla, používat konverzační tón pro čitelnost, v případě potřeby uvádět příklady a vizuální pomůcky, měla by být taktní a dobromyslná, přesná a nediskriminující. Nejasná, nepřesná nebo nezdvořilá obchodní komunikace může zapříčinit ztrátu drahocenného času, způsobit konflikty mezi zaměstnanci nebo odradit zákazníky, a zničit úsilí celého podniku“. (překlad autora)

Za posledních 20 let se díky pokroku v informačních technologiích zcela razantně změnilo způsoby, jakými se ve firemním prostředí komunikuje. Dnes je hlavním prostředkem pro komunikaci u většiny společností intranet. Firemní intranet lze s trochou nadsázky označit za takovou nástěnkou i když v elektronické podobě. Zaměstnanci zde mají „vystaveny“ své úkoly, pomocí emailů mohou komunikovat se svými kolegy, ukládat zde různé soubory atd...

## 2 TECHNOLOGIE A NÁSTROJE POUŽITÉ PRO TVORBU APLIKACE

Pokud předtím, než započneme proces tvorby aplikace, věnujeme trochu času na zvážení postupu, dále si rozmyslíme, které nástroje použijeme k tvorbě a nakreslíme si různé diagramy budoucího systému, můžeme tímto způsobem odhalit budoucí problémy a vyřešit je v době kdy nebyl napsán ani jeden řádek kódu a tím se vyhnout budoucím problémům a ušetřit tím i čas. Existuje mnoho metodik, jak tento postup v praxi použít mezi nejznámější patří metodika Unified Process. Zvolení vhodných nástrojů má celou řadu výhod, nejen že pro programátory je snadnější v daném prostředí pracovat, ale také kód pomocí nich vytvořený se snadněji ladí, opravuje a aktualizuje. Pro aplikaci jsou výhody rovněž nezanedbatelné. Při správné volbě nástrojů nabízí aplikace spolehlivý běh, efektivitu vykonávání příkazů, bezpečnost atd...

### 2.1 UML

UML (Unified Modeling Language, unifikovaný modelovací jazyk) je unifikovaný (= sjednocený) univerzální jazyk pro modelování systémů. Nejčastěji je používán k modelování objektově orientovaných aplikací. Toto však není to jediné k čemu je určen a k čemu ho lze použít, UML má mnohem širší využití. Avšak jazyk UML neobsahuje metodiky pro modelování, tzn. poskytuje nástroje pomocí nichž můžete vytvořit model, ale nedefinuje postup, kterým ho vytvoříte. Tento postup definují různé metodiky, jedna z nich je například metodika Unified Process.

### 2.2 UP

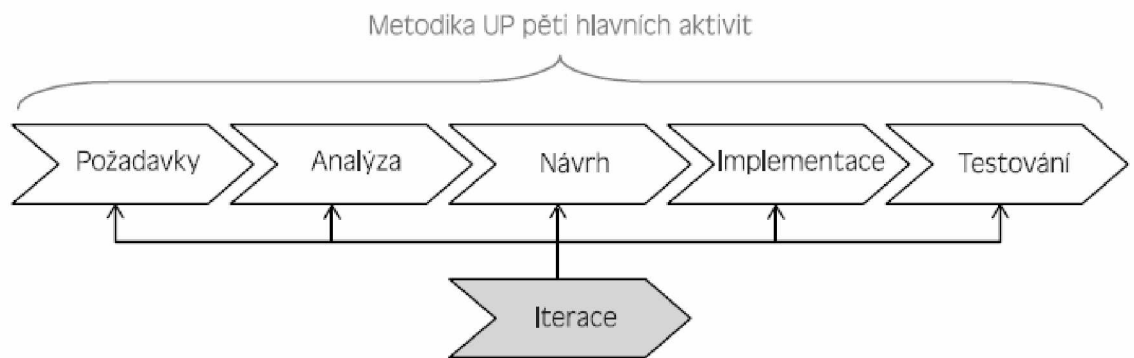
Metodika UP (Unified Process) standardem vyvinutý autory jazyka UML. Klíčovou postavou, která stojí za vznikem této metodiky je Ivory Jacobson. UP není dílem jednoho muže. Jedná se o dílo více osob, které se také podílely na tvorbě této metodiky.

Metodika UP je obecná metoda tvorby software. Neznamená to, že poskytuje jedno univerzální řešení, ale naopak poskytuje návod, jak jakýkoliv projekt vývoje rozložit na menší části a jak tyto části řešit. Z toho vyplývá, že pro každý projekt je nutné vytvořit vlastní novou instanci metody vývoje.



Základem metodiky UP jsou takzvané iterace. Každý problém, který se musí během vývoje software řešit, se dle metodiky rozdělí na menší dílčí sub problémy, které se pak postupně řeší. Každý dílčí sub problém může být považován za jednu iteraci (opakování), po dokončení všech iterací, se takto dostáváme k výsledku, tedy k řešení problému. Jednotlivé iterace prochází podobnými stádii, kterými prochází software při svém vývoji.

- Nejprve určíme, co má daná iterace vyřešit, jinými slovy, co od ní **požadujeme**
- detailně zkoumáme (**analyzujeme**) požadavky, požadavky by měly být co nejkonkrétnější a co nejdetailněji specifikované
- na základě předchozích zjištění **navrhujeme** postup řešení
- dle návrhu vytvoříme (**implementujeme**) řešení iterace
- nakonec **otestujeme**, zda iterace splňuje požadavky, jež na ni byly v počátku kladeny.



Obrázek 4 Kroky návrhu software

### 2.2.1 Požadavky

Před začátkem tvorby je vhodné si rozmyslet čeho chceme vytvoření daného objektu dosáhnout. Tedy co od něj požadujeme, proto je specifikace softwarových požadavků prvním krokem při tvorbě software. Při specifikaci požadavků musíme být co možná nejpodrobnější a nic nevynechat. Může se to zdát triviální, prostě pokud něco zapomenou, tak to doplním a jdu dál. Ale v praxi to tak nefunguje, opomenutím jisté funkce a její následným přidáním může dojít k narušení již implementovaných funkcí, které se následně nebudou chovat správně a musí se opravit. Tímto dochází k prodloužení času vývoje, s tím dochází i k prodražení. Síly, které by se jinak vynaložily na další rozvoj software se musí věnovat opravování zbytečných chyb a

celková integrita systému tím klesá. Vždy je totiž lepší vlastnost systému specifikovat hned na začátku než ji tam dodatečně dodělávat. Proto je specifikace požadavků velmi důležitou a zásadní částí postupu tvorby software.

Každý systém má svého uživatele, proto se musí určit, co který uživatel potřebuje ke své práci a kam už jeho pravomoci nesahají. Jinak řečeno, vymezit na co má oprávnění a na co ne. A dále určit jakou má uživatel v systému prioritu, tedy jak rychle budou jeho požadavky zpracovány ve chvíli, kdy je systém zahlcen požadavky ostatních uživatelů. I na tento aspekt bychom se měli při tvorbě požadavků zaměřit. Celkově bychom se měli zaměřit na:

- Určení uživatelů (aktérů) systému.
- K čemu bude konkrétně aktér systém používat (případ užití).
- Souvislosti případů užití, tedy jak jednotlivé případy užití společně vykonávají určitou funkci (struktura případů užití)

Požadavky můžeme dělit do různých skupin, my je rozdělíme na:

- Funkční požadavky – tj. určení co by měl systém dělat.
- Nefunkční požadavky – tj. omezení systému.

### 2.2.2 Analýza

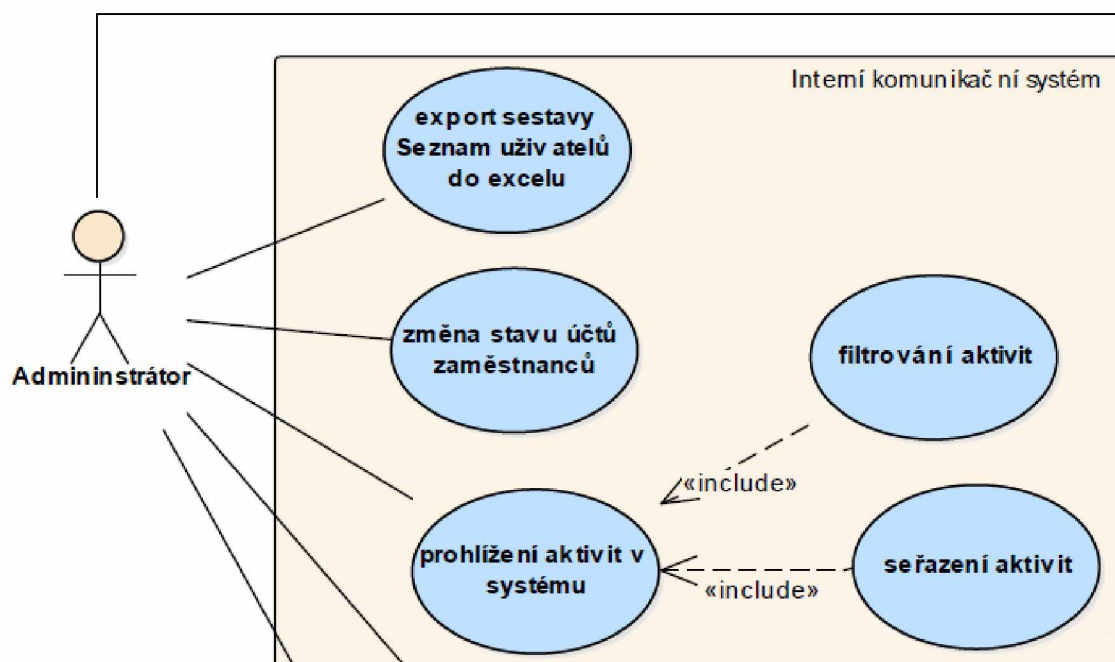
Po sesbírání požadavků, nastává proces analýzy. Při analýze se jednotlivé požadavky rozeberou do detailu a stanoví se co daný systém má ve výsledku umět.

Po provedení analýzy vznikají tyto důležité diagramy:

- Analytické třídy – tento diagram obsahuje třídy systému a jejich atributy. Dále definuje vztahy mezi jednotlivými třídami.
- Diagram případů užití (Use Case) – jedná se o diagram, který specifikuje, k čemu je daný systém možné použít a definuje, kdo ho takto používá. Tedy obsahuje dvě entity: entitu uživatele neboli aktéra a entitu obecné funkcionality neboli případu užití. To že Use Case diagram obsahuje obecné funkcionality znamená, že v diagramu není znázorněno, jak konkrétně bude systém danou funkcionalitu vykonávat, nýbrž pouze obecný popis toho, co daná funkcionalita umí, tedy k čemu je ji možné použít.

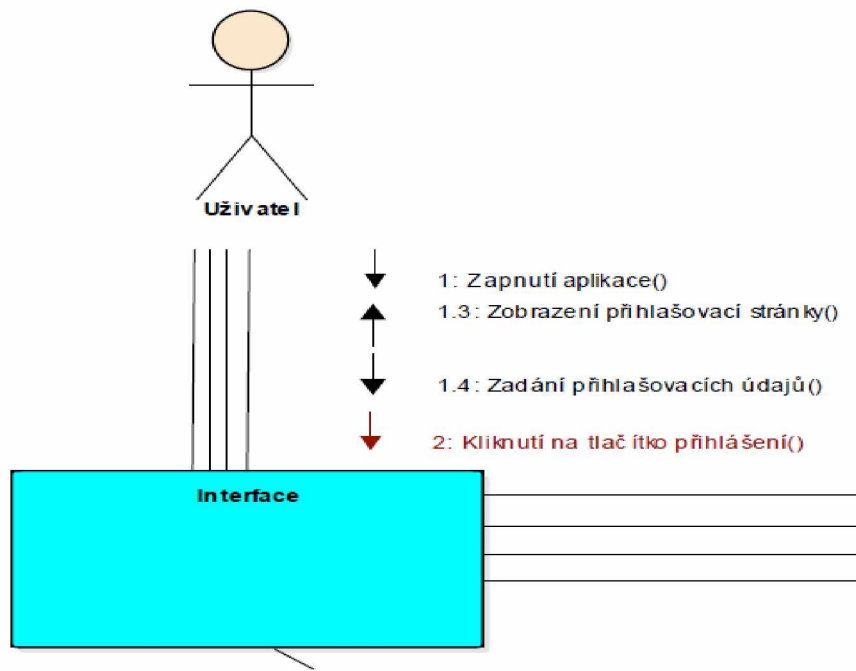
Aktérem se obecně rozumí entita, která využívá jednotlivé funkcionality systému. Aktérem může být člověk, jiný systém nebo i čas.

