

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ

Fakulta elektrotechnická

Katedra počítačů



Skill Matrix

Bakalářská práce

Vypracoval: Milan Němeček

Vedoucí práce: Ing. Božena Mannová, Ph.D.

Květen 2022

## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Němeček** Jméno: **Milan** Osobní číslo: **483693**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra počítačů**  
Studijní program: **Otevřená informatika**  
Specializace: **Software**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Aplikace Skill Matrix**

Název bakalářské práce anglicky:

**Skill Matrix application**

Pokyny pro vypracování:

Pro účely společnosti zaměřené na vývoj software, která má stovky pracovníků s velmi širokou škálou znalostí v IT, naprogramujte webovou aplikaci, ve které bude možné uchovávat informace o znalostech pracovníků.

- Přihlášení do aplikace bude možné pomocí Active Directory.
  - Jednotlivé znalosti budou rozdělené do kategorií a podkategorií.
  - Pro uchování dat použijte SQL databázi.
  - Data bude možno exportovat do .csv formátu.
  - Aplikace bude mít vhodně řešené logování chybových i provozních stavů
- Aplikaci navrhňte, implementujte a otestujte. Použijte vhodných prostředků SE.

Seznam doporučené literatury:

- [1] Pressmann R. S.: Software Engineering,
- [2] <https://www.analyticsinhr.com/blog/create-skills-matrix-competency-matrix/>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Božena Mannová, Ph.D., kabinet výuky informatiky FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **04.03.2021**

Termín odevzdání bakalářské práce: \_\_\_\_\_

Platnost zadání bakalářské práce: **19.02.2023**

Ing. Božena Mannová, Ph.D.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta

## Prohlášení

„Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.“

Němeček Milan

## Poděkování:

Chtěl bych poděkovat vedoucí mého projektu Ing. Boženě Mannové Ph.D. za její ochotu a pomoc na konzultacích týkající se tvorby této bakalářské práce.

Dále bych rád poděkoval i Ing. Jiřímu Šebkovi za jeho rady a připomínky při mé práci na této bakalářské práci.

## Abstrakt

Tématem mé bakalářské práce je webová aplikace Skill matrix. Jedná se o nástroj pro mapování dovedností/schopností jednotlivce, či třeba zaměstnanců firmy. V rámci této práce se zabývám analýzou problému, návrhem a implementací aplikace splňující stanovené specifikace. Součástí této práce je i výběr vhodných nástrojů a technologií pro splnění zadání. Pro implementaci aplikace jsem zvolil monolitickou architekturu klient-server, kde klientská část byla vytvořena ve frameworku Angular a serverová část ve frameworku Spring Framework. V rámci analýzy jsem v souladu se zadáním práce vytvořil seznam požadavků na funkcionality, které má výsledná aplikace uživateli poskytovat. Dané funkcionality jsem implementoval a otestoval vhodnými prostředky.

**Klíčová slova:** skill matrix, webová aplikace, klient-server, Active Directory, REST – API, SQL databáze

The topic of my bachelor thesis is web application Skill matrix. It is a tool for mapping skills/abilities of an individual or maybe employees of a company. Within this thesis I deal with the analysis of the problem, the design and implementation of an application that meets the specified specifications. Part of this work is the selection of suitable tools and technologies to fulfill the assignment. For the implementation of the application I chose programming model client-server, where the client part was created in Angular framework and the server part in Spring Framework. Within the analysis in accordance to assignment I created a list of requirements for functionalities, which should the application provide to user. I implemented and tested the functionalities with appropriate tools.

**Key words:** skill matrix, web application, client-server, Active Directory, REST – API, SQL database

## Obsah

1. Úvod.....	8
1.1. Co je Skill matrix.....	8
1.2. Motivace.....	8
1.3. Možnosti řešení.....	8
1.3.1. Google Sheets.....	8
1.3.2. Webová aplikace .....	9
2. Analýza požadavků.....	10
2.1. Specifikace zadání.....	10
2.2. Sběr požadavků.....	10
2.3. Shrnutí rolí uživatelů.....	12
2.4. Poznámka k návrhu:.....	14
2.5. Doménový model.....	14
3. Implementace.....	16
3.1. Možnosti implementace a porovnání.....	16
3.1.1. Google Sheets.....	16
3.1.2. Webová aplikace .....	17
3.1.3. Zvolené řešení.....	17
3.2. Architektura aplikace.....	18
3.2.1. Klient.....	19
3.2.2. Serverová část.....	20
3.2.3. Komunikace klient – server.....	20
3.2.4. Databáze.....	21
4. Testování.....	21
4.1. Unit testy.....	21
4.2. Výsledky.....	21
5. Rozšíření aplikace.....	22
Závěr.....	22
Grafické uživatelské rozhraní.....	23
Použité zdroje.....	25

## Seznam obrázků

Obr. č.1: Ukázka skill matrix v goosle.docs - <a href="https://www.valamis.com/documents/10197/878064/skill-matrix-template.png">https://www.valamis.com/documents/10197/878064/skill-matrix-template.png</a> .....	9
Obr č.2: UML diagram business domain modelu.....	14
Obr č.3: UML diagram business domain modelu.....	15
Obr. č.4: Ukázka skill matrix v goosle.docs - <a href="https://www.valamis.com/documents/10197/878064/skill-matrix-template.png">https://www.valamis.com/documents/10197/878064/skill-matrix-template.png</a> .....	16
Obr. č.5: Ukázka výsledné aplikace skill matrix.....	17
Obr. č.6: Diagram komponent – zobrazení 3 hlavních vrstev.....	18
Obr. č.7: Dialogové okno pro přihlášení.....	19
Obr. č. 8: ukázka GUI aplikace.....	23
Obr. č. 9: ukázka zdrojového kódu serverové části.....	24
Obr. č. 10: ukázka kódu klientské části.....	24

## 1. Úvod

Pro upřesnění významu používaných termínů v této práci na začátku uvedu pár často používaných termínů a jejich vzájemný vztah. Slova *schopnost*, *dovednost*, *znalost* používám v této práci jako synonyma. Slovem *odbornost* v dovednosti/znalosti mýním úroveň kterou daný člověk disponuje v dané dovednosti/znalosti.

