

# Expertný systém na predikciu úspechu startupu

Bc. Roman Čech

---

Diplomová práce  
2024



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
Ústav informatiky a umělé inteligence

Akademický rok: 2023/2024

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

Jméno a příjmení: Bc. Roman Čech  
Osobní číslo: A21492  
Studijní program: N0613A140022 Informační technologie  
Specializace: Softwarové inženýrství  
Forma studia: Kombinovaná  
Téma práce: Expertní systém pro predikci úspěchu startupu  
Téma práce anglicky: Expert System for Predicting Startup Success

### Zásady pro vypracování

1. Proveďte literární rešerši v oblasti knowledge-based systémů.
2. Zhodnoťte možnosti evaluace potenciálu startup firem a vyberte vhodné atributy podílející se na rozhodovacím procesu, např. cílový trh, profil klienta, analýza trhu a konkurence.
3. Vytvořte znalostní bázi vhodnou pro znalostní expertní systém pro finální ohodnocení a zdůvodnění výsledné evaluace.
4. Využijte a dolaďte předtrénovaný velký jazykový model pro výsledné slovní reporty.
5. Implementujte back-end se znalostním systémem, front-end a API ke komunikaci mezi front-endem a back-endem.
6. Zabývejte se zabezpečením aplikace.

Forma zpracování diplomové práce: **tištěná/elektronická**

**Seznam doporučené literatury:**

1. SEPPÄNEN, Pertti, et al. How Are Product Ideas Validated? The Process from Innovation to Requirements Engineering in Software Startups. In: *Software Business: 8th International Conference, ICSOB 2017, Essen, Germany, June 12-13, 2017, Proceedings 8*. Springer International Publishing, 2017. p. 3-17.
2. KEMELL, Kai-Kristian, et al. Startup metrics that tech entrepreneurs need to know. In: *Fundamentals of Software Startups: Essential Engineering and Business Aspects*. Cham: Springer International Publishing, 2020. p. 111-127.
3. CHIARELLI, Andrea. *Beginning React: Simplify your frontend development workflow and enhance the user experience of your applications with React*. Packt Publishing Ltd, 2018.
4. VIPUL, A. M.; SONPATKI, Prathamesh. *ReactJS by Example-Building Modern Web Applications with React*. Packt Publishing Ltd, 2016.
5. SALATINO, Mauricio; DE MAIO, Mariano; ALIVERTI, Esteban. *Mastering Jboss drools 6*. Packt Publishing Ltd, 2016.
6. AKERKAR, Rajendra; SAJJA, Priti. *Knowledge-based systems*. Jones & Bartlett Publishers, 2009.

Vedoucí diplomové práce: **prof. Ing. Zuzana Komínková Oplatková, Ph.D.**  
Ústav informatiky a umělé inteligence

Datum zadání diplomové práce: **5. listopadu 2023**

Termín odevzdání diplomové práce: **13. května 2024**

**doc. Ing. Jiří Vojtěšek, Ph.D. v.r.**  
děkan



**prof. Mgr. Roman Jašek, Ph.D., DBA v.r.**  
ředitel ústavu

Ve Zlíně dne 5. ledna 2024

### **Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně;
- byl/a jsem seznámen/a s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 11.5.2024

Roman Čech, v.r.

.....  
podpis studenta

## **ABSTRAKT**

Hlavným predmetom tejto práce je vytvoriť expertný systém pomocou systému Drools, ktorý dokáže odhadnúť pravdepodobnosť úspechu finančno-technologických startupov v počiatočnej fáze. Používa metódu hodnotenia podľa skóre. Vytvorenie moderného webového rozhrania pomocou Reactu zlepšuje používateľský zážitok. Systém slúži hlavne začínajúcim podnikateľom, ktorí majú záujem začať podnikat' a zároveň majú jasnú predstavu o produkte a trhu na ktorom sa chcú uchytiť. Prednosťou aplikácie je komunikácia so serverom OpenAI s využitím modelu GPT-3.5, ktorá umožňuje generovanie cennej spätnej väzby pre podnikateľa.

Kľúčová slova: Expertný systém, metóda hodnotenia podľa skóre, Drools, React, GPT-3.5

## **ABSTRACT**

The main objective of this diploma thesis is to develop an expert system using the system Drools that can estimate the probability of success of pre-seed financial technology startups. It uses a scorecard valuation method. A modern web interface using React improves the user experience. The system mainly serves start-ups that are interested in starting a business and have a clear idea of the product and market they want to conquer. The strength of the application is the communication with the OpenAI server using the GPT-3.5 model, which allows the generation of valuable feedback for the entrepreneur.

Keywords: Expert system, scorecard valuation method, Drools, React, GPT-3.5

Rád by som sa poďakoval vedúcej diplomovej práce prof. Ing. Zuzane Komínkovej Oplatkovej Ph.D. za užitočné pripomienky, usmernenia a čas pri spracovaní diplomovej práce. Taktiež ďakujem konzultantovi Ing. Luiz Faria z Univerzity ISEP za odbornú pomoc pri spracovaní praktickej časti diplomovej práce.

Prehlasujem, že som použil nástroj generatívneho AI [GPT 3.5 Turbo; openai.com] za účelom integrácie generovania spätnej väzby pre expertný systém, ktorý odhaduje pravdepodobnosť úspechu startupu.

Prehlasujem, že odovzdaná verzia diplomovej práce a verzia elektronická nahraná do IS/STAG sú totožné.