Je vždy důležité se předem shodnout na tom co systém bude umět a až pak řešit, jak konkrétně to bude dělat. A proto je Use Case diagram základním diagramem, který by měl vzniknout.



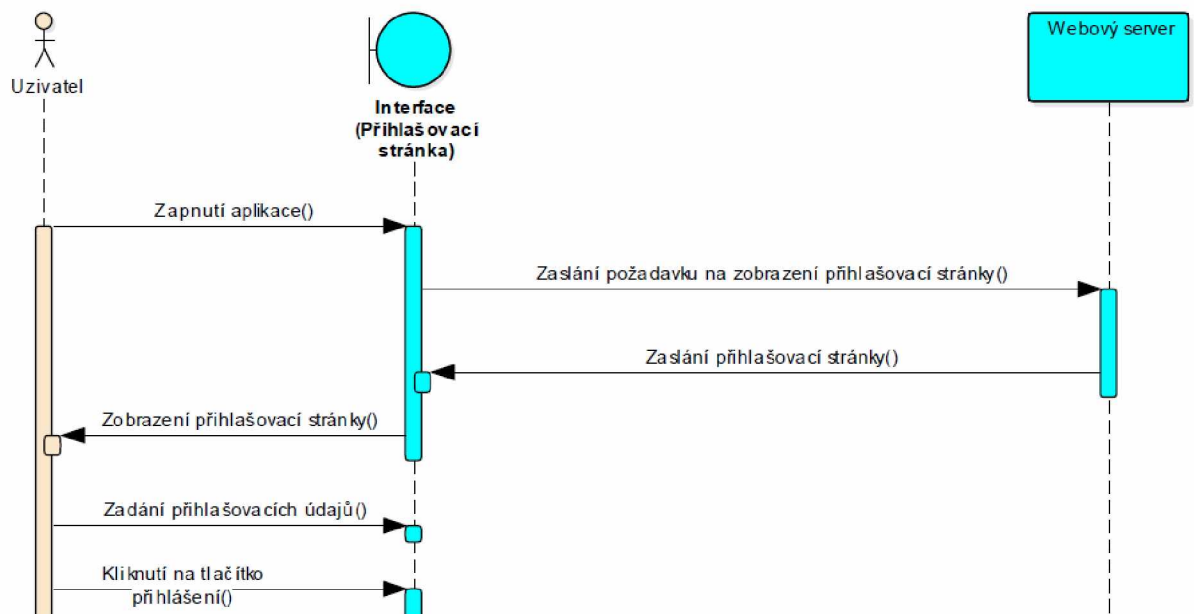
Obrázek 5 Ukázka Use Case diagramu

- Komunikační diagram – tento diagram znázorňuje předávání zpráv po určitých spojovacích drahách mezi jednotlivými objekty systému. Každý objekt může tedy poslat jinému objektu systému zprávu. Objekt, který zprávu posílá nemusí být nutně součástí systému, může to být i například člověk. V takovém případě člověk, komunikuje přes rozhraní aplikace a rozhraní pak předává uživatelské zprávy objektům aplikace.



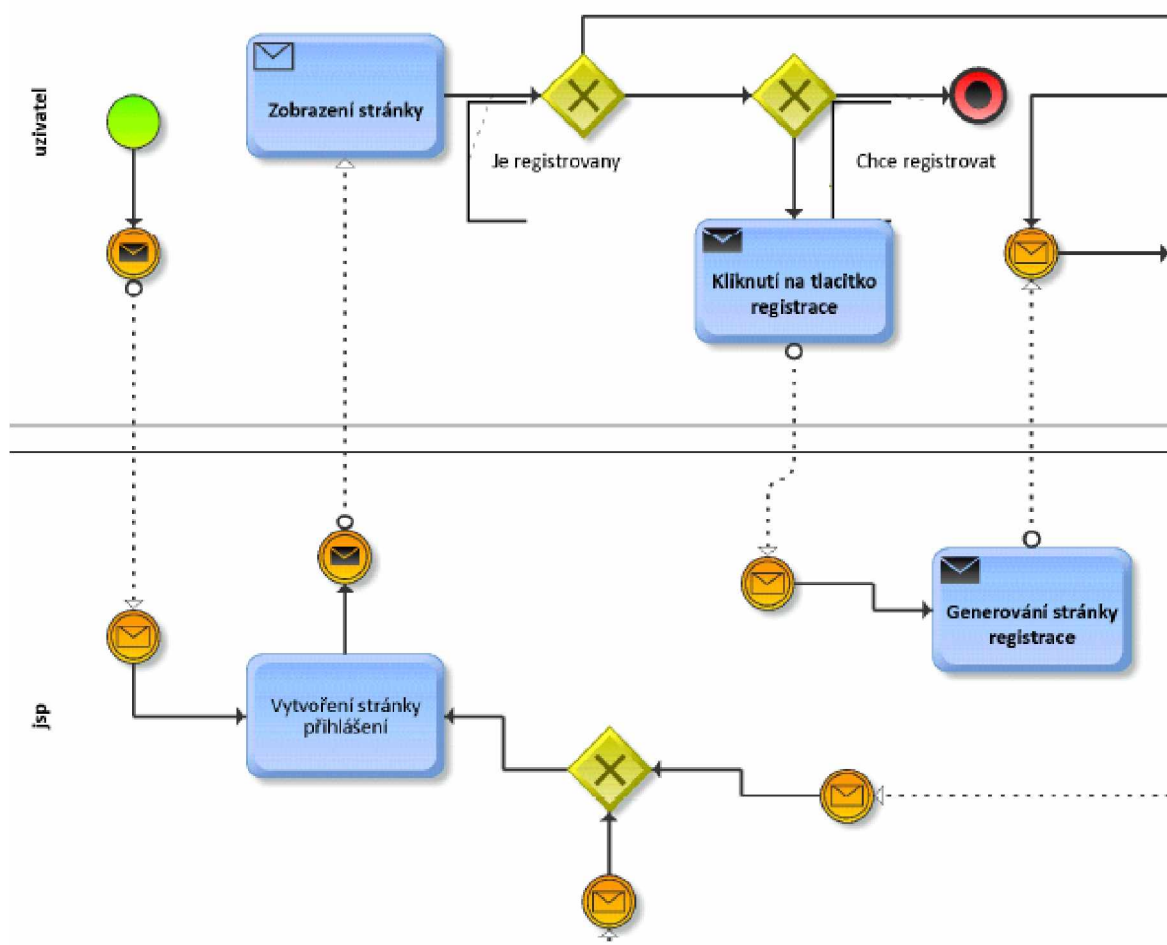
Obrázek 6 Ukázka komunikačního diagramu

- Sekvenční diagram – tento diagram je spjatý s komunikačním diagramem. Zatímco komunikační diagram znázorňuje hlavně objekty a zprávy mezi nimi a nijak zvlášť nezvýrazňuje jejich časovou posloupnost, tak sekvenční diagram se v tomto ohledu liší. Sekvenční diagram znázorňuje objekty a zprávy v časové posloupnosti.



Obrázek 7 Ukázka sekvenčního diagramu

- BPMN diagram – slouží pro modelování procesů, které probíhají v systému. Tento diagram sleduje tok každého požadavku, kterými větvemi požadavek prochází a které části systému ho obsluhují. Celý systém se pro snadnější orientaci v BPMN diagramu rozděluje do částí, které nazýváme swimlanes, v češtině bychom mohli použít ekvivalentní pojmenování buď „plavecká dráha“. Plavecké dráhy se dále seskupují do takzvaných pool neboli „bazénů“, což jsou větší celky systému, mezi nimiž se požadavky předávají pomocí zpráv.



Obrázek 8 Ukázka BPMN diagramu

### 2.2.3 Návrh

Při analýze dochází k obecné definici chování, ale o jeho konkrétní implementaci se rozhoduje ve fázi návrhu. To znamená, že v této fázi se přesně specifikuje způsob, jak se každá jednotlivá funkce implementovat, kde bude implementace umístěna, jaké datové struktury se použijí atd...

## **2.2.4 Implementace**

Implementace dalším krokem při tvorbě software. Jedná se o proces skutečné tvorby finální aplikace, založené na výstupech z předchozích kroků. Při pohledu ze strany analytika nebo návrháře, je implementací myšlena tvorba implementačního modelu. U tohoto modelu jsou navržené třídy rozděleny na jednotlivé komponenty systému. Způsob, jímž je tento proces realizován, obvykle ve značné míře závisí na vhodnosti zvoleného programovacího jazyka. Implementace se zaměřuje hlavně na tvorbu spustitelného kódu.

## **2.2.5 Testování**

Proces testování je finální proces, kterým musí aplikace projít poté co je vytvořen kód. Účelem testování je ověřit, zdali byly splněny definované požadavky, jestli systém umožňuje vše, co umožňovat má a také je-li jeho použití bezpečné a efektivní. Efektivnost závisí na rychlosti provádění funkcí, na správnosti výsledku a dostupnosti funkce v systému.

## **2.3 Enterprise Architect**

Enterprise Architect je nástroj, který nabízí mnoho funkcí potřebných při návrhu aplikace. Obsahuje různé moduly, které lze využít od prvních fází projektu, až po koncové zhodnocení celé aplikace. Enterprise architekt umožňuje využít různé metodiky a diagramy při tvorbě projektů například UML, Use Case diagramy, BPMN diagramy, Class Diagramy atd...

## **2.4 Netbeans**

NetBeans je open source vývojové prostředí pro tvorbu programů v různých programovacích jazycích. Prostedí netbeans umožňuje tvorbu, debugování, testování i nasazení vytvořených aplikací. V prostředí Netbeans jsou již v základu integrovány servery Glassfish a Tomcat.

## 2.5 ArisExpres

ArisExpres je free software poskytující nástroje pro modelování diagramů BPMN. Poskytuje veškeré elementy potřebné pro vytvoření modelu systému a následně je možné takto vytvořený model exportovat do souboru.

## 2.6 Java

Java je objektově orientovaný programovací jazyk, který vyvinula společnost Sun Microsystems. Jazyk Java byl založen na jazyku C ++, cílem vývoje bylo, aby nový jazyk byl malý, jednoduchý a mohl být přenášen napříč operačními systémy i napříč platformami.

## 2.7 HTML

HTML je zkratka Hypertext Markup Language, v překladu se jedná o Hypertextový Značkovací Jazyk, který používá definované značky (tagy) k tvorbě a formátování dokumentů pro webové aplikace. Jazyk Java a HTML zde zmiňuji jen jako poznámku, protože JSP je kombinací obou těchto technologií, a myslím si, že je vhodné je uvést.

## 2.8 JSP

JSP je zkratka pro Java Server Pages (javovské serverové stránky). Jedná se o technologii, pomocí níž můžeme vytvářet dynamické webové stránky. Dynamická webová stránka JSP se liší od klasických statických stránek HTML hlavně tím, že její obsah se nemusí obnovovat manuálně. U stránek JSP může obsah záviset na různých faktorech, například na denní či noční době, na historii prohlížení, na typu prohlížeče, atd... Díky tomu jsou JSP stránky schopny přizpůsobit svůj obsah na míru každému uživateli. Klíčovým prvkem při tvorbě dynamických JSP stránek jsou databáze, bez nich by nebylo možné zapamatovávat data spojená s uživateli. Pro práci s Java kódem definuje JSP takzvané tagy. Pokud chceme do kódu JSP přidat Java kód musíme předem definovat typ vkládaného kódu pomocí příslušného tagu.