### 1.1. Co je Skill matrix

Skill matrix je nástroj pro mapování dovedností/schopností, které jsou k dispozici nebo jsou vyžadované pro nějaký projekt či pracovní tým. Umožňuje ukládat, aktualizovat a vizualizovat dostupné a potřebné dovednosti členů týmu. Pomáhá odhalovat nedostatky znalostí v dané oblasti/oboru a zároveň zefektivňuje plánování práce na nových projektech.[1]

V praxi se může jednat o excelovskou nebo Google Sheets tabulku obsahující data o jednotlivých zaměstnancích. V rámci této práce jsem se zabýval analýzou problému, návrhem aplikace a samotnou implementací požadavků vytvořené v souladu se specifikacemi uvedené v zadání práce.

### 1.2. Motivace

S velkým počtem zaměstnanců s různými úrovněmi dovedností roste náročnost plánování projektů. Skill matrix je aplikace, která umožňuje přehledně uchovávat, aktualizovat a zobrazovat prostředky, které má firma k dispozici. Díky skill matrix lze lépe plánovat práci na projektech, detekovat nedostatky jednotlivců, týmů, ale zároveň umožňuje jednotlivým zaměstnancům vyjádřit zájem o určitou dovednost, ve které by se chtěli zlepšit.

Aplikace by měla pomoci firmám usnadnit plánování nových projektů, detekování nedostatku odborníků v určité dovednosti a zároveň motivovat zaměstnance ke zlepšování svých schopností. (Lze ověřit pomocí dotazníku/ankety a porovnání stavu firmy na začátku nasazení Skill matrix a rok poté.)

### 1.3. Možnosti řešení

#### 1.3.1. Google Sheets

([www.valamis.com/hub/skills-matrix](http://www.valamis.com/hub/skills-matrix))[1]<sup>1</sup>

Veškerá data jsou uchovávána a zobrazena v tabulkách soboru typu. xcl. Náhled tohoto řešení je na Obr. č.1. Na obrázku z náhledu, jak takový soubor může vypadat, vidíme, že odbornost v dovednostech zaměstnance je popsána pouze jako číslo. Pro větší detail, jako datum poslední změny, vyjádření zájmu o danou dovednost či třeba poznámku nadřízeného by bylo zapotřebí buď pro každou dovednost přiřadit více sloupců nebo vytvořit pro tato další data nové tabulky. V obou případech by tím se tím snížila přehlednost pro uživatele.

Přístup k dokumentům by byl omezen pouze pro zaměstnance firmy. Google Sheets umožňuje udělovat 4 úrovně oprávnění: vlastnictví, editování, komentování, náhled. Ale





## 2. Analýza požadavků

Analýza požadavků byla provedena v souladu se specifikacemi ze zadání práce. Výsledkem je seznam požadavků. Požadavky obsahují informace o požadované funkcionalitě, informace o zadavateli a odůvodnění požadavku.

### 2.1. Specifikace zadání

Pro účely společnosti zaměřené na vývoj software, která má stovky pracovníků s velmi širokou škálou znalostí v IT, naprogramujte aplikaci, ve které bude možné uchovávat informace o znalostech pracovníků.

- Aplikaci vytvořte tak, aby byla přístupná z webového rozhraní
- Přihlášení do aplikace bude možné pomocí Active Directory
- Aplikace bude obsahovat několik rolí uživatelů od administrátorské úrovně až po běžné
- pracovníky, kteří upravují pouze svůj profil.
- Jednotlivé znalosti budou rozdělené do kategorií a podkategorií
- Pro uchování dat bude použita SQL databáze
- Dojde k oddělení databáze od prezentační vrstvy pomocí REST-API rozhraní
- Data půjde exportovat do .csv formátu
- Aplikace bude mít vhodně řešené logování chybových i provozních stavů.

### 2.2. Sběr požadavků

Jelikož zadání práce bylo příliš obecné a neexistovala firma v roli zadavatele, tak sběr požadavků byl proveden v rámci konzultace s vedoucím bakalářské práce a analýze předchozích řešení.

#### **BRQ001 – Přístup přes webový prohlížeč**

Jako uživatel potřebuji, aby byla aplikace přístupná z webového rozhraní, abych k ní měl snadný přístup.

#### **BRQ002 – Přihlášení**

Jako uživatel potřebuji, abych se mohl do aplikace přihlašovat pomocí Active Directory.

**BRQ003 – Více rolí**

Jako majitel aplikace potřebuji, aby aplikace obsahovala několik rolí uživatelů od administrátorské úrovně až po běžné pracovníky, aby každá z rolí mohla mít své specifické pravomoce.

(Role :

1. Administrátor
2. Manažer
3. Vedoucí týmu
4. (Běžný) pracovník
5. Nepřihlášený uživatel

)

**BRQ004 – Kategorie a podkategorie**

Jako uživatel potřebuji mít možnost rozdělovat znalosti do kategorií a podkategorií, pro větší přehlednost a praktičnost při používání aplikace.

**BRQ005 – Úprava profilu**

Jako uživatel potřebuji, abych měl možnost upravit svůj profil, abych mohl včas informovat o změně svých osobních dat.

**BRQ006 – Úrovně znalostí**

Jako administrátor potřebuji, aby bylo u dané dovednosti daného člověka uvedená úroveň znalosti a zájem v dané oblasti, abych mohl jasně rozlišit dovednosti pracovníků.

(Úrovně znalostí:

1. Naprosto žádná zkušenost (Uživatel nemá žádnou zkušenost v dané dovednosti)
2. Základní znalosti (Uživatel dokáže s pomocí aplikovat danou dovednost)
3. Střední úroveň znalosti (Uživatel je naprosto samostatný v aplikování dané dovednosti)
4. Vysoká úroveň znalosti (Uživatel je expertem v dané dovednosti, umí pomáhat druhým)

)

**BRQ007 – Export dat**

Jako manažer potřebuji mít možnost exportovat vybraná data z databáze do .csv formátu pro další zpracování pro konkrétní potřeby.

**BRQ008 – Update**

Jako manažer potřebuji, aby zaměstnanci měli možnost požádat změny úrovně dovednosti svého nadřízeného, aby nemohli provádět změny individuálně.

**BRQ009 – Úprava informací o znalostech**

Jako manažer potřebuji, aby si uživatelé nemohli sami upravovat informace o svých znalostech, aby nemohli falšovat údaje o svých znalostech.

**BRQ010 – Historie úprav**

Jako manažer potřebuji mít možnost vygenerovat tabulku obsahující záznam o provedených změnách (přidání pracovníka, přidání nové znalosti pracovníka, změna úrovně znalosti pracovníka) za měsíc, 6 měsíců a rok, abych věděl, kde došlo ke změnám a mohl je dál prezentovat (svým nadřízeným).