# OBSAH

<b>ÚVOD</b> .....	<b>9</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>11</b>
<b>1 PREHLAD LITERATÚRY</b> .....	<b>12</b>
<b>2 ZNALOSTNÉ SYSTÉMY</b> .....	<b>13</b>
2.1    UMELÁ INTELIGENCIA .....	13
2.2    CHARAKTERISTIKA ZNALOSTNÝCH SYSTÉMOV .....	13
2.3    ZÁKLADNÉ KATEGÓRIE ZNALOSTÍ .....	15
2.3.1    Deklaratívne znalosti .....	15
2.3.2    Procedurálne znalosti .....	15
2.3.3    Metaznalosti .....	15
2.4    REPREZENTÁCIA ZNALOSTÍ .....	16
2.5    ARCHITEKTÚRA ZNALOSTNÝCH SYSTÉMOV .....	16
<b>3 EXPERTNÝ SYSTÉM</b> .....	<b>18</b>
3.1    UŽIVATELSKÉ ROZHRAŇIE .....	19
3.2    BÁZA ZNALOSTÍ .....	19
3.3    INFERENČNÝ MECHANIZMUS .....	21
3.3.1    Postupné reťazenie .....	21
3.3.2    Spätné reťazenie .....	23
3.3.3    Pracovné úložisko .....	24
<b>4 SYSTÉM DROOLS</b> .....	<b>25</b>
4.1    JAZYK PRAVIDEL DROOLS .....	25
4.2    DROOLS RUNTIME .....	26
4.3    MECHANIZMUS PRODUKČNÝCH PRAVIDEL .....	26
<b>5 VEĽKÝ JAZYKOVÝ MODEL</b> .....	<b>28</b>
5.1    MECHANIZMUS LLM .....	28
5.2    VÝHODY A NEVÝHODY LLM.....	29
<b>II PRAKTICKÁ ČÁST</b> .....	<b>31</b>
<b>6 ROZHODOVACÍ MODUL</b> .....	<b>32</b>
6.1    ZDROJ DÁT .....	32
6.1.1    Prehľad expertov .....	32
6.1.2    Získavanie vedomostí.....	33
6.1.3    Hodnotenie faktorov.....	33
6.2    REPREZENTÁCIA DÁT .....	34
6.2.1    Produkčné pravidlá.....	34
6.3    METÓDA HODNOTENIA PODĽA SKÓRE .....	36

6.3.1	Zohľadnenie všetkých faktorov.....	36
6.3.2	Výpočet hodnotenia .....	37
6.4	IMPLEMENTÁCIA .....	38
6.5	ZHRNUTIE IMPLEMENTÁCIE.....	43
<b>7</b>	<b>SERVEROVÁ ČASŤ .....</b>	<b>45</b>
7.1	ZABEZPEČENIE POMOCOOU JWT .....	45
7.2	KONFIGURÁCIA DROOLS .....	46
7.3	REDIS .....	47
7.3.1	Konfigurácia.....	47
7.3.2	Vydavateľ, Konzument a Pracovník .....	47
7.4	REST API.....	48
7.4.1	Spring MVC.....	48
7.4.2	Spring WebFlux .....	50
7.5	KONTEJNERIZÁCIA .....	51
7.6	JAZYKOVÝ MODEL GPT .....	52
7.6.1	Požiadavka na OpenAI API .....	52
7.6.2	Odpoveď z OpenAI API .....	53
<b>8</b>	<b>KLIENSKÁ ČASŤ.....</b>	<b>54</b>
8.1	ŠTRUKTÚRA PROJEKTU .....	54
8.2	ROZLOŽENIE STRÁNKY .....	59
	<b>ZÁVER .....</b>	<b>61</b>
	<b>ZOZNAM POUŽITEJ LITERATÚRY .....</b>	<b>63</b>
	<b>ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK.....</b>	<b>63</b>
	<b>ZOZNAM OBRÁZKOV .....</b>	<b>65</b>
	<b>ZOZNAM TABULIEK .....</b>	<b>68</b>



## ÚVOD

V dynamickom prostredí podnikania a investícií je ocenenie začínajúcich startupov v počiatočnom štádiu veľmi dôležité. Vzhľadom na obmedzené zdroje a vysoké riziká musia začínajúci podnikatelia alebo investori vedieť dôkladne posúdiť potenciál. Existuje niekoľko metodík oceňovania a zvyčajné hlavné faktory týchto metód sú tím, finančná prognóza alebo obchodný model, jedinečná ponuka hodnoty a porovnanie s konkurenciou.

Hlavným cieľom tejto práce je vybrať vhodnú metódu oceňovania, správne zanalyzovať základné faktory a vytvoriť znalostnú bázu vhodnú na ocenenie finančno-technologických startupov v počiatočnom štádiu. Dosiahnutie ocenenia je ale zložitý a neistý proces, ktorý patrí medzi kľúčové ukazovatele pre investíciu a očakávaný výnos. Zložitost' a subjektívnosť ocenenia sa ešte viac zvyrazňuje v prípade, keď sa hovorí o začínajúcich startupoch, ktoré ešte nedosiahli žiadny významný predaj alebo záväzky investorov. Preto sa ocenenie výrazne opiera o skúsenosti, znalosti správania a intuíciu expertov, najmä pri rozhodovaní o investíciách alebo vstupovaní na trh. Okrem toho je cieľom poskytnúť podnikateľovi spätnú väzbu v oblastiach, ktoré je potrebné zlepšiť predtým, ako požiada o investíciu alebo sa rozhodne pokračovať v ďalšom úsilí.

Prvá kapitola je venovaná prehľadu literatúry na tému oceňovania startupov s dôrazom na riziká s tým spojené. Zaoberá sa rôznymi metódami oceňovania a odkazuje sa na existujúci výskum v tejto oblasti. Druhá kapitola sa zaoberá teoretickým poznatkom znalostných systémov a samotných znalostí. Sú objasnené základné kategórie znalostí, ich reprezentácia a architektúra znalostných systémov. Obohatením o teoretické poznatky z druhej kapitoly sa tretia kapitola zaoberá podrobnejšiemu vysvetleniu expertného systému. Sú vysvetlené komponenty systému ako užívateľské rozhranie, báza znalostí, inferenčný mechanizmus a priblížené dva najčastejšie spôsoby zdôvodňovanie postupné a spätné reťazenie. Štvrtá kapitola obsahuje prechod z čisto teoretických poznatkov na praktickú časť, ktorou je systém Drools. Je predstavený jazyk pravidiel, Drools runtime a mechanizmus produkčných pravidiel. Piata a posledná kapitola teoretickej časti sa zaoberá priblížením veľkého jazykového modelu. Vysvetľuje jeho mechanizmus a fungovanie procesu tréningu. Definuje výhody a nevýhody veľkého jazykového modelu.