Tag deklarace:

```
<%!    int i = 100;  %>
```

Tag výrazu:

```
<%=    i ++;    %>
```

Tag pro skriptlet:

```
<%    if(i > 100) {...} %>
```

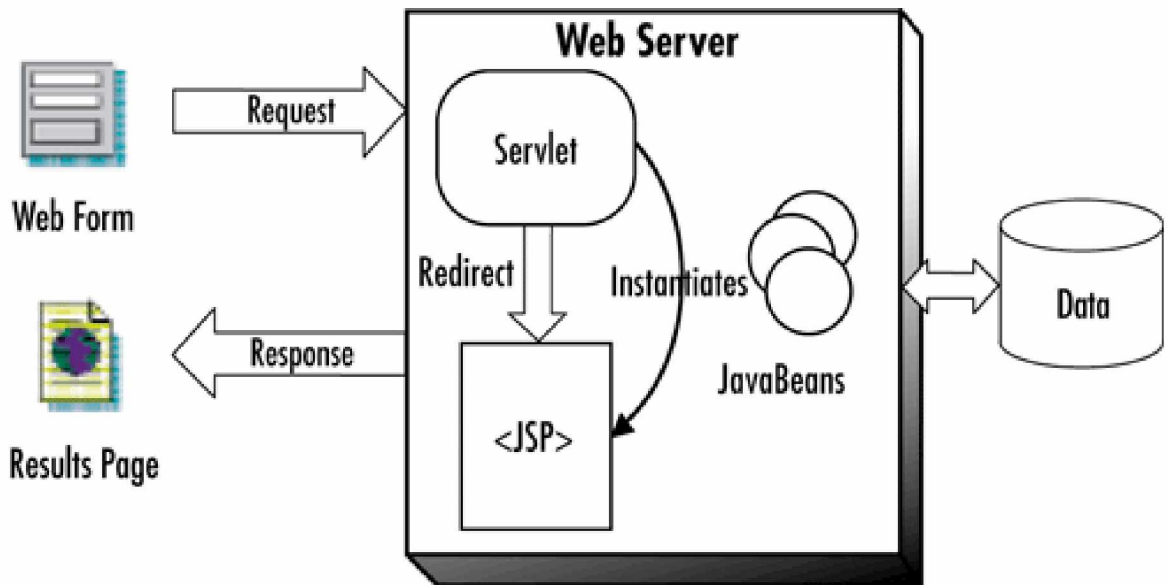
## 2.9 Historie JSP

Jazyk JSP byl představen společností Sun Microsystems v roce 1999. Jazyk JSP byl dále rozšiřován o další značky, až nakonec vznikla knihovna JSP Standard Tag Library (JSTL). JSTL je kolekcí vlastních knihovnických značek, které zapouzdřují funkčnost mnoha standardních aplikací JSP, tímto dochází k eliminaci opakování, a to činí aplikace stabilnější a kompaktnější. Spolu s JSTL byl vyvinut JSP Expression Language (EL).

Při zavedení JSP 2.0 v roce 2003 je EL včleňen do specifikace JSP. Jako novinka v JSP 2.0 je možnost tvorby vlastních souborů značek, čímž se dosahuje větší rozšiřitelnosti jazyka. Spolu s vývojem JSP byly vyvíjeny i další technologie pro tvorbu webových aplikací. V roce 2004, se jedna z nich, konkrétně technologie JavaServer Faces (JSF), zaměřila na tvorbu uživatelských rozhraní (UI). Tato technologie je založena JSP.



## 2.10 Architektura JSP



Obrázek 9 Architektura JSP

Každá komunikace mezi uživatelem a serverem začíná odesláním požadavku (HTTP request) z uživatelského prohlížeče na webový server. Webový server ověří, jestli je požadavek určen jemu, respektive pro jeho JSP stránku a předá ji do engine JSP. Rozdíl mezi standardní HTML a JSP stránkou je, že JSP stránka končí příponou .jsp místo .html. Engine JSP si načte stránku JSP z paměti a překonvertuje ji na obsah servletu. Při této konverzi dochází k převodu veškerého textu stránky na příkaz `println()` a všechny prvky JSP jsou převedeny na kód v jazyce Java.

Engine JSP posléze zkompileje servlet do spustitelné třídy. Předá požadavek servletovému engine. A servletový engine načte třídu servletu a provede ji. Během provádění servlet vytvoří výstup ve formátu .jsp. Při provádění servletový engine spolupracuje s JavaBeans, které slouží pro ukládání dat získaných z databáze nebo z jiných zdrojů. Tyto data jsou pak zahrnuta ve vygenerované stránce .jsp.

Webový server po provedení předchozích kroků odešle odpověď (HTTP response) do uživatelského prohlížeče. Zpracování odpovědi, tedy dynamicky generované stránky .jsp probíhá stejně jako kdyby šlo o statickou stránku .html.

JSP navíc kontroluje, zda servlet pro soubor JSP již existuje a porovnává, zda je datum změny v JSP starší než v servletu. Pokud je JSP starší než generovaný servlet, předpokládá se, že se obsah JSP nezměnil a že vygenerovaný servlet stále odpovídá obsahu JSP. Díky tomu je proces efektivnější, než u jiných skriptovacích jazyků jako je například PHP, a proto je JSP rychlejší.

### 2.10.1 JSP Direktivy

Direktivy JSP můžeme označit jako zprávy pro kontejner JSP. V direktivách jsou uvedeny informace, které kontejner potřebuje pro překlad příslušné stránky. V JSP existují tři základní direktivy jsou to direktiva page, include a taglib. Jejich syntaxe je následující:

```
<% jmeno_direktivy atribut1="hodnota_direktivy1" [atribut2="hodnota_direktivy2" ...] %>
```

#### <%@ page ... %>

Atributy definované pomocí této direktivy by se měly objevit na stránce pouze jednou. Tato direktiva definuje například atributy, jako je atribut skriptovacího jazyka nebo chybová stránka, která bude použita v případě výskytu chyby anebo atribut požadavku na uložení do vyrovnávací paměti. Všechny tyto atributy se vztahují k aktuální stránce JSP, při vytvoření další stránky je nutné tyto atributy znovu definovat. Tuto direktivu je možné použít v kterékoliv části stránky JSP, ale obvykle se tato direktiva uvádí v horní části stránky. Příklad:

```
<%@page import="java.io.*, java.sql.*" %>
```

#### <%@ include ... %>

Direktiva include umožňuje do stránky JSP vložit text z jiného souboru, jinak řečeno sloučí obsah aktuální stránky s jinou stránkou. Tuto direktivu je výhodné použít při definici záhlaví a zápatí stránky například takto:

```
<%@include file="includes/header.jsp" %>
```

Díky této direktivě se kód zpřehledňuje a snáze upravuje. Pokud je na různých stránkách za pomoci direktivy include přidáno záhlaví stránky, stačí upravit kód záhlaví v souboru, kde je záhlaví definováno a tato změna se automaticky promítne do všech

stránek, kde je záhlaví definováno pomocí této direktivy. Na stránce JSP tedy není zapotřebí uvádět celý kód záhlaví, stačí jeden řádek, pomocí něhož se záhlaví sloučí se stránkou a díky tomu je kód stránky úspornější, snáze modifikovatelný a přehlednější pro programátory.

### **<%@ taglib ... %>**

Jak již bylo řečeno dříve, JSP umožňuje definici vlastních značek, ale to také znamená že musí existovat nástroj, který takto definované značky povolí použít na stránce JSP. Tímto nástrojem je právě direktiva taglib. Pomocí této direktivy je kontejner informován o tom, že se na této stránce nachází nové vlastní značky. Při použití direktivy taglib se musí definovat cesta ke knihovně značek. Například takto:

```
<%@taglib uri="http://mojestranka.cz/mojeTagy" prefix="moje"%>
```

Tagy knihovny se pak definují takto:

```
<moje:můjPrvníTag> ... </moje:můjPrvníTag >
```

### **Výhody**

- Jednou z hlavních výhod JSP je možnost kombinování HTML kódu s kódem v jazyce Java a využívání servletů.
- Dynamické stránky – obsah stránek JSP se může měnit dle různých kritérií (datum, čas, záznamy v databázi atd...).
- Základní značky používané v JSP jsou snadné na pochopení.
- Externí knihovny značek – vzhledem k tomu, že v JSP může uživatel definovat a následně použít vlastní značky, vznikla řada knihoven od jiných autorů.
- Čitelnost kódu v prohlížečích – jelikož je výstupem JSP stránka v HTML je možné její kód otevřít v jakémkoliv prohlížeči.

### **Nevýhody**

- Čas prvního spuštění – při prvním spuštění vyžadují stránky více času, je to způsobeno tím, že se musí stránky JSP kompilovat na serveru.
- Zvýšený důraz na syntaxi v JSP – chyba v kódu JSP znepřístupní celou stránku.

### 2.10.2 Servlet

Servlet je třída, která přijímá požadavky (requesty) ze strany JSP stránek, které následně zpracovává a odpovídá na ně pomocí response. Prostřednictvím formuláře HTML se od uživatele nashromáždí informace, které se předají servletu. Servlet vybere odpovídající záznamy z databáze a dynamicky vytvoří webovou stránku s těmito vybranými údaji.

Servlety obsluhuje Javovská aplikace Servlet Container. Když je aplikace spuštěná na webovém serveru a obdrží požadavek, server předá žádost do servlet containeru, který ji následně předá cílovému servletu.

Fáze životního cyklu servletu:

- 1) Vytvoření servletu je spuštěno (inicializováno) zavoláním metody **init()**.
- 2) Následně servlet zavolá metodu **service()**, která zpracuje požadavek klienta.
- 3) Po obslužení požadavku je servlet ukončen metodou **destroy()**.
- 4) Servlet i po ukončení zabírá místo v paměti, a proto je servlet uklizen garbage colectorem.

### 2.10.3 web.xml

K určení toho, jak se URL adresy mapují na servlety používají webové aplikace soubor deployment descriptor. V tomto souboru se nastavují vlastnosti adres, jsou například určeny URL adresy, které vyžadují ověření. Soubor je umístěn ve složce WEB-INF/ a jmenuje se web.xml. Webový server používá konfiguraci obsaženou v tomto souboru pro identifikaci servletu, kterému má předat zpracování daného požadavku. A určuje volání metody třídy, která odpovídá metodě požadavku. Metoda požadavku je buď GET nebo POST.

Při mapování adresy URL na servlet, se použije speciální tag <servlet>, a poté se definuje mapování z adresy URL na servlet pomocí tagu <servlet-mapping>.

```

<servlet>

    <servlet-name>RegistracniServlet</servlet-name>

    <servlet-class>servlets.RegistracniServlet</servlet-class>

</servlet>

<servlet-mapping>

    <servlet-name>RegistracniServlet</servlet-name>

    <url-pattern>/RegistracniServlet</url-pattern>

</servlet-mapping>

```

#### 2.10.4 doGet

Metoda doGet () se zavolá, když uživatel klikne na odkaz nebo zadá adresu URL do adresního řádku webového prohlížeče. Může se to také volat pomocí formulářů HTML, ve formuláři se nastaví metoda s parametrem get (method="get"). V metodě doGet jsou uloženy parametry, které se při odeslání formuláře připojí k URL a odeslány ke zpracování, u JSP tuto metodu zpracovávají servlety. Metodou doGet předávané parametry mají omezenou maximální velikost a nejsou nijak šifrovány. Výhodou metody doGet je její rychlost, je to způsobeno tím, že se používá stále stejné aktivní připojení a tím se zvyšuje výkon serveru.

#### 2.10.5 doPost

Metoda doPost se provádí, když formulář odešle požadavek HTTP POST. (nastaví se method="post"). Oproti metodě doGet jsou parametry požadavku POST velikostně neomezené a posílají se v zašifrované formě. Tím dochází ke zvýšení bezpečnosti, ale požadavek doPost je ve srovnání s požadavkem doGet pomalejší.