**BRQ011 – Vytváření projektů s požadavky**

Jako manažer potřebuji mít možnost vytvořit projekt se svými požadavky na potřebné dovednosti, abych mohl porovnávat prostředky (zaměstnance), které mám k dispozici, a které potřebuji.

**BRQ012 – Vyjádření zájmu**

Jako pracovník potřebuji mít možnost upravit svůj zájem o danou dovednost, aby můj nadřízený věděl, jakým směrem bych rád rozvíjel své schopnosti.

(Úrovně zájmu o danou dovednost:

1. Ano (Uživatel má zájem se rozvíjet v dané dovednosti.)
  2. Ne (Uživatel nemá zájem se rozvíjet v dané dovednosti.)
- )

**BRQ013 – Práva administrátora**

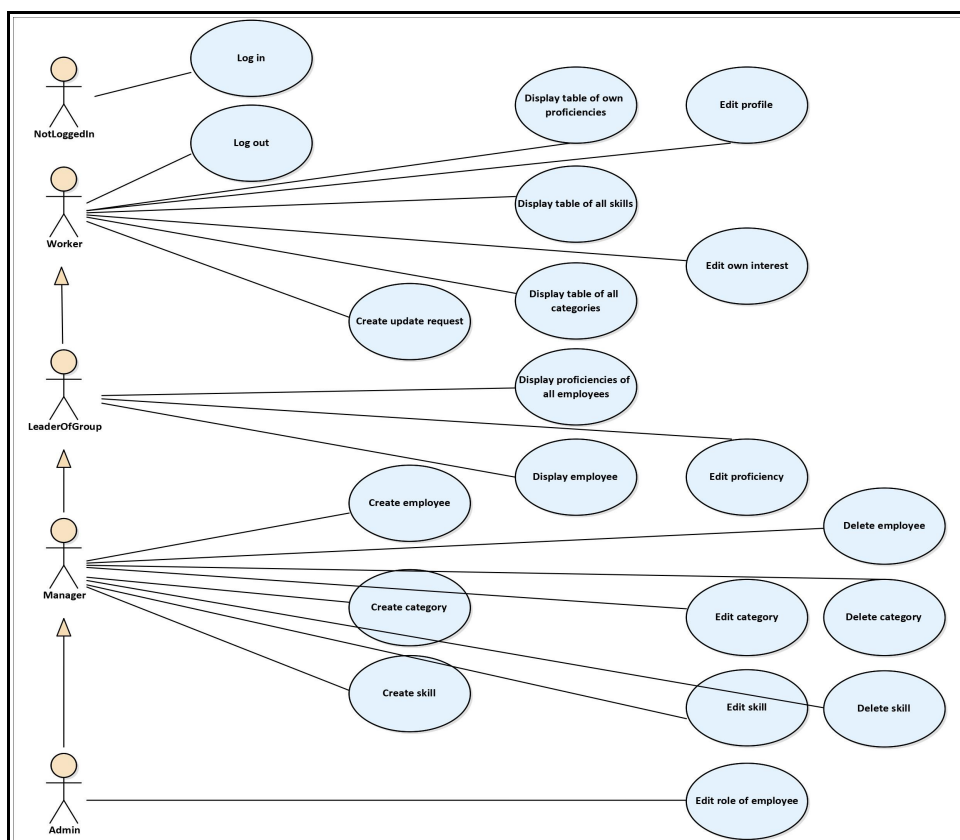
Jako administrátor potřebuji mít přístup ke všem informacím mít práva ke přihlašovacích údajů, abych měl kontrolu nad tím, kdo se smí do systému připojit a kdo ne.

**2.3. Shrnutí rolí uživatelů**

Zde je uveden seznam akcí možných pro jednotlivé typy uživatelů. Přehled případů užití je zobrazen na Obr č.2.

- Nepřihlášený uživatel
  - Přihlásit se do systému
- Pracovník
  - Upravit vlastní profil
  - Zažádat o úpravu informací o svých dovednostech
  - Zobrazit tabulku vlastních dovedností
  - Zobrazit tabulku všech dovedností a kategorií
  - Vyjádřit zájem/nezájem pro danou dovednost

- Vedoucí týmu
  - Vše, co může pracovník
  - Zobrazit tabulku dovedností uživatele
  - Upravit odbornost uživatele
- Manažer
  - Vše, co může vedoucí týmu
  - Vytvořit/odstranit pracovníka
  - Vytvořit/odstranit/upravit kategorii dovedností
  - Vytvořit/odstranit/upravit dovednost
- Administrátor
  - Vše, co může manažer
  - Přiřadit novou roli uživateli



Obr. č.2: UML diagram business domain modelu

## 2.4. Poznámka k návrhu:

Zvažoval jsem přidat funkcionalitu umožňující přiřazení uživatelů k projektům včetně vedoucích projektu. Po konzultaci s vedoucí mé bakalářské práce jsem nakonec usoudil, že by se jednalo o rozšíření mimo rámec účelu skill matrix. Tato funkcionalita byla přidána do možných rozšíření aplikace.

Některé funkcionality nebyly kvůli časové náročnosti implementovány. Jedná se o požadavky: **BRQ010 – Historie úprav**

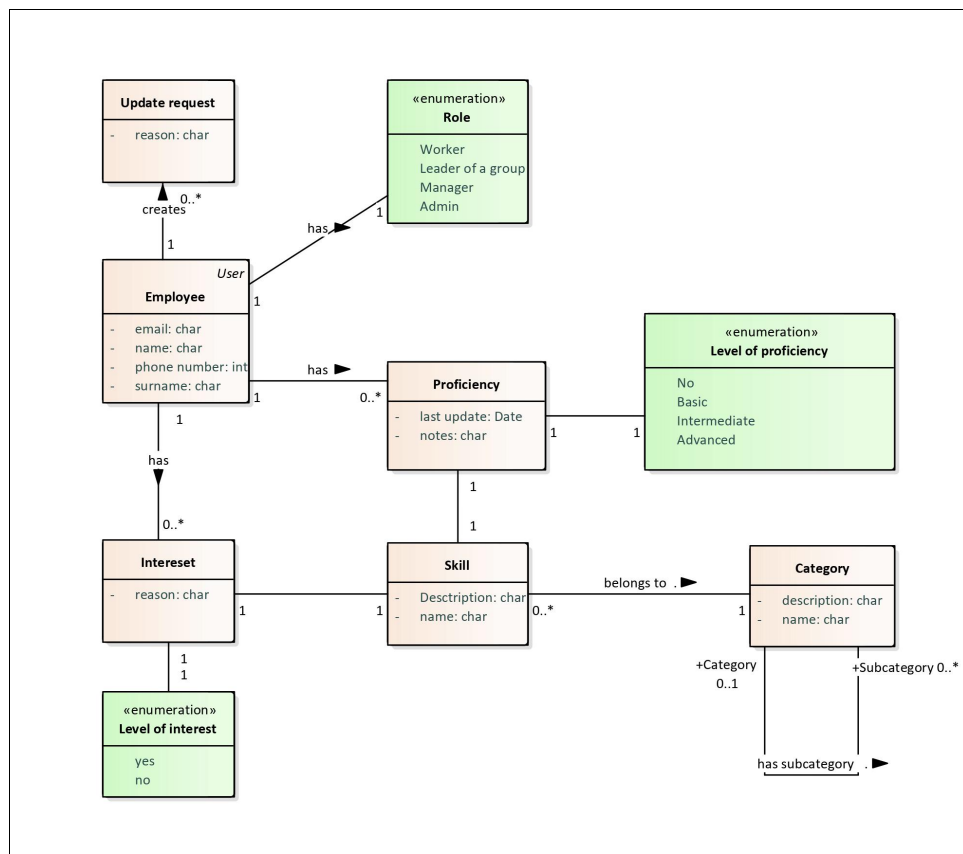
**BRQ011 – Vytváření projektů s požadavky**