Praktickou implementáciou rozhodovacieho modelu sa zaoberá šiesta kapitola. Začiatok kapitoly poskytuje prehľad odborníkov v danej oblasti, opisuje proces získavania znalostí a spôsob, akým odborníci hodnotili hlavné faktory úspechu. Následne je predstavená

technická reprezentácia dát a metóda hodnotenia podľa skóre, ktorá je použitá ako metóda na ocenenie startupu. Záver kapitoly upiera pozornosť na praktickú časť, ktorá definuje formu pravidiel, výpočet pravdepodobností úspechu a zhrnutie úspešnosti expertného systému. Siedma kapitola sa zaoberá serverovou časťou aplikácie. Popisuje postupne zabezpečenie aplikácie pomocou JWT tokenu, konfiguráciu projektu s Drools, integráciu Redisu pre neblokujúce spracovanie serverových požiadaviek, implementáciu REST API serveru, kontejnerizáciu projektu a nakoniec integráciu s veľkým jazykovým modelom GPT-3.5. Ôsma kapitola poskytuje vysvetlenie štruktúry projektu klientskej časti s dôrazom na hlavný priečinok zdrojového kódu. Na záver kapitoly sú zobrazené rozloženia stránok pre jednoduchšiu predstavu.

## **I. TEORETICKÁ ČÁST**

## 1 PREHLAD LITERATÚRY

Začiatok každého podnikania nepochybne vyžaduje určité investície. Riziká spojené s touto cestou tvoria jednu z najväčších prekážok každej investície. Práve preto je dôležité risk minimalizovať a predvídať ho vopred. Bayesové siete (BN) získavajú veľkú pozornosť ako výkonná metóda na modelovanie rozhodovania v podmienkach neistoty v rôznych odvetviach. Získavanie pravdepodobností BN je možné pomocou analýzy a názoru expertov [1].

Na ocenenie startupu existuje niekoľko metód. Niektoré sa používajú na ocenenie pred dosiahnutím príjmov, zatiaľ čo iné sú korektnejšie až po ich dosiahnutí. Metódy ocenenia zahŕňajú napríklad metódu hodnotenia podľa skóre, metódu rizikového kapitálu, metódu diskontovaných peňažných tokov, metódu nových podnikov založenú na riziku a ďalšie.

Existuje výskum využívajúci metódu nových podnikov, ktorá je založená na riziku. Tento výskum sa zaoberá ocenením po prieskume trhu, beta testovaní a akvizícií predajných kanálov pri určitých rizikách s odporúčaním a identifikovanou potrebou ďalšieho preskúmania [2].

Rovnako existuje výskum, ktorý využil metódu diskontnej sadzby, kde dôležitú úlohu zohráva koeficient beta ako pravdepodobný risk. Autor sa odvoláva, že ich návrh môže byť aplikovaný na ľubovoľnú začínajúcu spoločnosť, ktorý má možnosť porovnať sa s konkurenciou [3].

Keďže sa diplomová práca primárne zaoberá podnikmi v štádiu pred investíciou, vhodnou metódou je metóda hodnotenia podľa skóre (angl. Scorecard Method). Na formulovanie pravdepodobnostných pravidiel využíva Bayesovskú inferenciu. Riešenie prezentované v tejto práci je obohatené o integráciu veľkého jazykového modelu za účelom generovania slovných reportov. Kombináciou veľkého jazykového modelu a expertného hodnotiaceho systému by mohol vzniknúť unikátny systém, ktorý používateľovi poskytne spätnú väzbu generovanú jazykovým modelom založeným na výkonnom výpočtovom prvku. Aby sa používateľovi čo najviac uľahčila práca, je riešenie obohatené o moderné web rozhranie pre ešte jednoduchšiu správu.

## 2 ZNALOSTNÉ SYSTÉMY

V neustále sa vyvíjajúcej oblasti správy informácií a riešenia problémov sa znalostné systémy stávajú kľúčovými štruktúrami, ktoré spájajú umelú inteligenciu a reprezentáciu znalostí. Táto úvodná kapitola vytvára základ pre pochopenie základných princípov, ktoré sú podstatou znalostných systémov. Zaoberá sa kľúčovými pojmami, ako je oblasť umelej inteligencie, charakteristika znalostných systémov, ich reprezentácia a základné kategórie znalosti. Súčasťou kapitoly 2.5 je vizualizácia základnej architektúry znalostných systémov.

### 2.1 Umelá inteligencia

Umelá inteligencia (AI, z artificial intelligence) je simulácia procesov ľudskej inteligencie strojmi, hlavne počítačovými systémami. Medzi aplikácie umelej inteligencie patria expertné systémy, rozpoznávanie reči, spracovanie prirodzeného jazyka a strojové učenie [4].

Vo všeobecnosti sa hovorí, že aplikácia zahŕňa AI, keď vykazuje akékoľvek znaky inteligencie. V oblasti informatiky je AI definovaná ako študijný odbor, ktorý sa snaží napodobniť a vysvetliť inteligentné správanie vo výpočtových procesoch. Rieši problémy, pri ktorých si konvenčné metódy vyžadujú nekonečné množstvo času a tie, pre ktoré nie sú známe účinné riešenia. Zároveň je jej cieľom zvýšiť efektivitu riešenia [5].

### 2.2 Charakteristika znalostných systémov

Znalostný systém (KBS) je počítačový systém založený na znalostiach, ktorý je hlavnou súčasťou umelej inteligencie. Vytvára a využíva znalosti z poznatkov, informácií a údajov. Tento systém je schopný pochopiť spracúvaným informáciám a môže na ich základe prijať rozhodnutie, zatiaľ čo bežné počítačové systémy nepoznajú a nerozumejú vstupom, ktoré spracúvajú.

V súčasnosti s dostupnosťou všetkých moderných výpočtových zariadení, sa pozornosť zameriava na komplexnejšie úlohy, ktoré môžu vyžadovať určitú formu inteligencie. Spoločnosť a priemysel sa upriamuje na znalosti a pri riešení náročnejších problémov sa spoliehajú na rozhodovacie schopnosti expertov v danej problematike.

KBS môže slúžiť aj ako expert na požiadanie, kedykoľvek a kdekoľvek. Dokáže ušetriť peniaze tým, že využije rôznych expertov, podporí konzistentnosť a umožní používateľom

fungovať na kvalitnejšej úrovni. Je to produktívny nástroj, ktorý poskytuje kolektívne znalosti jedného alebo viacerých expertov.

V týchto systémoch je striktne oddelená znalosť od uvažovania, čo znamená, že riadenie programu je nezávislé od špecifikácie znalostí. Znalosť môže byť kódovaná pomocou vhodnej reprezentačnej schémy ako logika, sémantické siete, odvodzovacie pravidlá, skripty či jazyky pre reprezentáciu znalosti.