## 2.11 CSS

CSS je zkratka pro Cascading Style Sheets. Důraz kladu hlavně na slovo „styl“. HTML se používá k vytvoření struktury webové aplikace, je tím myšleno definování prvků, jako jsou titulky, odstavce ale i formuláře, odkazy atd..., CSS naopak určuje výsledný vzhled aplikace: určuje rozvržení stránky, tedy kde bude nadpis, navigace, záhlaví a zápatí (pokud je stránka obsahuje), dále barvy pozadí a písma, u písma dále jeho font, velikost, zarovnání atd. Pro lepší názornost si představte HTML jako nově postavený dům a CSS, jako designéra interiéru. CSS tedy určí, jakou barvou bude který pokoj vymalován, kde bude nábytek atd...

```
table{  
  
    width: 80%;  
  
    border: 1px solid black;  
  
    color: black;  
  
}
```

## 2.12 Regulární výrazy

Regulární výrazy jsou používány pro zpracování textů. K tomuto účelu používají vzorec regulárního výrazu, pomocí něj se rychle prohled daný text a najde se příslušná odpovídající část. Mezi výhody regulární výrazy patří hlavně to, že jsou výkonné, flexibilní a efektivní.

V mé aplikaci se regulární výrazy používají hlavně pro kontrolu vstupních informací. Například pro tento regulární výraz pro kontrolu emailové adresy.

```
[a-zA-Z0-9._%+-]+@[a-z0-9.-]+\.[a-z]{2,}$
```

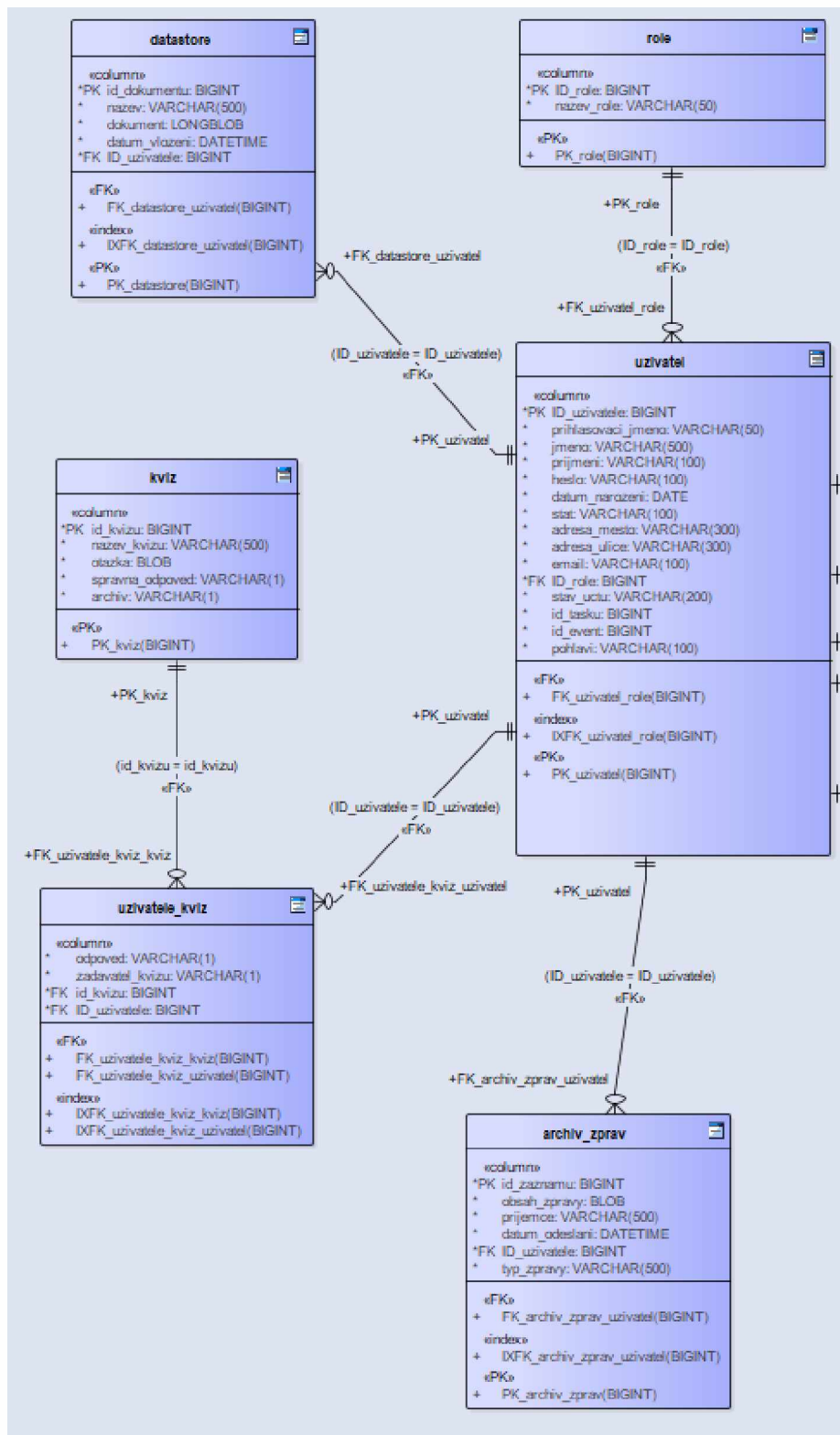
Pro lepší pochopení regulárního výrazu rozebereme tento příklad podrobněji. V první části regulárního výrazu, tato část je oddělena hranatými závorkami, je specifikováno, že emailová adresa musí obsahovat alespoň jeden znak. Tento znak může být malé i velké písmeno, číslice v rozmezí nula až devět a může to být i speciální znak tečka, podtržítka, procento, plus nebo mínus. Za tímto znakem, nebo skupinou znaků musí následovat povinně znak zavináč.

Následně opět řetězec jednoho nebo více znaků, teď již pouze malých písmen, číslic anebo dvou speciálních znaků, tedy tečky a pomlčky. Následuje povinný znak tečka a za ní musí následovat minimálně dva malé písmenkové znaky. Celý regulární výraz je ukončen znakem dolaru, který značí konec zpracovávaného řetězce.

## **2.13 Apache Tomcat**

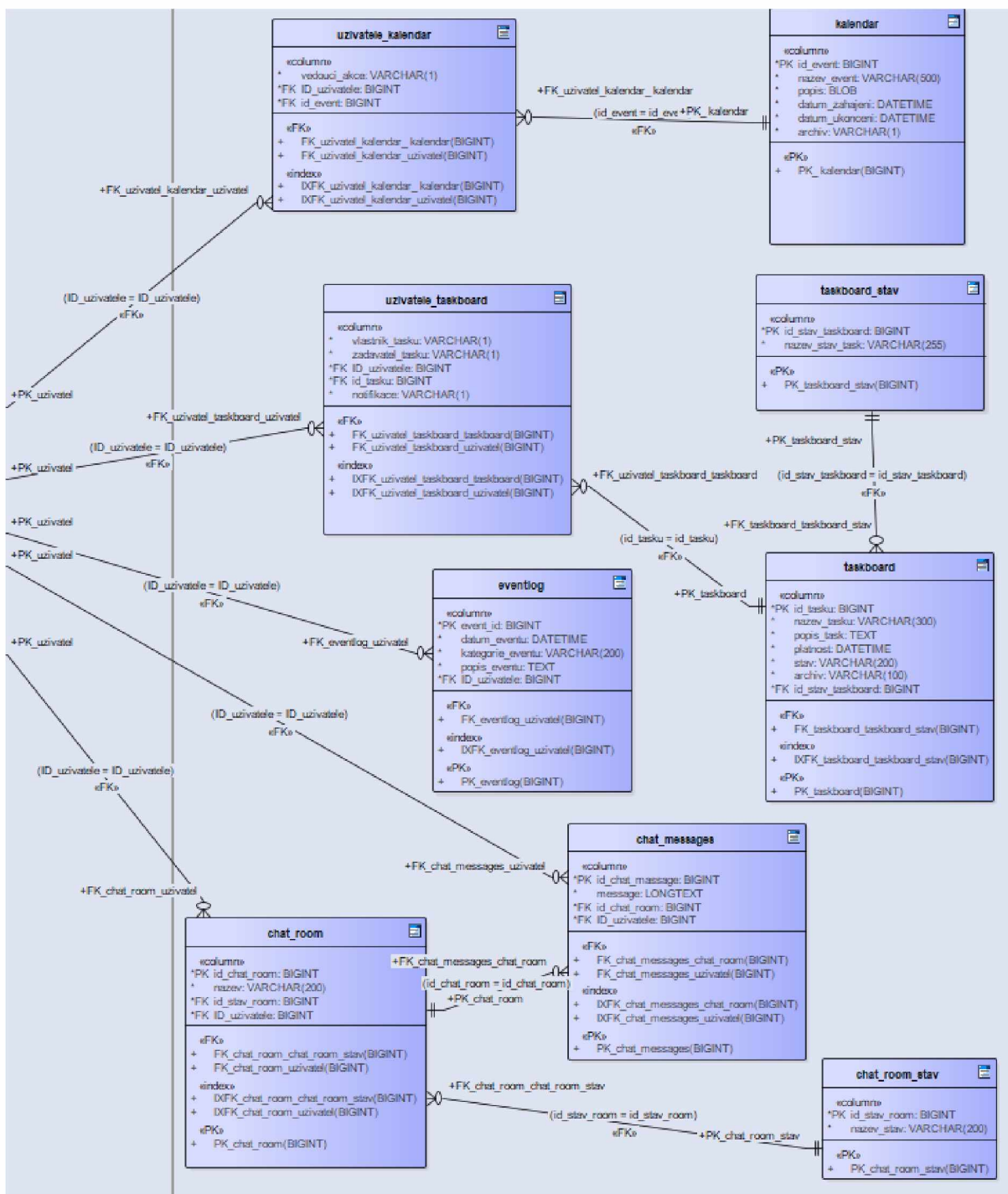
Apache Tomcat je open source server. Je poskytován současně s distribucí Netbeans. Slouží pro nasazení a běh aplikací vytvořených v Netbeans za pomoci různých technologií jako je například Java Servlet a JavaServer Pages.

### 3 DATOVÝ MODEL



Obrázek 10 Datový model část 1





Obrázek 11 Datový model část 2

## Tabulka uživatelů.

| UZIVATEL           |              |               |
|--------------------|--------------|---------------|
| Atribut            | Typ          | Klíč          |
| ID_uzivatele       | bigint (255) | Primární klíč |
| prihlasovací_jmeno | varchar(50)  |               |
| heslo              | varchar(500) |               |
| jmeno              | varchar(100) |               |
| prijmeni           | varchar(100) |               |
| pohlavi            | varchar(100) |               |
| datum_narozeni     | date         |               |
| stat               | varchar(100) |               |
| adresa_mesto       | varchar(300) |               |
| adresa_ulice       | varchar(300) |               |
| email              | varchar(100) |               |
| stav_uctu          | varchar(200) |               |
| Role_ID_role       | bigint(2)    | Cizí klíč     |

*Tabulka 1 Uživatelé*

Tabulka uživatelů je centrální tabulkou spojující všechny ostatní tabulky. Slouží k uchování údajů o všech uživateli systému. Jako primární klíč slouží ID\_uzivatele, pro přihlášení do webové aplikace slouží atributy prihlasovací\_jmeno a heslo. Heslo je uchovávané v šifrované formě. Dále jsou podrobnější osobní informace o uživateli, jako například datum narození atd. Jako poslední je v tabulce uveden atribut Role\_ID\_role, který slouží jako cizí klíč do tabulky Role. Tato tabulka je určena pro identifikaci přístupových práv uživatele.

### Tabulka rolí.

| ROLE       |             |               |
|------------|-------------|---------------|
| Atribut    | Typ         | Klíč          |
| ID_role    | bigint(2)   | Primární klíč |
| nazev_role | varchar(50) |               |

*Tabulka 2 Role*

Tabulka rolí slouží k uchování názvů aktérů a jejich příslušných ID, na jejichž základě uživatel získává příslušná přístupová práva do aplikace. Tabulka byla vytvořena pro dodržení normálových forem, v praxi však uchovává pouze 3 hodnoty a to admin, vedoucí, uživatel.