Tyto funkcionality jsou uvedeny v možných rozšířeních, vylepšeních pro tuto aplikaci.

## 2.5. Doménový model

Na Obr. č.3 můžeme vidět UML diagram všech základních entit a enumerací, které byly použity pro ukládání potřebných dat do databáze.

Jak můžete vidět z obrázku č.1, tak názvy pro třídy, entity jsou v angličtině. Angličtina byla použita i při psaní samotného kódu, komentářů, názvů funkcí. Proto pro přehlednost pro anglické názvy tříd z obrázku uvedu překlad.



Obr. č.3: UML diagram business domain modelu

**Update request** – žádost o aktualizaci informací

**Role** – role (**worker** – běžný pracovník, **leader of a group** – vedoucí(týmu), **manager** – manažer, **admin** – administrátor)

**Employee** – zaměstnanec/uživatel

**Proficiency** – odbornost

**Level of proficiency** – úroveň odbornosti

(**no** – žádná, **basic** – základní, **intermediate** – střední, **advanced** - pokročilá)

**Skill** – dovednost/schopnost

**Interest** – zájem

**Level of interest** – úroveň zájmu

**Category** – kategorie

### 3. Implementace

Tato část je věnována popisu vybraní řešení a vhodných prostředků pro implementování navržených požadavků.

#### 3.1. Možnosti implementace a porovnání

##### 3.1.1. Google Sheets

([www.valamis.com/hub/skills-matrix](http://www.valamis.com/hub/skills-matrix))[1]<sup>3</sup>

Veškerá data jsou uchovávána a zobrazena v tabulkách soboru typu .xcl. Náhled tohoto řešení je na Obr. č.4. Na obrázku z náhledu, jak takový soubor může vypadat, vidíme, že odbornost v dovednostech zaměstnance je popsána pouze jako číslo. Pro větší detail, jako datum poslední změny, vyjádření zájmu o danou dovednost či třeba poznámku nadřizného by bylo zapotřebí buď pro každou dovednost přiřadit více sloupců nebo vytvořit pro tato další data nové tabulky. V obou případech by tím se tím snížila přehlednost pro uživatele.

Přístup k dokumentům by byl omezen pouze pro zaměstnance firmy. Google Sheets umožňuje udělovat 4 úrovně oprávnění: vlastnictví, editování, komentování, náhled. Ale neumožňuje udělovat pravomoce přidávat, upravovat, zobrazovat či mazat specifická data pro různé typy uživatelů.

- Výhody:** Levné a relativně rychlé řešení
- Nevýhody:** Problémy s přidáváním a zobrazováním nových dat.  
Některé funkcionality a případná rozšíření by nebyli možné implementovat.
- Závěr:** Vhodné pro osobní účely jednotlivce, ne však pro firmu s desítkami či stovkami zaměstnanců.

Skills Matrix																	
Department	0 Awareness																
Total Number of Staff Members	1 Novice																
Total Number of skills / competences	2 Professional																
	3 Expert																
	4 Leading-edge expert																
	Skill or competence	Skill or competence	Skill or competence	Skill or competence	Skill or competence	Skill or competence	Skill or competence	Skill or competence	Skill or competence	Skill or competence	Skill or competence	Skill or competence	Skill or competence	Skill or competence	Skill or competence	Skill or competence	Proficiency level
John Smith	0	4	2	3	4	4	2	3	1	2	1	4	2	4	2	4	64.7%
Staff Member Name	0	1	2	3	4	4	2	4	0	4	0	4	0	0	0	0	79.6%
Staff Member Name	0	1	2	3	4	4	1	3	0	0	0	0	0	0	0	0	21.9%
Staff Member Name	0	1	2	3	4	4	2	3	0	0	0	0	0	0	0	0	23.4%
Staff Member Name	0	1	2	3	4	4	2	3	0	0	0	0	0	0	0	0	25.0%
Staff Member Name	0	1	2	3	4	4	1	3	0	0	0	0	0	0	0	0	21.9%
Staff Member Name	0	1	2	3	4	4	2	3	0	0	0	0	0	0	0	0	25.0%
Staff Member Name	0	1	2	3	4	4	0	3	0	0	0	0	0	0	0	0	23.4%
Staff Member Name	0	1	2	3	4	4	2	4	0	0	0	0	0	0	0	0	25.0%
Staff Member Name	0	1	2	3	4	4	1	3	0	0	0	0	0	0	0	0	21.9%
Staff Member Name	0	1	2	3	4	4	0	3	0	0	0	0	0	0	0	0	25.0%
Staff Member Name	0	1	2	3	4	4	2	3	0	0	0	0	0	0	0	0	18.8%
Staff Member Name	0	1	2	3	4	4	0	4	0	0	0	0	0	0	0	0	28.6%
Staff Member Name	0	1	2	3	4	4	1	3	0	0	0	0	0	0	0	0	21.9%
Staff Member Name	0	1	2	3	4	4	2	3	0	0	0	0	0	0	0	0	23.4%
Staff Member Name	0	1	2	3	4	4	0	3	0	0	0	0	0	0	0	0	25.0%
Staff Member Name	0	1	2	3	4	4	1	3	0	0	0	0	0	0	0	0	21.9%
Staff Member Name	0	1	2	3	4	4	2	4	0	0	0	0	0	0	0	0	28.1%
Staff Member Name	0	1	2	3	4	4	0	3	0	0	0	0	0	0	0	0	25.0%
Percent of skill coverage	0.0%	25.0%	50.0%	75.0%	100.0%	46.4%	77.4%	8.3%	4.8%	7.1%	4.8%	6.0%	7.1%	7.1%	9.5%	7.1%	