Podľa Tuthill-a a Levy-ho (1991) existuje päť hlavných typov KBS [5]:

- **Expertný systém** – aplikácia, ktorá sa zameriava na riešenie zložitých problémov podobným spôsobom ako by to urobili ľudskí experti.
- **Prepojený systém** – zvyčajne sa vzťahuje na skupinu prepojených komponentov, ktoré vykazujú určitú inteligenciu a kooperujú na dosiahnutie cieľa. Ide o hypermediálne systémy, ktoré zahŕňajú formáty ako hypertext, hyperaudio a hypervideo.
- **CASE Systém** – usmerňuje vývoj inteligentných alebo informačných systémov na dosiahnutie efektívnosti a vyššej kvality, na základe predchádzajúcich skúseností na riešenia nových problémov. Poskytuje potrebné usmernenia pre výber modelu, získanie podstatných požiadaviek na vyvíjaný produkt. Tieto systémy podporujú činnosti riadenia rizík.
- **Databáza v spojení s inteligentným používateľským rozhraním** – ponúka efektívny spôsob správy a vyhľadávania informácií. Databáza sa stará sa o ukladanie a organizáciu a rozhranie poskytuje intuitívne funkcie. Nedokáže spracovať neúplné informácie a chýba mu samostatnosť pri zdôvodňovaní či rozhodovaní.
- **Inteligentné výučbové systémy** – sa používajú na výučbu študentov, školiteľov a odborníkov v špecifických odboroch a na rôznych úrovniach. Tieto systémy sa používajú na identifikáciu úrovne používateľov a ďalších obmedzení na poskytovanie školení v technických a netechnických odvetviach. Medzi známe, patria výučbové systémy založené na dialógu [5].

Expertným systémom sa podrobnejšie venuje kapitola 3.

## 2.3 Základné kategórie znalostí

Nonaka a Takeuchi (1995) popisujú znalosť ako „odôvodnené pravdivé presvedčenie“ súvisiace s ľudskou aktivitou. Znalosť je možné získať štúdiom či skúsenosťami a je založená na učení a správnom pochopení určitej problematiky jednotlivcom. Možno ju považovať za syntézu procesov vnímania ľudí, ktorá im pomáha vyvodit' zmysluplné závery. Znalosť možno chápať ako súbor integrovaných faktov a vzťahov, ktoré pri správnej interpretácii môžu viesť k efektívnemu výsledku v rôznych úlohách [5].

Medzi základné kategórie znalostí patria deklaratívne znalosti, procedurálne znalosti a metaznalosti.

### 2.3.1 Deklaratívne znalosti

Deklaratívne znalosti slúžia ako opisná reprezentácia informácií, ktorá sprostredkúva fakty o tom, aké veci sú v zrozumiteľnom tvare. Vyjadrujú sa vo forme faktických tvrdení a poskytujú ich experti v danej oblasti, ktoré objasňujú súvislosti. Dôležité sú najmä v počiatočnej fáze získavania znalostí.

**Napríklad:** Fajčenie môže spôsobiť rakovinu pľúc.

### 2.3.2 Procedurálne znalosti

Procedurálne znalosti sa skladajú z postupnosti krokov a inštrukcií „ako na to“. Zahŕňajú automatizované reakcie na podnety, ponúkajú návod na využívanie deklaratívnych znalostí a vyvodzovanie záverov. Zameriavajú sa na postupy riešenia problémov ako je definícia problému, zber údajov, postupy riešenia a kritériá hodnotenia. Tradičným spôsobom reprezentácie znalostí sú pravidlá, stratégie a procedúry.

**Napríklad:** Jazda na bicykli (Ako postupovať)

### 2.3.3 Metaznalosti

Metaznalosti sa zaoberajú pochopením samotných znalostí. V oblasti KBS sa metaznalosti vzťahujú na pochopenie fungovania systémov založených na znalostiach, a ich schopnosti zdôvodňovania záverov [6].

**Napríklad:** Vzdelávanie sa

## 2.4 Reprezentácia znalostí

Reprezentácia znalostí je oblasť, ktorá sa venuje vytváraniu symbolických rámcov, ktoré zobrazujú informácie, pojmy a súvislosti v rámci určitej oblasti. Hlavným cieľom je umožniť počítačom ukladať, spracovávať a chápať zložité údaje, a tým im umožniť uvažovanie a vyvodzovanie záverov. Cieľom reprezentácie znalostí je v konečnom dôsledku zefektívniť riešenie problémov a rozhodovanie využitím efektívneho využívania a interpretácie uložených znalostí [7].

Medzi bežné používané techniky reprezentácie znalostí patria:

- **Sémantická sieť** – je štruktúra založená na grafoch, ktoré predstavujú vzťahy medzi entitami a ich atribútmi. Sú ľahko pochopiteľné a dajú sa pomerne jednoducho rozširovať.
- **Rámec** – je dátová štruktúra, ktorá sa skladá zo súboru atribútov a ich hodnôt na opis entity. Zahrňuje sloty a k nim priradené hodnoty, ktoré sa líšia typom a veľkosťou. Každý slot má názov a zodpovedajúcu hodnotu.
- **Logika** – je štruktúrovaný jazyk riadený jasnými pravidlami, špeciálne navrhnutý na zaobchádzanie s vetami bez dvojakeho významu. Vyznačuje sa dobre definovanou syntaxou a sémantikou.
- **Produkčné pravidlá** – skladajú sa z dvojíc IF (podmienka) a WHEN (akcia). Pozostáva najmä z troch častí: súbor pravidiel, pracovná pamäť a cyklus na rozpoznávanie činností. [8]

Prostredníctvom štruktúrovanej reprezentácie znalostí môžu tieto systémy interpretovať zložité informácie, vyvodzovať závery, vytvárať stratégie a prijímať rozhodnutia založené na existujúcich znalostiach, čím napodobňujú inteligenciu podobnú ľudskej. Neexistuje jediná metóda reprezentácie vhodná pre všetky typy znalostí [7].