### Tabulka úkolů.

| TASKBOARD            |              |               |
|----------------------|--------------|---------------|
| Atribut              | Typ          | Klíč          |
| id_tasku             | bigint(255)  | Primární klíč |
| nazev_tasku          | varchar(300) |               |
| popis_task           | blob         |               |
| platnost             | datetime     |               |
| id_stav_taskboard_fk | varchar(100) | Cizí klíč     |
| archiv               | varchar(100) |               |

*Tabulka 3 Taskboard*

Taskboard slouží k uchování jednotlivých úkolů, které by uživatelé měli splnit, a jejich podrobností. Úkoly jsou jasně identifikovány pomocí svého id, dále obsahují název, který by měl být co nejstručnější a nejdůležitější. Dále podrobnější popis daného úkolu, ve kterém by mělo být detailně vysvětleno, co podle daného úkolu má uživatel dělat. Na toto navazuje platnost úkolu čili do kdy musí být úkol splněn. Pokud daný úkol nebyl ještě dokončen a na dokončení úkolu zbývá uživateli tři nebo méně dní, pozadí daného úkolu se zbarví oranžově. Pokud zbývá na dokončení úkolu pouze jeden den, nebo úkol nebyl v zadaném termínu splněn,

potom se pozadí úkolu zbarví červeně. Rozdílné zbarvení pozadí úkolů by mělo uživatele upozornit na nutnost dokončit úkol. Dále se v tabulce, jak již bylo naznačeno výše, nachází identifikační číslo stavu úkolu, to odlišuje, jestli byl úkol teprve zadán a momentálně na něm nikdo nepracuje, nebo jestli byl úkol již rozpracován, ale není ještě dokončen anebo jestli příslušný pracovník s daným úkolem již skončil a stav úkolu je tedy dokončen. Tyto stavy může měnit pouze uživatel, kterému byl úkol zadán k řešení. Ani spoluúčastníci úkolu ani vedoucí nemá právo měnit stav úkolu.

Je nutné věnovat pozornost rozdílu mezi vlastníkem úkolu a jeho zadavatelem, jelikož tyto osoby ne vždy musejí být nutně stejné. Jelikož ve firmách existuje jistá hierarchie, její odraz by se měl objevit i v aplikaci, protože zde vedoucí pracovníci mají právo zadávat svým podřízeným úkoly. Tím pádem, pokud nadřízený zadá úkol podřízenému, zmíněný pracovník (podřízený) se stává majitelem onoho úkolu, jelikož mu byl přidělen a on ho musí vyřešit, ovšem zadavatelem je jeho nadřízený. Nadřízenému se na stránce s úkoly navíc zobrazují i úkoly, které zadal podřízeným, avšak nemá právo ovlivňovat jejich stav, to může jen majitel onoho úkolu.

Atribut archiv značí, jestli je příslušný úkol, uložený v archivu či nikoliv. Archiv slouží pro uskladnění již dokončených úkolů, které nemá smysl zobrazovat na stránce úkolů, jelikož by se tato stránka, pokud by na ní zůstávaly všechny dokončené úkoly mohla časem neúměrně rozrůst, a to by ztížilo její používání.

#### **Tabulka pro uchování stavu úkolů.**

| TASKBOARD_STAV    |              |               |
|-------------------|--------------|---------------|
| Atribut           | Typ          | Klíč          |
| id_stav_taskboard | bigint(255)  | Primární klíč |
| nazev_stav_task   | varchar(255) |               |

*Tabulka 4 Stav tasku*

Tato tabulka byla vytvořena pro dodržení druhé normálové formy. Obsahuje identifikační číslo stavu úkolu a konkrétní stav. Konkrétní stav je definován svým názvem.

### Průniková tabulka úkolů a uživatelů.

| UZIVATEL_TASKBOARD    |             |              |
|-----------------------|-------------|--------------|
| Atribut               | Typ         | Klíč         |
| uzivatel_id_uzivatele | bigint(255) | Složený klíč |
| taskboard_id_tasku    | bigint(255) | Složený klíč |
| vlastnik_tasku        | varchar(1)  |              |
| zadavatel_tasku       | varchar(1)  |              |
| notifikace            | varchar(1)  |              |

*Tabulka 5 Taskboard uživatelů*

Jak již bylo naznačeno výše, každý úkol může mít přiřazeny vedoucí nebo spolupracovníky. Spolupracovníkem je myšlena osoba, která se podílí na realizaci daného úkolu, ale nemá právo jeho stav nijak ovlivňovat. Pro jejich uchování vznikla průniková tabulka mezi tabulkou uživatelů a úkolů.

V této tabulce se nachází složený klíč z identifikačního čísla uživatele a identifikačního čísla úkolu. Atributy `vlastnik_tasku`, `zadavatel_tasku` a `notifikace` nabývají hodnot A (ano) nebo N (ne). Na základě kombinace atributů `vlastnik_tasku` a `zadavatel_tasku` se určuje vztah uživatele k úkolu. Posledním atributem tabulky je atribut `notifikace`. Který značí, jestli bylo na daný úkol rozesláno emailové upozornění.

### Různé kombinace znamenají tyto vztahy.

| vlastnik_tasku | zadavatel_tasku | vztah  |
|----------------|-----------------|--|
| A              | A               | Uživatel je zadavatelem a zároveň řešitelem daného úkolu. V podstatě uživatel zadal úkol sám sobě. |
| A              | N               | Uživatel je řešitelem úkolu, úkol dostal zadaný od svého nadřízeného.                              |
| N              | A               | Uživatel je zadavatelem úkolu. Uživatel je vedoucí pracovník, který zadal tento úkol podřízenému.  |
| N              | N               | Uživatel je spoluřešitelem úkolu.  |

Tabulka 6 Vysvětlivky vztahů v tabulce uzivatel\_taskboard

### Tabulka kalendář.

| KALENDAR       |              |               |
|----------------|--------------|---------------|
| Atribut        | Typ          | Klíč          |
| id_event       | bigint(255)  | Primární klíč |
| nazev_event    | varchar(500) |               |
| popis          | blob         |               |
| datum_zahajeni | datetime     |               |
| datum_ukonцени | datetime     |               |
| archiv         | varchar(1)   |               |

Tabulka 7 Kalendář

Tabulka kalendář slouží pro uchovávání a evidenci akcí konaných ve firmě, jako například porady, školení, sportovní akce atd. Obsahuje název akce, její popis, datum a čas zahájení a ukončení. Atribut archiv označuje, jestli je událost uložena v archivu či nikoliv, jelikož v kalendáři akcí nemá smysl zobrazovat události, které už proběhly. Jako cizí klíč v tabulce figuruje identifikační číslo uživatele, který akci vytvořil.

### Průniková tabulka událostí a uživatelů.

| UZIVATELE_KALENDAR    |             |              |
|-----------------------|-------------|--------------|
| Atribut               | Typ         | Klíč         |
| uzivatel_id_uzivatele | bigint(255) | Složený klíč |
| kalendar_id_event     | bigint(255) | Složený klíč |
| vedouci_akce          | char(1)     |              |

Tabulka 8 Kalendářní události uživatelů

Tato tabulka slouží pro uchování účastníků příslušné akce. Ve složeném klíči je v atributu `uzivatel_id_uzivatele` uloženo id uživatele, který založil akci, která je definovaná hodnotou atributu `kalendar_id_event`. Posledním atributem tabulky je atribut `vedouci_akce`, který může nabývat hodnot A (ano) nebo N (ne). Pokud je hodnota atributu `vedouci_akce` = A znamená to, že příslušný uživatel je zadavatelem akce. V opačném případě, pokud se hodnota atributu `vedouci_akce` = N, uživatel je účastníkem akce.

### Tabulka pro uchování událostí v systému.

| EVENTLOG              |              |               |
|-----------------------|--------------|---------------|
| Atribut               | Typ          | Klíč          |
| event_id              | bigint(255)  | Primární klíč |
| datum_eventu          | datetime     |               |
| kategorie_eventu      | varchar(200) |               |
| popis_eventu          | text         |               |
| uzivatel_id_uzivatele | bigint(255)  | Cizí klíč     |

Tabulka 9 Události v systému

Eventlog je tabulka určená čistě pro administrátora. Obsahuje veškeré události provedené uživateli systému. Při provedení jakékoliv manipulace s ostatními tabulkami se do eventlogu uloží záznam o tom, o jakou akci se jedná, do které kategorie spadá, její popis a kým byla akce provedena. Kategorii je celkem deset konkrétně se jedná o tyto kategorie:

- Registrace
- Login
- Event
- Mail
- SMS
- ChatRoom
- Chat
- Dokument
- Task
- Ucet

Na základě těchto kategorií a dalších vlastností je poté administrátorovi umožněno filtrovat a řadit jednotlivé eventy, tudíž si udržovat přehled o činnostech v rámci systému.

#### Tabulka pro ukládání souborů.

| DATASTORE             |              |               |
|-----------------------|--------------|---------------|
| Atribut               | Typ          | Klíč          |
| id_dokumentu          | bigint(255)  | Primární klíč |
| nazev                 | varchar(500) |               |
| dokument              | longblob     |               |
| datum_vlozeni         | datetime     |               |
| uzivatel_id_uzivatele | bigint(255)  | Cizí klíč     |

*Tabulka 10 Datastore*

Tabulka datastore slouží pro uchování souborů. Uchovávají se ve formátu BLOB (Binary Large Object), což je formát blíže nespecifikovaných dat v binární formě. Z uvedené definice vyplývá, že v poli dokument můžeme ukládat jakýkoliv soubor s jakoukoliv příponou, velikost nahrávaného souboru je omezena buď maximální velikostí formátu BLOB, nebo limitem velikosti při nahrávání souborů na straně serveru.



Jelikož je nahrávaný soubor uložen v databázi jako typ BLOB, což je typ, který jen ukládá binární data, u kterých nepotřebuje vědět v jakém jsou formátu, je nutné při získávání dat z databáze zjistit jejich formát. To znamená mít ho někde uložený, v tomto případě je uchován v atributu nazev.

Uživatel musí uvést při vkládání do souboru do databáze, celý jeho název i s příponou. Pokud by ho neuvedl, tak by se při pokusu o jeho stažení stáhl soubor nepřehledných binárních dat. Proto je při zadávání souboru do databáze vyžadováno vyplnění polí pro název souboru a jeho příponu, jež jsou následně spojeny v příslušném servletu a uloženy do atributu nazev.

### Tabulka chatovacích místností (chat roomů).

| CHAT_ROOM             |              |               |
|-----------------------|--------------|---------------|
| Atribut               | Typ          | Klíč          |
| id_chat_room          | bigint(255)  | Primární klíč |
| nazev                 | varchar(200) |               |
| id_stav_room_fk       | bigint(255)  | Cizí klíč     |
| uzivatel_id_uzivatele | bigint(255)  | Cizí klíč     |

Tabulka 11 Chatovací místnosti

V tabulce chat\_room se ukládají informace o chatovacích místnostech. Každý uživatel má právo ji vytvořit, nebo vstoupit do již existující chatovací místnosti. Při vytvoření chat roomu je požadován pouze libovolný název, který by měl sloužit jen jako popis pro ostatní uživatele, pokud by se chtěli spojit s majitelem chat roomu nebo s jiným účastníkem diskuze. Jméno by měl tedy zakladatel chat roomu zvolit co možná nejvýstižnější.

Vlastník roomu, jenž je uveden v atributu uzivatel\_id\_uzivatele má právo zavřít jakýkoliv vlastněný room, k tomuto účelu slouží atribut id\_stav\_room\_fk. Pokud vlastník uzavře chat room, není pro něj možné onen chat room opětovně otevřít, může jen založit nový se stejným jménem, ale tento nový room nebude obsahovat žádné zprávy. Pokud by došlo k situaci, že by bylo nutné znovu otevřít již zavřený chat room, bylo by nutné kontaktovat správce databáze, aby příslušný chat room znovu otevřel.