Obr. č.4: Ukázka skill matrix v goosle.docs -

<https://www.valamis.com/documents/10197/878064/skill-matrix-template.png>



### 3.1.2. Webová aplikace

#### (APLIKACE SKILL MATRIX - Zika Adam)[2]<sup>4</sup>

Tato aplikace byla vyvinuta v rámci bakalářské práce jako webová aplikace. Umožňovala mimo jiné přiřadit uživatelům specifické role, od kterých se odvíjeli jejich pravomoce. Při implementaci byly použity preferovaně open-source prostředky, aby se předešlo placení licencí.

- Výhody: Při implementování skill matrix jako webovou aplikaci je možné vymyslet a přidat funkcionality, které Google Sheets nenabízí.
- Nevýhody: Časově náročnější řešení implementace

### 3.1.3. Zvolené řešení

Pro splnění všech požadavků ze zadání práce bylo potřeba vyvinout vlastní webovou aplikaci. Ukázka GUI je na Obr. č.5.

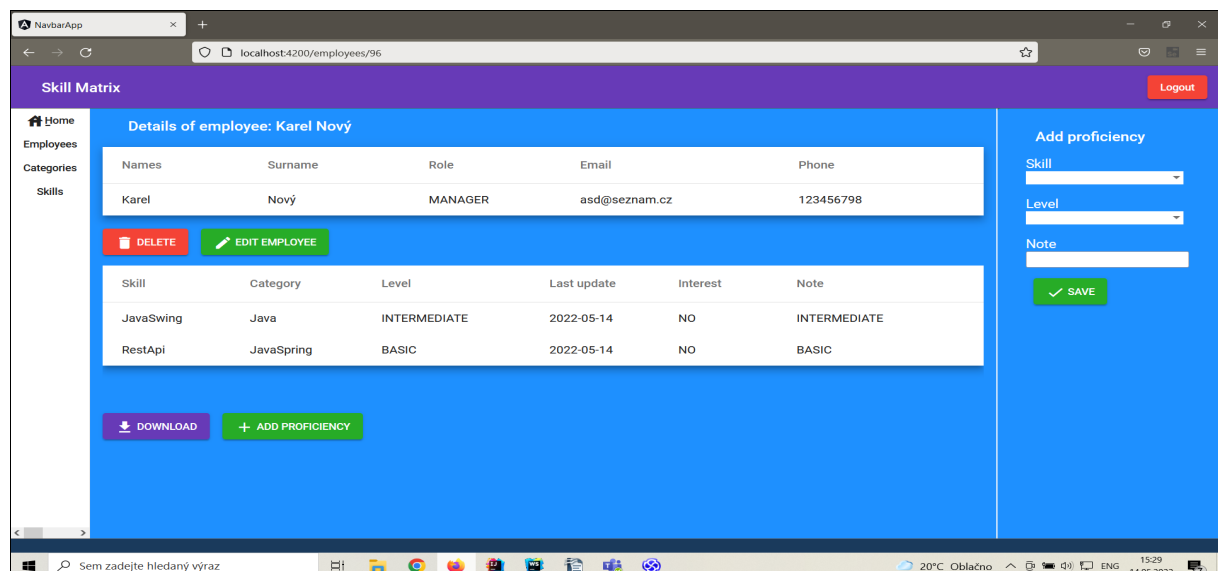
Vylepšení oproti předchozímu řešení:

Data je možné exportovat do .csv formátu

Kategorie je možné vrstvit do více úrovní podkategorií.

Uživatel má možnost upravit svůj zájem o danou dovednost.

Přidání uživatelské role: Manažer



Obr. č.5: Ukázka výsledné aplikace skill matrix

## Upravování osobních informací o znalostech/schopnostech

V předchozím řešení bylo dovoleno standardnímu uživateli (pracovník) upravovat své osobní informace o svých znalostech. To bylo ve specifikaci zakázáno. Tuto úpravu může provádět pouze manažer nebo administrátor a to buď jako reakce na žádost uživatele, a nebo samostatně, třeba kvůli pravidelné aktualizaci informací.

Při návrhu aplikace jsem se musel rozhodnout, které technologie, frameworky, programovací jazyky použiju pro její implementaci. Rozhodoval jsem se na základě těchto kritérií:

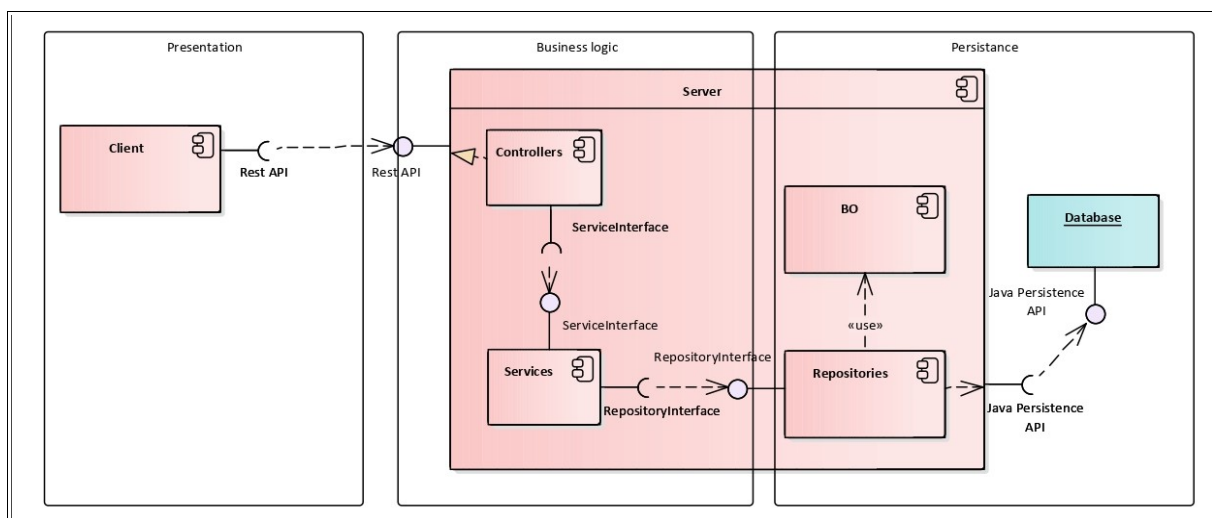
1. Specifikace zadání
2. Vlastní obeznámenost s danou technologií
3. Licenční náklady

Při výběru jsem se především řídil specifikacemi ze zadání. Pokud nebyla specifikace pro užití konkrétní technologie uvedena, tak jsem dával přednost technologiím, se kterými se obeznámen, měl s nimi už nějaké zkušenosti, a které nevyžadují placenou licenci.