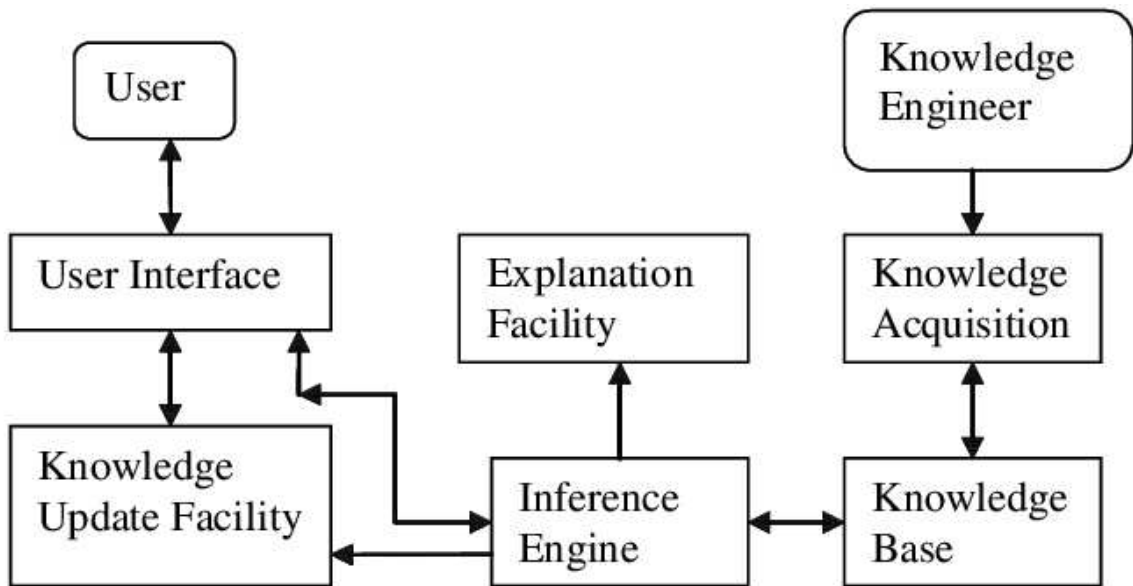
## 2.5 Architektúra znalostných systémov

Architektúra znalostného systému zahŕňa niekoľko komponentov (Obrázok 1), z ktorých každý zohráva kľúčovú úlohu v jeho funkčnosti. Jeho jadrom je znalostná báza (angl. Knowledge Base), úložisko znalostí špecifických pre danú oblasť, ktoré sa získavajú od ľudských expertov rôznymi technikami získavania. Inferenčný mechanizmus (angl. Inference Engine) riadi rozhodnutia a jeho úlohou je na základe pravidiel vyvodzovať



závery. Používateľské rozhranie (angl. User Interface) slúži ako brána medzi chápaním používateľa a fungovaním systému. Medzi dielčie komponenty patria proces na získavanie znalostí (angl. Knowledge Acquisition), ktorý uľahčuje zhromažďovanie znalostí a nástroj na vysvetľovanie (angl. Explanation Facility), ktorý poskytuje prehľad o fungovaní.

So systémom interaguje používateľ, znalostný inžinier (angl. Knowledge Engineer), ktorý vyvíja mechanizmus a doménový expert (angl. Domain Expert), ktorý prináša do systému znalosti [9].



Obrázok 1 Architektúra znalostných systémov

### 3 EXPERTNÝ SYSTÉM

Expertný systém je aplikácia umelej inteligencie, ktorá je zameraná na riešenie komplexných problémov. Táto oblasť je priekopníkom v riešení komplexných problémov, ktoré sa tradičnými metódami riešia náročne. Výskumníci dosiahli výrazný pokrok v oblasti umelej inteligencie za posledných päť desaťročí.

Expertné systémy, ktoré sa zrodili v 70. rokoch 20. storočia a ich rozšírenie nastalo v 80. rokoch 20. storočia, znamenajú prvý úspech umelej inteligencie. Systémy vychádzajú z jedinečnej paradigmy AI známej ako expertné systémy založené na pravidlách. Ich vývoj si vyžaduje prácu znalostného inžiniera, ktorý analyzuje rozhodovacie procesy ľudských expertov a transformuje ich pravidlá do počítačom zrozumiteľnej podoby.

Proces vytvárania expertných systémov sa nazýva znalostné inžinierstvo. Produkčné pravidlové systémy sú implementácie pravidlového prístupu, založené na pravidlách a na realizáciu expertných systémov [10].

Definície expertných systémov:

- Expertný systém je aplikácia, ktorá symbolizuje rozhodovacie odborné znalosti experta. Navrhuje sa aby riešil komplexné problémy tak, že sa na znalosti pozerá ako by to urobil ľudský expert.
- Expertný systém je program, ktorý simuluje správanie a úsudok človeka, ktorý má v určitej oblasti odborné znalosti a skúsenosti. Vo väčšine prípadov takýto systém zahŕňa znalostnú bazu, ktorá obsahuje nahromadené skúsenosti a súbor pravidiel na využitie znalostnej bázy na každú špecifickú situáciu, ktorá je počítačovému programu definovaná.
- Expertný systém je počítačový systém, ktorý realizuje úlohu, ktorú by v inom prípade vykonával ľudský expert. Momentálne poznáme expertné systémy, ktoré dokážu vytvárať finančné prognózy, diagnostikovať ľudské choroby či plánovať trasy doručovacích vozidiel.
- Expertný systém je možné dokonca nazvať aj ako inteligentný program, ktorý poskytuje odborné znalosti v riešení problémov pomocou špecifických znalostí príslušných pre konkrétnu oblasť, ako by to robil ľudský expert.

Každý kvalitný expertný systém by mal byť ľahko použiteľný aj pre začínajúceho používateľa, schopný vysvetliť poskytnuté rady, učiť sa nové informácie, prijímať včasné rozhodnutia a vykazovať vysoký výkon.

Najčastejším riešením pre expertné systémy je systém založený na pravidlách alebo produkčný systém. Tento druh systému využíva znalosti zapísané vo forme produkčných pravidiel, čiže pomocou pravidiel formátu IF...ELSE. Každé pravidlo obsahuje malú časť vedomosti z konkrétnej oblasti odbornosti. Ak sa realita zhoduje z ľavou časťou pravidla, vykoná sa akcia [10].