### Tabulka pro uchování stavu chat roomu.

| CHAT_ROOM_STAV |              |               |
|----------------|--------------|---------------|
| Atribut        | Typ          | Klíč          |
| id_stav_room   | bigint(255)  | Primární klíč |
| nazev_stav     | varchar(200) |               |

Tabulka 12 Stavý chatovacích místností

Tabulka je určena pro uchování stavů, ve kterých se může chat room nacházet. Tabulka obsahuje id záznamu a název stavu, stav nabývá jen dvou hodnot a to: otevřen a zavřen.

### Tabulka pro uchovávání chat zpráv.

| CHAT_MESSAGES          |             |               |
|------------------------|-------------|---------------|
| Atribut                | Typ         | Klíč          |
| id_chat_message        | bigint(255) | Primární klíč |
| message                | longtext    |               |
| chat_room_id_chat_room | bigint(255) | Cizí klíč     |
| uzivatel_id_uzivatele  | bigint(255) | Cizí klíč     |

Tabulka 13 Chatové zprávy

Krom klíčových atributů obsahuje tabulka chat\_messages pouze jediný neklíčový atribut, a to atribut message ve formátu longtext, jelikož je to nejvhodnější formát k uložení textu, o kterém předem nevíme, jak bude dlouhý. Dále tabulka obsahuje klíčové reference na jiné tabulky, atribut uzivatel\_id\_uzivatele odkazuje na uživatele, který zprávu napsal, a ve výpisu zpráv na webové stránce se tak dosahuje zvýšení přehlednosti zpráv.

Atribut chat\_room\_id\_chat\_room je klíčovým ve více ohledech, bez něj by totiž nebylo možné vracet se do příslušného chat roomu. Při prvním vstupu do chat roomu se uživateli do jeho session uloží atribut room, který se nastaví na hodnotu identifikačního čísla příslušného roomu, pokud ovšem se do roomu vrací po napsání zprávy (tzn. id roomu je již nastaveno), vyžádá si chat identifikační číslo roomu, do kterého chce uživatel opětovně vstoupit a na jehož základě vypíše všechny příslušné zprávy.

**Tabulka pro uložení odeslaných zpráv přes mail nebo sms.**

| ARCHIV_ZPRAV          |              |               |
|-----------------------|--------------|---------------|
| Atribut               | Typ          | Klíč          |
| id_zaznamu            | bigint(255)  | Primární klíč |
| obsah_zpravy          | varchar(500) |               |
| prijemce              | blob         |               |
| datum_odeslani        | datetime     |               |
| typ_zpravy            | varchar(500) |               |
| uzivatel_id_uzivatele | bigint(255)  | Cizí klíč     |

*Tabulka 14 Archivované zprávy*

V této tabulce jsou uloženy veškeré zprávy, které byly poslány pomocí aplikace. Do tabulky se ukládá příjemce zprávy, její obsah, datum odeslání a v atributu `uzivatel_id_uzivatele` je uloženo identifikační číslo uživatele, který zprávu odeslal. Tato tabulka uchovává jak emailové zprávy, tak zprávy sms, pro rozlišení, o jaký typ zprávy se jednalo slouží atribut `typ_zpravy`.

**Tabulka pro kvízových otázek.**

| KVIZ            |              |               |
|-----------------|--------------|---------------|
| Atribut         | Typ          | Klíč          |
| id_kvizu        | bigint(255)  | Primární klíč |
| nazev_kvizu     | varchar(500) |               |
| otazka          | Blob         |               |
| spravna_odpoved | varchar(1)   |               |
| archiv          | varchar(1)   |               |

*Tabulka 15 Tabulka kvízů*

Tato tabulka slouží pro uchování kvízových otázek. V attributech jsou uloženy informace o názvu kvízu, znění otázky, správná odpověď a příznak archivace kvízu. Správná odpověď má rozsah jen jedno písmeno, jelikož odpověď na otázku je ve formě a), b), c)...

#### Tabulka pro odpovědi na kvízové otázky.

| UZIVATELE_KVIZ        |             |              |
|-----------------------|-------------|--------------|
| Atribut               | Typ         | Klíč         |
| uzivatel_id_uzivatele | bigint(255) | Složený klíč |
| kviz_id_kvizu         | bigint(255) | Složený klíč |
| odpoved               | varchar(1)  |              |
| zadavatel_kvizu       | varchar(1)  |              |

Tabulka 16 Tabulka odpovědi na kvízové otázky

V této tabulce se uchovávají odpovědi na kvízové otázky uživatelů. Ve složeném klíči je identifikační kvízu. Atribut `uzivatel_id_uzivatele` slouží pro určení uživatele, který odpovídá na otázku. Dále atribut `zadavatel_kvizu` slouží jako příznak toho, jestli je uživatel zadavatelem kvízu či na něj odpovídá. Podobně jako u příznaků v předcházejících tabulkách nabývá hodnot A (ano je zadavatelem) a N (ne není zadavatelem = odpovídá na otázku).

#### Tabulka stavů odpovědí na otázku.

| Stav         | Příznak                   |
|--------------|---------------------------|
| Odpovězeno   | A-B-C-D... (s výjimkou N) |
| Neodpovězeno | N                         |

Tabulka 17 Stavů odpovědi

## 4 UŽIVATELSKÁ PŘÍRUČKA

### 4.1 Šifrování hesla

O zašifrování hesla se stará třída `hassPassword.java`. Při registraci se řetězec, ve kterém je heslo uvedeno, převede na pole `Bytů`, společně s předem zvolenou hodnotou soli (`SALT`) jsou tyto hodnoty spojeny do jednoho záznamu, který je zahashován funkcí `SHA-256` a převeden na řetězec v hexadecimálním tvaru. Následně je heslo v této podobě uloženo do databáze. Při přihlášení se heslo, které uživatel zadal opět zahashuje a porovná s heslem uloženým v databázi, jelikož sůl zůstává stejná a za předpokladu, že zadané heslo je validní, hodnoty zahashovaných hesel by měly odpovídat, a tudíž by se uživatel měl být schopný přihlásit.

### 4.2 Část pro registraci a přihlášení

#### 4.2.1 Přihlášení

Přihlašovací formulář je první, co se zobrazí uživateli po vstupu na stránky aplikace. Pokud je uživatel již zaregistrován, stačí mu vyplnit validní přihlašovací údaje (přihlašovací jméno + heslo) a bude přesměrován na úvodní stránku aplikace. Pokud ovšem ještě není uživatel zaregistrován nemá právo dále prohlížet stránky. Pokud chce tedy manipulovat s aplikací je registrace nutná.

#### 4.2.2 Registrace

Při registraci je od uživatele požadováno, aby zadal následující hodnoty:

- Jméno
- Příjmení
- Přihlašovací údaje:
  - Uživatelské jméno
  - Heslo
  - Ověření hesla

- Pohlaví
- Bydliště skládající se z:
  - Státu
  - Města
  - Adresy
- Datum narození
- Email

U všech uvedených hodnot je vyžadováno, aby byly vyplněny. Pro kontrolu validity vstupu, bylo využito regulárních výrazů, které zajišťují, aby byly hodnoty vyplněny v očekávaném tvaru a v příslušné délce. To se projevuje například u hesla, které musí být zadáno v minimální délce osmi znaků, s využitím malých i velkých písmen a číslic, tímto způsobem se zvyšuje síla hesla. Dalším příkladem využití regulárních výrazů je email, který musí být zadán ve formátu jméno@doména.typ (př. pavel@novak.cz). Pokud by zadaný řetězec nebyl vyplněn v tomto tvaru nebylo by možné odeslat registrační formulář k dalšímu zpracování. Pomocí regulárních výrazů se také zvyšuje zabezpečení celé aplikace. Po registraci je uživateli přiřazeno oprávnění uživatel, jedná se o základní oprávnění.

### 4.3 Úvodní stránka

Po úspěšném přihlášení do webové aplikace, se zobrazí úvodní stránka, na které se zobrazí dynamicky generované uvítání pro přihlášeného uživatele a maximálně tři nejbližší akce z kalendáře. Zobrazení akcí z kalendáře slouží jako připomínka blížících se termínů. Pokud má přihlášená osoba uživatelská oprávnění, může se na úvodní stránce zobrazit jedna nebo více kvízových otázek. Kvízové otázky vytváří nadřizený, tedy osoba s oprávněním vedoucí. Oproti předcházejícím stránkám (stránky registrace a přihlášení) je již na této a všech dalších stránkách záhlaví a zápatí. Záhlaví stránky obsahuje navigaci na další stránky.

Obrázek 12 Úvodní stránka

## 4.4 Stránka účtů

Jedná se o stránku, která je určena čistě pro administrátora. Na této stránce má administrátor právo měnit stav účtu a roli zaměstnance. Stavem účtu je myšlen záznam, který obsahuje informace, jestli je daný účet aktivní, či nikoliv. Pokud není účet aktivní je nemožné se přihlásit, ale záznam o tomto účtu stále existuje v tabulce uživatelů, jelikož systém uchovává veškeré činnosti, které se v něm odehrály. Odstraněním záznamu uživatele z tabulky uživatelé, by došlo k porušení konzistence databáze a následně celého systému.

Co se týče rolí zaměstnance, administrátor má právo měnit roli vedoucích a běžných zaměstnanců. Nemůže, změnit roli sobě nebo jiným administrátorům a běžným zaměstnancům nebo vedoucím pracovníkům nemůže přiřadit administrativní roli. Toho lze dosáhnout jen přímou manipulací příslušného záznamu v databázi. Co se týče změny práv u vedoucího, může mu nastavit běžná práva uživatele a uživatelská práva může povýšit na práva vedoucího. Seznam účtů je možno vyexportovat do excelu.

## 4.5 Stránka událostí

Stránka událostí je opět určena jen pro administrátory systému. Na této stránce se zobrazují veškeré činnosti prováděné všemi uživateli v systému. Jedná se tedy o výpis záznamů z databáze. Tyto záznamy pak administrátor může procházet a udržovat si tak přehled o činnostech uživatelů systému. Jelikož, se při každodenním užívání systému tato tabulka zvětšuje je nutné záznamy v ní obsažené nějak uspořádat a omezit. K tomuto účelu slouží funkce filtr a seřazení.

Pomocí funkce seřazení, může administrátor záznamy řadit dle identifikačního čísla záznamu, data vytvoření záznamu a kategorie záznamu. Pro zúžení výběru záznamů slouží funkce filtr, pomocí této funkce může administrátor vybrat jen záznamy, které ho zajímají. Filtrovat může nad uživatelem, datem záznamu a jeho kategorií. Filtry je navíc možno kombinovat. Takže administrátor může vyhledávat například veškerou aktivitu uživatele A, v den B v kategorii C. Nebo ho zajímají události dne B ve všech kategoriích a pro všechny uživatele.

## 4.6 Stránka dokumenty

Na této stránce mohou uživatelé ukládat dokumenty do databáze a následně je i z databáze stahovat. Při vkládání je nutné uvést název souboru, tedy pod jakým jménem má být v databázi uchován, příponu souboru a samozřejmě soubor, který chcete nahrát do databáze. Stahování funguje na principu uvedení identifikačního čísla souboru, které naleznete v seznamu níže na stránce a stisknutím tlačítka download. Jakýkoliv uživatel může stáhnout jakýkoliv soubor, ale na jeho smazání má právo jen administrátor nebo vedoucí.

| ID dokumentu | Název dokumentu | Datum vložení       |
|--------------|-----------------|---------------------|
| 5            | arcusys.zip     | 2019-07-19 00:00:00 |
| 6            | pokyny.pdf      | 2019-07-19 00:00:00 |
| 7            | zadani.png      | 2019-08-13 11:48:59 |

Obrázek 13 Stránka dokumentů



## 4.7 Stránka Taskboard

Zde uživatel vidí všechny své úkoly, které má momentálně zadány. Úkoly se mohou nacházet v různých fázích rozpracování. Od úkolů, které ještě nebyly započaty, přes započaté úkoly až k úkolům dokončeným nebo zrušeným. Tato stránka obsahuje dva odkazy na další stránky, jeden odkaz je na stránku pro přidání úkolu a druhý odkaz je na stránku archivu úkolů. V archivu se zobrazují dokončené úkoly, které již nemá smysl držet na hlavní stránce Taskboard.