### 3.2. Architektura aplikace

Pro naprogramování aplikace jsem zvolil architekturu klient-server (monolitická). Ta se skládá ze 3 hlavních vrstev: prezenční vrstva (klient), business logic vrstva a perzistentní vrstva (server a databáze). Každá z vrstev má svůj specifický účel a komunikují navzájem pomocí rozhraní. Tyto 3 hlavní vrstvy lze zobrazit v diagramu komponent na obrázku č.4.

Prezentační vrstva zajišťuje zobrazení dat a komunikaci uživatele s aplikací přes GUI – grafické uživatelské rozhraní. Business logic vrstva zpracovává požadavky prezenční



Obr. č.6: Diagram komponent – zobrazení 3 hlavních vrstev

vrstvy a vyžádaná data z perzistentní vrstvy. Perzistentní vrstva má nastarosti uchovávání a poskytování dat uchovávaných v databázi.

### 3.2.1. Klient

Pro implementaci klientské části aplikace jsem se rozhodl použít jazyk TypeScript a konkrétně framework Angular. Další možností byl framework React, ale dal jsem přednost Angularu, který se více hodí pro menší webové aplikace a umožňuje jednodušší implementaci obousměrné provázanosti dat.[3]<sup>5</sup>

#### TypeScript

TypeScript je open-source programovací jazyk. Je nadstavbou jazyku JavaScript, známý hlavně jako skriptovací jazyk pro webové stránky, ale používá se i v jiných oblastech vývoje softwaru. JavaScript rozšiřuje statické typování.[4]<sup>6</sup>

#### Angular

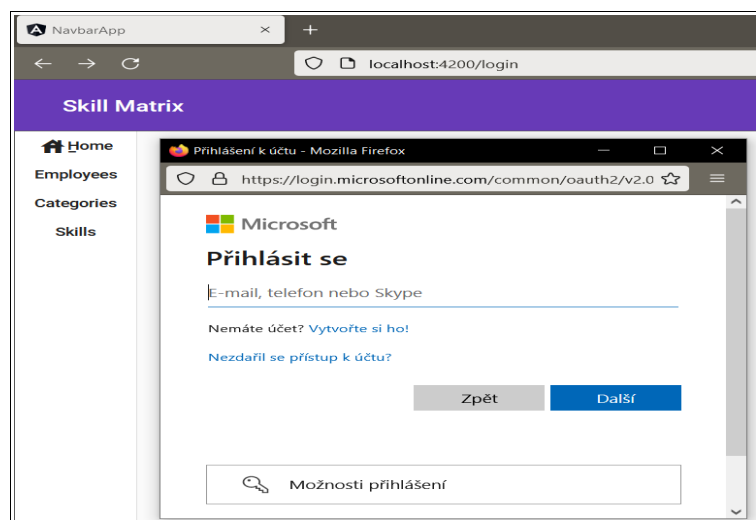
AngularJS je open-source front-end webový framework založený na jazyku JavaScript. Zaměřuje se především na vývoj webových aplikací používající architekturu MVC – model-view-controller.[5]<sup>7</sup>

#### React

React je JavaScript knihovna pro vytváření uživatelského rozhraní. React nepoužívá žádné šablony a veškerá jeho komponentová logika je napsána v JavaScriptu, což mu dává výhodu větší flexibility a možnosti zpracování velkého množství dat v naší aplikaci. Naše aplikace ale nebude zpracovávat velké množství dat a jedná se o menší aplikaci. Proto jsem zvolil Angular.[6]<sup>8</sup>

#### Active Directory

Pro přihlašování na klientskou část jsem použil Azure AD(active directory)[7]<sup>9</sup>. Jedná se o univerzální platformu pro správu a zabezpečení identit. Ke klientovi se tedy lze připojit přes jakýkoliv microsoft účet. Uživatel se přihlásí přes svůj microsoft účet v novém dialogovém okně ukázaném na obrázku č.7 a aplikace



Obr. č.7: Dialogové okno pro přihlášení

zkontroluje zda je daný uživatel zavedený v databázi. Pokud ano, je mu udělen přístup k datům a funkcím aplikace.

### 3.2.2. Serverová část

#### Programovací jazyk

Pro serverovou část aplikace jsem z nepoužívanějších programovacích jazyků jako Java, Python nebo C++ vybral jazyk Java, protože s ním mám nejvíc zkušeností v porovnání s ostatními. Pro vývoj aplikace jsem zvolil OpenJDK oproti OracleJDK, protože OracleJDK vyžaduje placenou licenci.[8]<sup>10</sup> [9]<sup>11</sup>

JDK neboli Java development kit je jedna ze tří nepostradatelných technologických balíčků používaných pro programování v Javě. JDK obsahuje kompilátor a základní knihovny pro tvorbu a build kódu.[10]<sup>12</sup>

Java je objektově zaměřený programovací jazyk vytvořený tak, aby v něm napsané zkompileované kódy spustitelné na všech platformách podporující jazyk Java bez potřeby znovu zkompileovat kód. Java aplikace se většinou kompilují do bytekódu spustitelný na JVM neboli „Java virtual machine“. Svou skladbou (syntax) je velmi podobný jazykům C, C++ a C#. A mimo jiné se jedná o velmi oblíbený jazyk a často používaný pro programování klient – server aplikací.[11]<sup>13</sup>

#### Nástroj pro build a řízení aplikace

Zde jsem měl možnost vybírat mezi nástroji: Maven, Gradle, Ant. Rozhodl jsem se pro Maven, se kterým mám nejvíce zkušeností. Gradle má užitečné vlastnosti jako použití substitučních pravidel pro kompatibilní knihovny nebo schopnost dynamicky nahrazovat dependence projektu s vnějšími a naopak. Ant je sice flexibilnější, protože nemá formalizované konvence, ale musí se psát vždy od začátku pro každý program zvlášť. Pro naše účely nám bude stačit Maven.[12]<sup>14</sup>

Maven je nástroj pro automatizaci buildu používaný hlavně pro Java projekty. Maven se stará primárně o stránky při buildu aplikace a to její sestavení a její dependence/závislosti. [13]<sup>15</sup>

### 3.2.3. Komunikace klient – server

Pro komunikaci mezi klientem a serverem bylo použito REST-API rozhraní.