**Syntax:** IF Podmienka THEN Akcia.

### 3.1 Užívateľské rozhranie

Užívateľské rozhranie funguje ako sprostredkovateľ medzi expertným systémom a používateľom, ktoré značne uľahčuje komunikáciu. Prezентuje otázky, možnosti a odporúčania vygenerované inferenčným mechanizmom, ktorým užívateľ rozumie. V závislosti od konkrétneho systému môže mať podobu textových a grafických prvkov. Spravidla sa volí rozhranie grafické, pretože je užívateľský intuitívnejšie. Práve preto sa o jeho podobe sa rozhoduje hneď na začiatku pri návrhu expertného systému.

Používateľ zadáva príkazy a odpovedá na príslušné otázky. Expertný systém reaguje na vstupy a kladie otázky v priebehu procesu odvodzovania. Sofistikované rozhrania vo veľkej miere využívajú rozdielne metódy ako napríklad vyskakovacie okná. Prostredníctvom tohto rozhrania užívatelia dostávajú prehľad o aktuálnych činnostiach systému vrátane aktuálnych výsledkov, opodstatnení a nasledujúcich krokov [10], [11].

### 3.2 Bába znalostí

Medzi základný kameň expertného systému patrí báza znalostí, ktorá obsahuje sadu pravidiel, faktov a heuristik reprezentujúce konkrétne odborné znalosti ľudskeho experta, ktoré sú vyjadrené vo forme pravidiel. Bázu znalosti vytvoril expert a je zakódovaná vo formálne zrozumiteľnom jazyku pre expertný systém. Expert znalosti nadobúda prostredníctvom univerzitných a hlavne niekoľkoročných skúseností z praxe. Tieto expertné znalosti sú získané znalostným inžinierom skrz techniky získavania znalostí, kedy je expert schopný definovať znalosti v podobe pravidiel [10], [11].

Báza znalostí pozostáva zo znalostí, ktoré majú podobu pravidiel:

- ľavá časť s IF, nazývaná ako podmienková časť alebo LHS
- pravá časť s THEN, nazývaná ako akciová časť alebo RHS

### **Napríklad:**

Pravidlo Počasie:

IF "vonku prší" THEN "použi dáždnik"

Pravidlo Účet:

IF "prázdny bankový účet" THEN "zamietni transakciu"

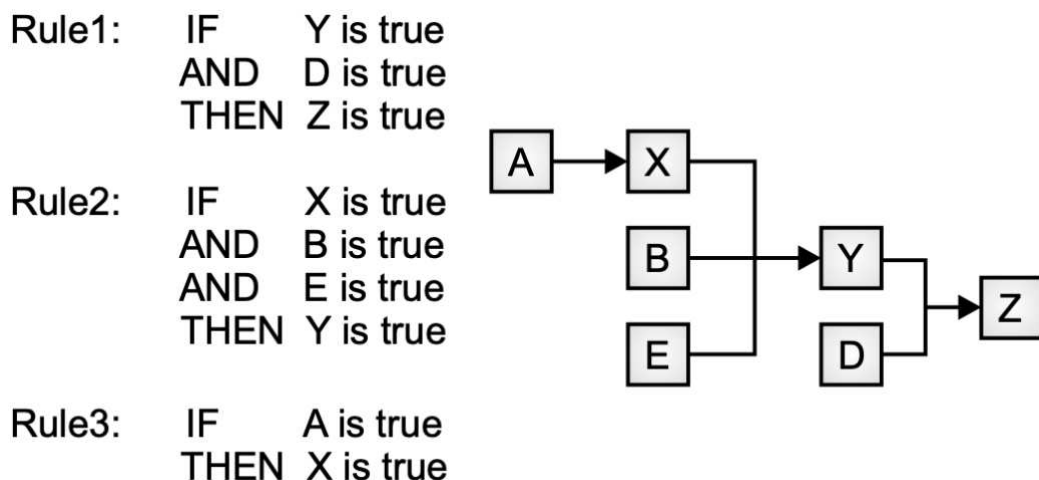
Existuje pravidlo, ktoré môže mať spojených viac podmienkových častí pomocou kľúčových slov A alebo ALEBO. Tento druh pravidla sa nazýva zložené pravidlo a má nasledujúcu štruktúru:

- 1) IF [podmienka 1]  
AND [podmienka 2]  
....  
AND [podmienka n]  
THEN [akcia]
- 2) IF [podmienka 1]  
OR [podmienka 2]  
....  
OR [podmienka n]  
THEN [akcia]
- 3) IF [podmienka 1]  
AND [podmienka 2]  
OR [podmienka 3]  
....  
THEN [akcia]

Dôležité je poznamenať, že pravidlá v znalostnej báze môžu obsahovať rôznorodé typy vzťahov, návrhov, heuristik či odporúčaní [10].

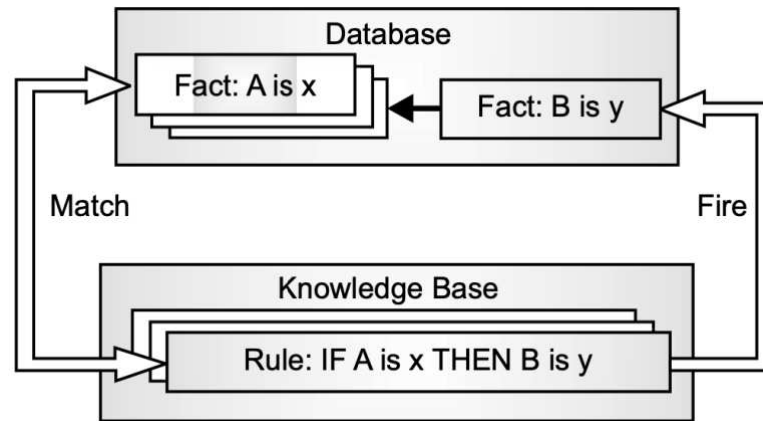
### 3.3 Inferenčný mechanizmus

Inferenčný mechanizmus je mechanizmus, ktorý vykonáva úlohy ako uvažovanie a rozhodovanie (napr. Obrázok 2). Aplikuje pravidlá a postupy definované v báze znalosti na spracovanie vstupných dát a vytvorenie očakávaného výstupu [11]. Tento mechanizmus funguje ako mozog systému a generuje zdôvodnenia pravidiel porovnávaním faktov a údajov s produkčnými pravidlami. Keď sa podmienková časť pravidla zhoduje s faktom, pravidlo sa aktivuje a vykoná sa akciová časť, prostredníctvom čoho dokáže motor generovať nové informácie zo znalostnej bázy a vstupných dát. Týmto porovnávaním vznikajú odvodzovacie reťazce, ktoré určujú ako uplatniť pravidla na dosiahnutie záveru.