Pro úpravu stavu slouží input area pro identifikační číslo úkolu, kterému chcete změnit stav a jedním ze tří tlačítek se provede změna stavu. Stav se změní podle toho, na které tlačítko uživatel kliknul.

Další funkce taskboardu.

- Přeplánování úkolu – slouží pro změnu data, do kdy má být příslušný úkol dokončen.
- Smazání úkolu – slouží pro odstranění úkolu z databáze.
- Zrušení úkolu – slouží pro změnu stavu úkolu, indikuje, že již není nutné úkol řešit.
- Archivace úkolu – slouží pro přemístění úkolu z hlavní stránky do archivu. Archivovat je možné jen dokončené úkoly.

Pod těmito funkcemi se může nacházet jedna nebo dvě tabulky, v závislosti na roli uživatele v systému. Pokud je uživatel zaměstnanec, vidí pouze své úkoly a může s nimi manipulovat. V popisu úkolu je zobrazeno:

- identifikační číslo,
- název,
- popis,
- majitel
- a datum platnosti, tedy datum, do kterého by měl být úkol splněn.

Pokud je uživatelské oprávnění vyšší než oprávnění zaměstnance, znamená to, že uživatel může úkoly zadávat podřízeným (vedoucí zaměstnancům, administrátor všem kromě jiných administrátorů), a tudíž by tito uživatelé měli mít o takovýchto úkolech přehled. K tomu slouží druhá tabulka (s jejím obsahem není možné manipulovat).

Dále se na stránce nachází tabulkový seznam všech nearchivovaných akcí.

| Nezačaté  | Momentálně v řešení   | Dokončené  |
|---|---|--|
| <p>ID: 53<br/> Název: Vytisknout diplomovou práci<br/> Majitel: LangerT<br/> Platnost: 2020-05-04 00:00:00<br/> Popis: Poslat diplomovou práci k vytisnutí</p> <p>ID: 54<br/> Název: Odevzdat diplomovou práci<br/> Majitel: LangerT<br/> Platnost: 2020-05-15 00:00:00<br/> Popis: Odevst diplomovou práci do Pardubic</p> | <p>ID: 52<br/> Název: Dokumentace diplomové práce<br/> Majitel: LangerT<br/> Platnost: 2020-05-02 00:00:00<br/> Popis: Dodřat dokumentaci k diplomové práci</p> | <p>ID: 55<br/> Název: Praktická část diplomové práce<br/> Majitel: LangerT<br/> Platnost: 2020-05-01 00:00:00<br/> Popis: Naprogramovat diplomovou práci</p> |

Obrázek 14 Ukázka úkolů

#### 4.7.1 Stránka přidání úkolu

Na této stránce je zobrazen formulář pro přidání úkolu. V závislosti na právech může uživatel přidat úkoly podřízeným zaměstnancům. Pokud uživatel není vedoucí pracovník, může přiřadit úkol jen sám sobě. Další možností, pokud je uživatel vedoucí nebo administrátor je přiřazení spoluřešitelů úkolu, tedy osoby, která se na řešení úkolu podílí a nemůže nijak ovlivňovat stav úkolu v systému. Spoluřešitelem může být jen další uživatel systému.

#### 4.8 Stránka kalendář

Slouží k zadávání a zobrazení různých akcí konaných ve firmě. V horní části stránky se nachází formulář pro zadání akce. Tato část se zobrazí jen u uživatelů s vyšším oprávněním, než je oprávnění běžného uživatele tedy zaměstnance. Při zakládání události je nutné vyplnit její název, popis a datum konání, které se skládá z data a času zahájení a ukončení akce.

Další funkce kalendáře.

- Přeplánování události – slouží pro změnu data konání události.
- Smazání události – slouží pro smazání úkolu z databáze.
- Archivace události – slouží pro přemístění události z hlavní stránky do archivu.

## 4.9 Část komunikace

V záložce komunikace se nachází více stránek pro různé formy komunikace.

### 4.9.1 Mail

Na stránce je klasický formulář pro odeslání emailu. Vyplníte mail příjemce, předmět zprávy a její obsah a můžete zprávu odeslat. O odeslání se stará EmailSendingServlet ve spolupráci s externími knihovnami mail.jar a activation.jar.

### 4.9.2 SMS

Podobně jako u mailu je na stránce je formulář pro odeslání SMS. Vyplníte jen telefonní číslo a zprávu a můžete zprávu odeslat.

### 4.9.3 Chat

Stránky pro chat jsou rozděleny do dvou částí. Jedna část je pro chatovací místnosti, tzv. chat roomy, a druhá je pro jejich obsah. V první části máte možnost vytvořit, vstoupit a pokud jste majitel tak i zavřít chatovací místnost.

| ID chat roomu | Název chat roomu | Majitel chat roomu |
|---------------|------------------|--------------------|
| 3             | Třetí chat room  | LangerT            |

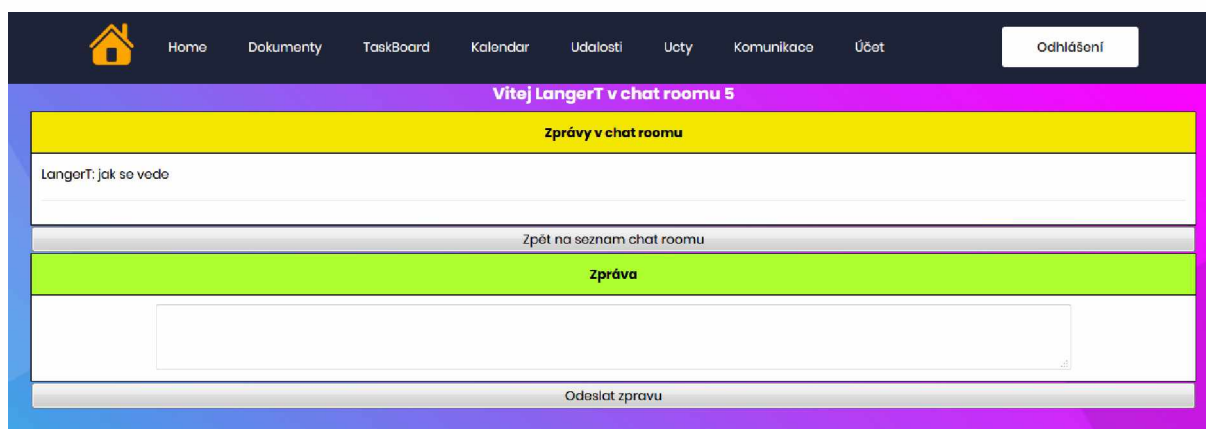
Vstoupit do chat roomu 3  
Zavřít chat room 3

| ID chat roomu | Název chat roomu | Majitel chat roomu |
|---------------|------------------|--------------------|
| 5             | Pátý chat room   | sin8               |

Vstoupit do chat roomu 5

Obrázek 15 Seznam chatovacích místností

V druhé části už v konkrétním vámi zvoleném chat roomu můžete psát zprávy pro ostatní uživatele.



Obrázek 16 Chatovací místnost

## 4.10 Část účet

V této části má uživatel možnost zobrazení podrobností o svém účtu a může si zde změnit osobní údaje.

### 4.10.1 Stránka podrobnosti

Na této stránce je zobrazen podrobný seznam uživatelových osobních údajů, které zadal při registraci.

### 4.10.2 Stránka změna údajů

Jak vyplývá z názvu, slouží tato stránka pro změnu údajů. Není rozdělena do sekcí, ale funkčnosti to nevadí. Pokud uživatel chce změnit jen jeden údaj, má tuto možnost. Stačí vyplnit příslušné pole, které chce změnit a následně se provede změna položky tabulky v databázi. Tuto činnost řídí servlet `zmenaUdajuServlet`.

### 4.10.3 Stránka Kvíz

Tato stránka je určena pouze pro vedoucího pracovníka. Pomocí této stránky může zadávat kvízy, sledovat jejich stav, případně již ukončené kvízy archivovat.

| Název kvízu     | Správná odpověď |
|-----------------|-----------------|
| 10 : druhy kvíz | B               |

Otázka: jak dopadla zkouška A)dobře B)špatně

Odpověď uživatele Filip : B

Odpověď uživatele Zdena : N

Obrázek 17 Stránka kvízů

## 5 PROGRAMOVÁ PŘÍRUČKA

### 5.1 Instalační příručka

Před prvním spuštěním aplikace je nutné naplnit tabulky hodnotami, které s v rámci systému používají. Konkrétně se jedná o tabulku rolí, v níž jsou uloženy identifikační čísla a názvy rolí uživatelů v systému, pokud by tato tabulka nebyla vyplněna, nebylo by možné rozlišovat oprávnění jednotlivých uživatelů. Tato tabulka musí být vyplněna těmito hodnotami a v tomto pořadí.

| id_role | nazev_role |
|---------|------------|
| 1       | admin      |
| 2       | vedouci    |
| 3       | uzivatel   |

*Tabulka 18 Konkrétní hodnoty pro tabulku rolí*

Dále by se měl vytvořit alespoň jeden uživatel systému s rolí administrátora. Jelikož při prvním spuštění není nijak možné vytvořit uživatele s administrátorskými právy pouze přes systém, musí se tento uživatel založit ve spolupráci s databází. Postup je tedy následující, budoucí administrátor se registruje přes aplikaci a následně si pomocí příkazu update upraví v tabulce uživatelů atribut role\_id\_role na roli administrátora. Tedy přibližně takto:

```
UPDATE uzivatel SET role_id_role = '1' WHERE id_uzivatele = '?'
```

Místo otazníku bude identifikační číslo uživatele, kterému chceme přiřadit administrátorská práva. Pokud systém nebyl ještě nikým použit mělo by toto číslo být jedna.

Poté je nutné konfigurovat emailovou službu. Jelikož se maily odesílají přes gmailový SMTP server, je nutné upravit nastavení jak v rámci aplikace, tak v rámci mailu, přes který se budou zprávy posílat.

V aplikaci se musí upravit soubor WEB-INF/web.xml v něm se nalézá konfigurace SMTP serveru. Upravíte hodnoty pouze dvou parametrů. A to hodnotu parametru user, jehož hodnota musí být váš emailový účet služby gmail. V mém případě se jedná o univerzitní účet, s příponou student.upce.cz, ale email může mít i jinou příponu, za předpokladu, že se jedná o

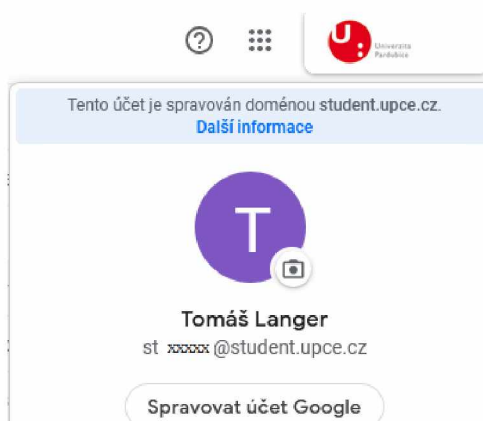
službu gmail (př: novak@gmail.com). V parametru pass následně vyplníte jako hodnotu vaše emailové heslo.

```
<context-param>
  <param-name>user</param-name>
  <param-value>stXXXXX@student.upce.cz</param-value>
</context-param>
<context-param>
  <param-name>pass</param-name>
  <param-value>vaše_heslo</param-value>
</context-param>
```

Jelikož jsem aplikaci vytvářel na localhostingu, musel jsem během posílání zpráv vypnout antivirový program, jelikož emailový štít bránit v odeslání zprávy.