#### REST – API

Rest: Jedná se o architekturu aplikačního rozhraní navrženou pro distribuované prostředí. Za použití http požadavků vytváří, načítá, upravuje a maže data mezi dvěma nezávislými zdroji.

API je rozhraní spojující dvě aplikace (dvě části aplikace). Definuje způsob, obsah i formát dat při komunikaci mezi aplikacemi.[14]<sup>16</sup>

### 3.2.4. Databáze

Nedílnou součástí budoucí aplikace bude databáze uchovávající naše vložená data. Máme dva základní kategorie databází, lišící se použitím jazyků SQL a NoSQL.

#### SQL database

SQL database neboli „structured query language database“ je relační databáze s předdefinovaným schématem. Pro uchovávání dat používá tabulky a jejich předdefinovanou strukturu.[15]<sup>17</sup>

#### NoSQL database

NoSQL database je nerelační databáze s dynamickým schématem a nestrukturovanými daty. Pro uchovávání dat může používat dokumenty, grafy a další možnosti, díky kterým dokáže zpracovávat velké objemy dat rychleji než databáze používající SQL.[16]<sup>18</sup>

Pro relativně malé množství dat, pro výhody relačních databází a specifikacím ze zadání jsem zvolil SQL databázi a konkrétně open-source PostgreSQL.[17]<sup>19</sup>

## 4. Testování

Testování bylo provedeno pomocí unit testů.

### 4.1. Unit testy

Pro testování jednotlivých tříd/objektů v serverové části aplikace byly být použity Unit (jednotkové) testy sloužící k testování malých jednotek kódu (jedna metoda, jedna třída) bez závislosti na správném fungování zbytku kódu. K tomu jsem použil nástroj JUnit sloužící k provádění jednotkových testů v jazyce Java.[18]<sup>20</sup>

### 4.2. Výsledky

Testování odhalilo nedostatky, týkající se neoprávněného přístupu uživatelů s menším oprávněním k datům, a funkcím, ke kterým by podle specifikací neměli mít přístup. Tyto nedostatky byly opraveny.

## 5. Rozšíření aplikace

Aplikace by se dala rozšířit o funkcionality, které byly uvedeny již dříve. Nebyly implementovány především kvůli časové náročnosti.

### **BRQ008 – Update**

Jako manažer potřebuji, aby zaměstnanci měli možnost zažádat změny úrovně dovednosti svého nadřízeného, aby nemohli provádět změny individuálně.

### **BRQ010 – Historie úprav**

Pro aplikaci jako skill matrix, kde se průběžně mění informace v závislosti na čase se vyplatí mít přístup k historii změn. V případě skill matrix by se jednalo o historii úprav dat o dovednostech jednotlivých uživatelů aplikace.

### **BRQ011 – Vytváření projektů s požadavky**

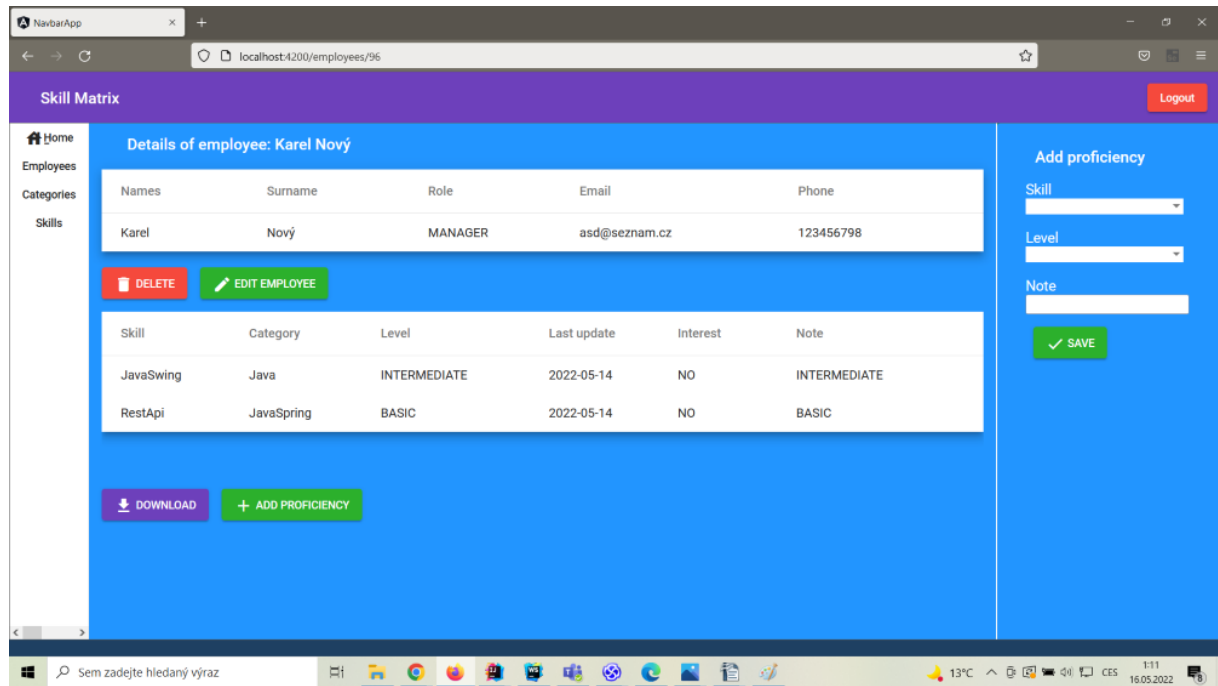
Další přidanou funkcionalitou by byla možnost vytvářet projekty s určitými požadavky na pracovníky s určitými schopnostmi. Tím se dosáhne jednoho z cílů skill matrix a to efektivnější plánování. Oprávnění uživatelé si mohou nechat zobrazit požadavky pro daný projekt a porovnat je s lidskými prostředky, které jsou k dispozici.

## **Závěr**

Provedl jsem analýzu zadaného problému. Do návrhu aplikace jsem zakomponoval všechny specifikace ze zadání bakalářské práce a vybral nástroje pro samotnou implementaci a testování aplikace. Aplikaci jsem implementoval, a otestoval.