Obrázok 2 Príklad inferenčného reťazca [10]

Inferenčný motor vykonáva proces, porovnávanie faktov (angl. Fact) s produkčnými pravidlami (angl. Production Rules), ktorý je známy aj ako porovnávanie vzorov (Obrázok 3). Jeho významnou úlohou je stanoviť priority pravidiel a spustenie pravidla s najvyššou prioritou. Pravidlá sa ukladajú do produkčnej pamäte a fakty sa ukladajú do pamäte pracovnej (angl. Knowledge Base) [10].



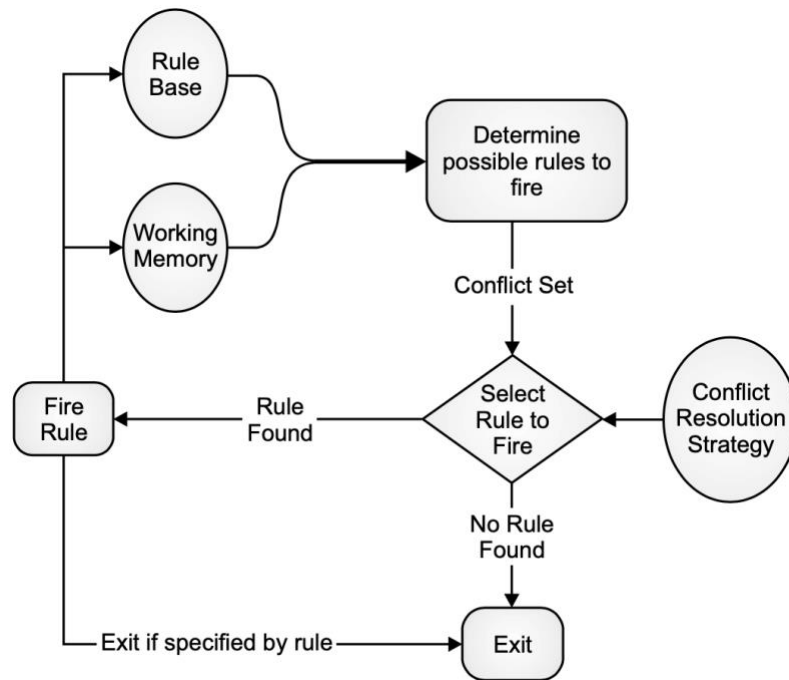
Obrázok 3 Fungovanie mechanizmu skrz postup Match-Fire [10]

V rámci expertných systémov existujú dva najčastejšie spôsoby zdôvodňovania:

- postupné reťazenie (angl. forward chaining)
- spätné reťazenie (angl. backward chaining)

### 3.3.1 Postupné reťazenie

Pri postupnom reťazení (Obrázok 4) expertný systém číta a spracováva súbor faktov, aby opodstatnene predpovedal, čo sa stane neskôr. Je to dátami riadený proces, v priebehu ktorého sa postupuje od faktov k záverom. To vyjadruje postupnosť dopredu, pričom fakty sa zaznamenávajú do pracovnej pamäte, čo vedie k jeho šíreniu, kedy je skutočne jedno alebo viac pravidiel korektných. Vyhodnocujú sa najvyššie pravidla a za každé vykonanie pravidla sa do pracovnej pamäte pridá nový fakt. Tento proces končí, keď už nie je možné nájsť žiadne pravidlo na vykonanie. Príkladom toho môže byť prognóza vývoja na burze [10], [11].

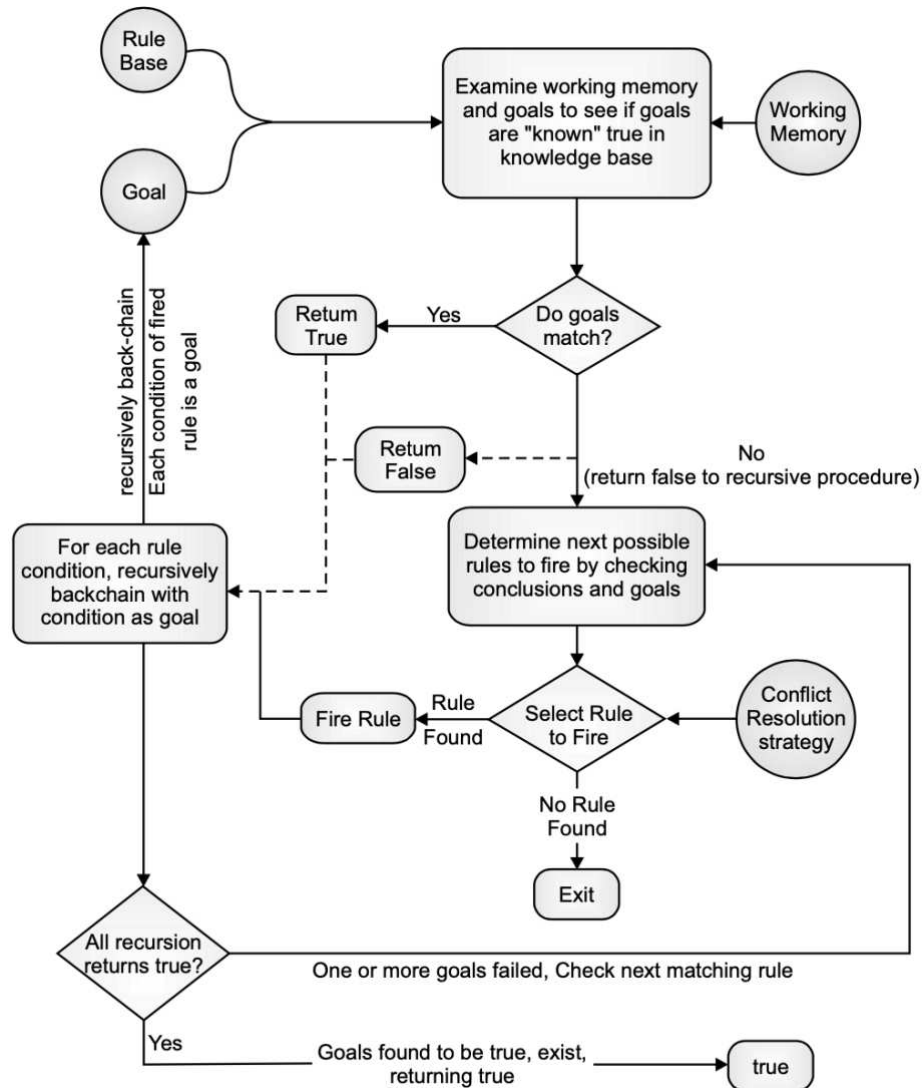


Obrázok 4 Postupné reťazenie [10]

Významnou nevýhodou je, že niektoré pravidlá postupného reťazenia sa vykonajú aj vtedy, keď nijako neprispievajú k dosiahnutiu cieľa. Vedie to k neefektívnemu uvažovaniu. Predovšetkým, ak je cieľom demonštrovať iba jeden fakt [10].