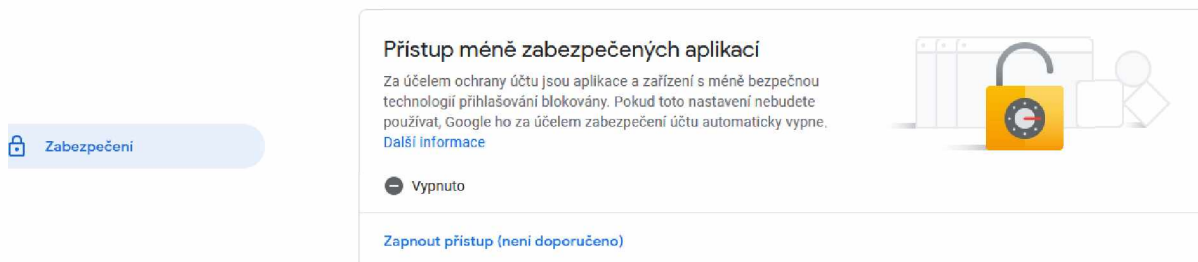
Po nakonfigurování aplikace se musí nakonfigurovat i gmailový účet. Jelikož Google své služby chrání před neznámými aplikacemi po prvním pokusu o navázání spojení s vaším účtem vám Google zašle zprávu, že se tato aplikace pokusila o přístup k vašemu mailu a zeptá se vás, jestli chcete této aplikaci důvěřovat? Abyste byli schopni mail odeslat, musíte této aplikaci důvěřovat.

Dále v rámci svého gmailového účtu musíte nakonfigurovat zabezpečení a umožnit přístup i méně bezpečným aplikacím. Toto provedete tím, že v pravém horním rohu kliknete na ikonu svého účtu a dále kliknete na tlačítko Spravovat účet Google.



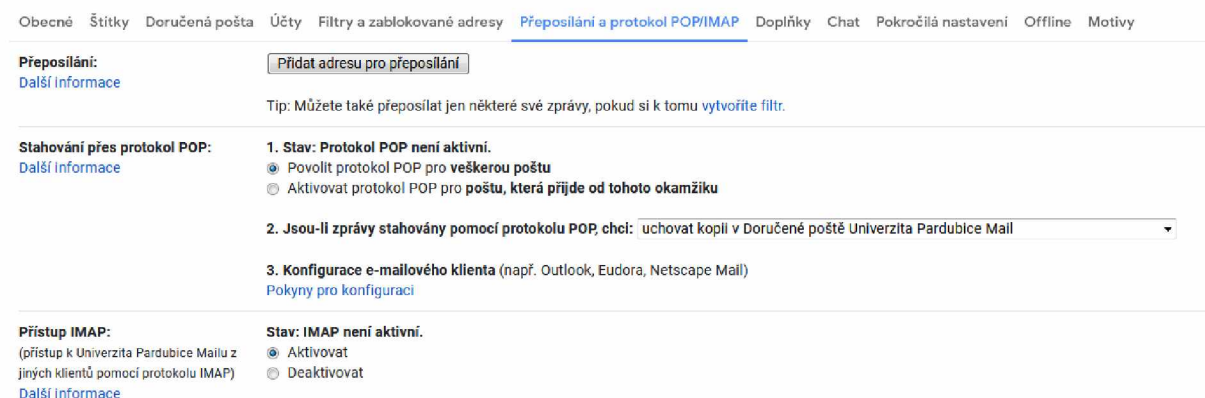
Obrázek 18 Nastavení zabezpečení mailu krok 1

Poté se vám otevře okno pro správu vašeho účtu, kliknete na záložku Zabezpečení a v ní povolíte přístup méně bezpečným aplikacím. Sice je zde varování, že Google nedoporučuje povolovat méně bezpečné aplikace, ale je nutné tuto volbu povolit.



Obrázek 19 Nastavení zabezpečení mailu krok 2

Jako poslední krok musíte v nastavení mailu v záložce Přeposílání a protokol POP/IMAP povolit služby POP a IMAP.



Obrázek 20 Nastavení zabezpečení mailu krok 3

Jako poslední je konfigurace SMS zpráv, jelikož v dnešní době všechny SMS brány, vyžadují registraci a příslušné poplatky za posílání zpráv, využil jsem k poslání SMS API Twilio, které umožňuje posílat testovací zprávy zdarma. V podstatě testovací zpráva, je zpráva, kterou pošlete na číslo +15005550006 a tato zpráva se sice neodešle, ale API Twilio vám v odpovědi do aplikace pošle zprávu, že pokud byste si zaplatil tuto službu vaše zpráva by se odeslala úspěšně, nebo vám v případě chyby zašle chybovou hlášku.

Pokud byste si službu předplatili musíte v aplikaci v servletu odesliSMSServlet, změnit parametry SID\_UCTU a AUTORIZACNI\_TOKEN, tak aby odpovídaly vašemu účtu Twilio. Dále pak jen odkomentovat na stránce sms.jsp kód, který slouží jako input pro sms zprávy.



## 5.2 Ukázka zdrojového kódu

Ukázka formuláře pro odeslání odpovědi na kvíz, který se nalézá na úvodní stránce.

```
<%  
if (session != null && (int) session.getAttribute("role") == 3 & session.getAttribute("uzivatel")  
!= null) {  
  
PreparedStatement psKviz = con.prepareStatement("SELECT * FROM kviz JOIN  
uzivatele_kviz ON uzivatele_kviz.kviz_id_kvizu = kviz.id_kvizu JOIN uzivatel ON  
uzivatele_kviz.uzivatel_id_uzivatele = uzivatel.id_uzivatele WHERE kviz.archiv = 'N' AND  
uzivatele_kviz.zadavatel_kvizu = 'N' AND uzivatele_kviz.odpoved = 'N' AND  
uzivatel.id_uzivatele = ? ");  
  
psKviz.setInt(1, (int) session.getAttribute("id_uzivatele"));  
ResultSet kviz = psKviz.executeQuery();  
  
while (kviz.next()) {  
String IDKvizu = kviz.getString("id_kvizu");  
  
%>
```

```
<table style="width: 100%; border: #000 solid">  
  <tr>  
    <td><div id="Kviz"><%= kviz.getString("id_kvizu")%> :  
    <%= kviz.getString("nazev_kvizu")%></div></td>  
    <td><div id="spravnaOdpoved"><%= kviz.getString("otazka")%> </div></td>  
  </tr>  
</table>  
<form action="odpovedetNaOtazkuServlet" method="POST">  
  <input type="text" name="kvizOdpoved" pattern="[a-zA-Z]" maxlength="1" required>  
  <input type="submit" name="kvizIDOdpoved" value="Odpověď na otázku  
  <%= kviz.getInt("id_kvizu")%>" >  
</form>
```

```
<%  
  }  
  }  
<%>
```

## 6 EXTERNÍ KNIHOVNY A TEMPLATY

### 6.1 mail.jar

Knihovna mail.jar zahrnuje úplné rozhraní JavaMail API, implementace a poskytuje protokoly: IMAP, SMTP a POP3. Tato knihovna je v systému velice důležitá, bez ní by nebylo možné z aplikace posílat emaily.

### 6.2 activation.jar

Díky této knihovně mohou vývojáři využít standardních služeb k určení libovolného typu dat, zapouzdřit přístup k nim, zjistit dostupné operace a vytvořit instanci příslušné beanu k provedení uvedené operace.

### 6.3 xssflt.jar

Tuto knihovnu jsem do své práce zahrnul z důvodů zvýšení bezpečnosti celé aplikace. Tato knihovna poskytuje filtr, kterým prochází jakýkoliv požadavek aplikace. Tento filtr zabráňuje Cross Site Scripting (XSS) útokům.

### 6.4 Twilio-7.9.1 -jar -with-dependencies.jar

Tato knihovna slouží pro posílání SMS zpráv, přes rozhraní Twilio. Tato knihovna obsahuje metody pro vytvoření a odeslání zprávy ve tvaru, který očekává služba Twilio. Při programování této služby, jsem se řídil návodem od společnosti Twilio.

### 6.5 mysql-connector-java-8.0.16.jar

Knihovna určená pro připojení k MySQL databázovému serveru. Poskytuje nástroje pro stabilnější a bezpečnější připojení k různým verzím databázových serverů.

## 6.6 Externí template

V mé diplomové práci jsem použil template (šablony) favison a eiser. Byly použity v částech registrace, přihlášení a úprava přihlašovacích údajů. Z těchto template jsem vybral soubory a části kódu, který jsem pak použil ve své diplomové práci.

Pro zlepšení vzhledu stránky jsem do své diplomové práce zahrnul templateové css soubory:

Bootstrap.css

Main.css

Responsive.css

Style.css

StyleReg.css

Themify-icons.css

Util.css

Těchto stylů jsem využil při vytváření těchto stránek aplikace:

Header.jsp

Footer.jsp

Registrace.jsp

Login.jsp

Zmena\_udaju.jsp

Kód těchto stránek je derivací kódu z templateů. Nejedná se o úplnou kopii kódu, ale kód těchto stránek je založen na kódu poskytnutém v templateu.

## ZÁVĚR

Diplomová práce dle mého názoru splnila cíle, jež jsem si vytýčil v úvodu, a odpovídá požadavkům, které na ni byly kladeny. Svým uživatelům tato aplikace jistě prospěje ke zpřehlednění a usnadnění každodenní práce, usnadní jim možnosti komunikace a uživatelé pomocí ní budou schopni dohlížet na činnosti, které se ve firmě aktuálně odehrávají.

V některých částech práce jsem se rozhodl využít možnosti šablon pro vylepšení celkového vzhledu aplikace, z tohoto důvodu jsem musel do výsledné aplikace zahrnout některé externí styly a části kódu. Jedná se však pouze o stránky pro registraci, přihlášení, záhlaví a zápatí. Tyto části prošly určitou modifikací, aby korespondovaly s obsahem práce, nicméně jejich základem je šablona. Dále aplikace vyžadovala použití externích knihoven, pro odeslání zpráv a zvýšení bezpečnosti.

## POUŽITÁ LITERATURA

- [1] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- [2] BURD, Barry A. *JSP: JavaServer Pages: podrobný průvodce*. Praha: Computer Press, 2003. ISBN 80-7226-804-X.
- [3] REDKO, Andriy. *Advanced Java: Prepering you for for Java Mastery*. Exelixis Media P.C, 2015.
- [4] LEMAY, Laura a Charles L. PERKINS. *Teach yourself Java in 21 days*. 1. vyd. INDIANA, Indianapolis: Sams.net, 1996. ISBN 1-57521-030-4.
- [5] ZAMBON, Giulio. *Beginning JSP, JSF and Tomcat*. 1. vyd. NEW YORK, New York: Springer Science+Business Media New York, 2012. ISBN 978-1-4302-4624-4.
- [6] BASHAM, Brian, Kathy SIERRA a Bert BATES. *Head first servlets and jsp*. 2. verze. Sebastopol: O'Reilly Media, Inc., 2008. ISBN 978-0-596-51668-0
- [7] KOSEK, Jiří. *HTML: tvorba dokonalých www stránek: podrobný průvodce*. Praha: Grada, 1998. Průvodce (Grada). ISBN 978-80-7169-608-7.
- [8] MURPHY, Hertha A, Herbert W HILDEBRAND a Jane P THOMAS, *Effective Businnes Communication*, Delhi: Prentice-Hall of India, 2004. ISBN: 9788120317093
- [9] Business process management discussions, news and articles | ARIS BPM Community [online]. Copyright © 2009 [cit. 15.05.2020]. Dostupné z: <https://www.ariscommunity.com/>
- [10] W3Schools Online Web Tutorials [online]. Refsnes Data: Copyright © 1999-2020 [cit. 15.5.2020]. Dostupné z: <https://www.w3schools.com/>
- [11] Stack Overflow – Where Developers Learn, Share, & Build Careers [online]. Stack Exchange Inc: Copyright © 2020 [cit. 15.5.2020]. Dostupné z: <https://stackoverflow.com/>

## **PŘÍLOHY**

|                                       |    |
|---------------------------------------|----|
| Příloha A – Obsah přiloženého CD..... | 70 |
|---------------------------------------|----|

## **PŘÍLOHA A – OBSAH PŘILOŽENÉHO CD**

Na přiloženém CD se nachází kompletní programová implementace interního komunikačního systému.

Diagramy:

Obsahuje projekt Enterprise Architect, ve kterém jsou diagramy uloženy.

Projekt ArsiExpres obsahuje Diagram BPMN.

Aplikační část:

IKS\_Langer.zip (archiv se zdrojovými kódy aplikace)

Dokumentace:

Diplomova\_prace\_Langer.pdf (text diplomové práce v elektronické podobě)