## Ukázka aplikace

### Grafické uživatelské rozhraní



Obr. č. 8: ukázka GUI aplikace

### Ukázka kódu serverové části

```

23 import javax.validation.Valid;
24 import java.util.Date;
25 import java.util.LinkedHashMap;
26 import java.util.Map;
27 import java.util.stream.Collectors;
28
29 @RestController
30 @CrossOrigin(origins = "http://localhost:4200")
31 @RequestMapping("/categories")
32 public class CategoryServiceController {
33
34     @Qualifier("categoryServiceImpl")
35     CategoryService categoryService;
36
37     Mapper mapper;
38
39     @Autowired
40     public CategoryServiceController(CategoryService categoryService, Mapper map
41         this.categoryService = categoryService;
42         this.mapper = mapper;
43     }
44
45     @GetMapping
46     public ResponseEntity<Object> allCategories() {
47         return new ResponseEntity<>(categoryService.getAll().stream()
48             .map(mapper::categoryToDto).collect(Collectors.toList()), HttpSt
49     }
50
51     @GetMapping(value = "{/categoryId}")
52     public ResponseEntity<Object> getCategory(@PathVariable Long categoryId) {
53         Category retVal = categoryService.findById(categoryId);
54     }
55

```

Obr. č. 9: ukázka zdrojového kódu serverové části

## Ukázka kódu klientské části

```

1 import ...
2
3 @Component({
4     selector: 'app-add-edit-employee',
5     templateUrl: './add-edit-employee.component.html',
6     styleUrls: ['./add-edit-employee.component.css']
7 })
8 export class AddEditEmployeeComponent implements OnInit {
9
10
11     @Input()
12     title: string = "Add employee"
13
14     @Input()
15     employee?: Employee
16
17     @Output()
18     updateEmployees = new EventEmitter<string>();
19
20     roles: String[] = [];
21
22     checkoutForm = this.formBuilder.group( controlsConfig: {
23         name: '',
24         surname: '',
25         email: '',
26     });
27

```

Obr. č. 10: ukázka kódu klientské části



## Použité zdroje

[1] Skills Matrix. [www.valamis.com](http://www.valamis.com) [online]. [cit. 2022-01-04].

Dostupné z: <https://www.valamis.com/hub/skills-matrix>

[2] ZÍKA, Adam. Skill Matrix Application. Praha, 2020.

Dostupné také z: <https://dspace.cvut.cz/handle/10467/90116>. Bakalářská práce. ČVUT.

[3] VÝVOJ SINGLE-PAGE APLIKACE: ANGULAR JS VS. REACT. [Rascasone.com](http://Rascasone.com) [online]. [cit. 2022-01-04].

Dostupné z: <https://www.rascasone.com/cs/blog/angular-react-spa-web-aplikace>

Front-end JavaScript Frameworks Showdown: Vue vs. React vs. Angular. [Clockwise.software](http://Clockwise.software) [online]. [cit. 2022-01-04].

Dostupné z: <https://clockwise.software/blog/angular-vs-react-vs-vue/>

[4] TypeScript. [Wikipedia.org](http://Wikipedia.org) [online]. [cit. 2022-01-04].

Dostupné z: <https://cs.wikipedia.org/wiki/TypeScript>

[5] AngularJS. [Wikipedia.org](http://Wikipedia.org) [online]. [cit. 2022-01-04]. Dostupné z:

<https://en.wikipedia.org/wiki/AngularJS>

[6] React: A JavaScript library for building user interfaces. [Reactjs.org](http://Reactjs.org) [online].

[cit. 2022-01-04].

Dostupné z: <https://reactjs.org/>

[7] Mahajan, Rini, Manish Mahajan, and Dheerendra Singh. "Window azure Active Directory Services for Maintaining Security & Access Control.". [cit. 2022-05-16].

Dostupné z: <http://www.csjournals.com/IJITKM/PDF%2011-2/2.%20Rini.pdf>

Azure Active Directory. [Azure.microsoft.com](http://Azure.microsoft.com) [online]. [cit. 2022-01-04].

Dostupné z: <https://azure.microsoft.com/cs-cz/services/active-directory/#overview>

[8] Oracle. [Oracle.com](http://Oracle.com) [online]. [cit. 2022-01-04].

Dostupné z: <https://www.oracle.com/index.html>

[9] Temurin JDK 8. [Adoptium.net](http://Adoptium.net) [online]. [cit. 2022-01-04].

Dostupné z: <https://adoptium.net/>

[10] What is the JDK?: Introduction to the Java Development Kit. [Infoworld.com](http://Infoworld.com) [online]. [cit. 2022-01-04].

Dostupné z: <https://www.infoworld.com/article/3296360/what-is-the-jdk-introduction-to-the-java-development-kit.html>

[11] Java: (programming\_language). Wikipedia.org [online]. [cit. 2022-01-04].  
Dostupné z: [https://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

[12] Ant vs Maven vs Gradle. Baeldung.com [online]. [cit. 2022-01-04].  
Dostupné z: <https://www.baeldung.com/ant-maven-gradle>

[13] Apache Maven Project. Maven.apache.org [online]. [cit. 2022-01-04].  
Dostupné z: <https://maven.apache.org/>

Apache Maven. Wikipedia.org [online]. [cit. 2022-01-04].  
Dostupné z: [https://en.wikipedia.org/wiki/Apache\\_Maven](https://en.wikipedia.org/wiki/Apache_Maven)

[14]1. díl: WordPress REST API: co umí a jak ho můžete využít i vy. Master.cz [online]. [cit. 2022-01-04].  
Dostupné z: <https://www.master.cz/blog/wordpress-rest-api-vysvetleni-prvni-dil/>

API. Wikipedia.org [online]. [cit. 2022-01-04].  
Dostupné z: <https://en.wikipedia.org/wiki/API>

[15] SQL vs NoSQL: 5 Critical Differences. Integrate.io [online]. [cit. 2022-01-04].  
Dostupné z: <https://www.integrate.io/blog/the-sql-vs-nosql-difference/>

[16] Databáze NoSQL – co je NoSQL? Azure.microsoft.com [online]. [cit. 2022-01-04].  
Dostupné z: <https://azure.microsoft.com/cs-cz/overview/nosql-database/>

[17] PostgreSQL: The World's Most Advanced Open Source Relational Database.  
Postgresql.org [online]. [cit. 2022-01-04]. Dostupné z: <https://www.postgresql.org/>

[18] Junit5. Junit.org [online]. [cit. 2022-01-04]. Dostupné z: <https://junit.org/>

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20