### 3.3.2 Spätné reťazenie

Pri reťazení spätnom (Obrázok 5) expertný systém číta a vyhodnocuje súbor informácií retrospektívne, aby dospel k opodstatnenému záveru o cieľi. Je to reťazenie riadené cieľom, čo znamená, že sa začína záverom, ktorý sa mechanizmus snaží splniť. Proces sa začína hľadaním pravidla v znalostnej báze, ktorý má tento cieľ. Ak sa pravidlo nájde a ak sa podmienky pravidla zhodujú s dátami v databáze, vykoná sa pravidlo a dokáže sa cieľ. Ak sa zodpovedajúce pravidlo nenájde, hľadajú sa závery, ktoré sa môžu čiastočne splniť. Tu sa hľadajú ďalšie pravidlá, aby sa zdôvodnila platnosť čiastkového cieľa. Tento cyklus pokračuje až kým sa nedosiahne pôvodný záver, alebo kým sa nepotvrdia čiastkové ciele. Príkladom spätného reťazenia je prognóza v lekárskom diagnostickom systéme [10].



Obrázok 5 Spätne reťazenie [10]

### 3.3.3 Pracovné úložisko

Pracovné úložisko, označované aj ako databáza, uchováva fakty, ktoré inferenčný mechanizmus využíva na porovnanie s podmienkovou časťou pravidla za účelom nájdenia záveru. Toto úložisko zachytáva rozličné prvky relevantné pre aktuálnu úlohu, vrátane odpovede používateľa na otázky akéhokoľvek priebežného výsledku uvažovania, všetkých záverov a údajov z externých zdrojov ku ktorým sa doteraz dospelo. Slúži ako dočasné úložisko pre mechanizmus počas jeho aktivity a obsahuje údaje, ktoré sú charakteristické pre riešený problém. Je dôležité vedieť rozlišovať medzi databázou a bázou znalostí. Zatiaľ čo znalostná báza uchováva opakovane použiteľné znalosti, naopak databáza obsahuje údaje kľúčové len pre jednotlivé prípady [10].



## 4 SYSTÉM DROOLS

Táto kapitola bude zameraná na popis softvérového nástroja Drools, ktorý výrazne uľahčuje implementáciu znalostných systémov.

System Drools predstavuje riešenie systému správy obchodných pravidiel (BRMS), ktoré obsahuje motor obchodných pravidiel (BRE), webový nástroj na tvorbu a správu pravidiel, plnú podporu rozhodovacích modelov a notácií (DMN) a doplnok Eclipse IDE na zjednodušenie vývoja.

Drools, vydaný pod licenciou Apache 2.0, je softvér s verejným zdrojovým kódom vytvorený výlučne v jazyku Java, takže je kompatibilný s akýmkoľvek JVM a prístupný prostredníctvom repozitára Maven Central [12].

### 4.1 Jazyk pravidiel Drools

Jazyk pravidiel Drools (DRL) slúži ako notácia vytvorená spoločnosťou Red Hat na automatizáciu obchodných pravidiel. Pravidlá sú zapuzdrené do textových súboroch s koncovkou .drl a môžu obsahovať jedno alebo viac pravidiel, ktoré sa skladajú z podmienky a akcie. Súbor DRL pozostáva z viacerých položiek.

1. package
2. unit
4. import
5. declare // Optional
6. query // Optional
7. rule "Rule name"
8. when
9. // Conditions
10. then
11. // Actions
12. end

Každé pravidlo musí mať jedinečný názov. Ak sa v niektorom súbore použije rovnaký názov pravidla viackrát, pravidlo sa neskompiluje. Názvy pravidiel je možné zapísať pomocou štandardných identifikátorov jazyka Java alebo vo forme pravidiel v dvojítych úvodzovkách ako napríklad "Názov pravidla" [12].

## 4.2 Drools Runtime

Drools ponúka flexibilitu pri vytváraní inštancií mechanizmu pravidiel prispôbených konkrétnym problémom. Každá inštancia funguje ako zapuzdrený kontext, v ktorom sa pravidlá vyhodnocujú na základe poskytnutých údajov. Na rozdiel od tradičných monolitických serverových procesov umožňuje systém Drools vytvárať lokálne inštancie v rámci aplikácií, čo uľahčuje paralelné spracovanie rôznych pravidiel a údajov. Medzi kľúčové koncepty vytvárania nových inštancií patria:

- **KieServices** – je trieda, ktorá poskytuje prístup k rôznym konceptom prostredníctvom registra služieb. Na získanie inštancie KieServices sa používa metóda `KieServices.Factory.get()`.
- **KieContainer** – môže vytvárať viacero inštancií mechanizmu pravidiel s rôznymi konfiguráciami a zbierkami pravidiel, čo umožňuje flexibilnú správu pravidiel v aplikáciách.
- **KieModule** – je štandardizovaný projekt Java-Maven, ktorý obsahuje prevažne pravidlá a obchodné procesy. Na prispôbenie konfigurácie modulu sa používa súbor `kmodule.xml` umiestnený v adresári `META-INF`, ktorý určuje vnútornú konfiguráciu.
- **KieBase** – predstavuje skompilovanú verziu súboru pravidiel. Používa sa ako základ pre `KieSession`.
- **KieSession** – je inštancia mechanizmu pravidiel v systéme Drools, ktorá obsahuje súbor pravidiel zostavených v rámci `KieBase`. Konfigurujú sa rozdielnymi spôsobmi, aby vyhovovali viacerým prípadom použitia.

Pomocou týchto piatich konceptov je možné nakonfigurovať každú inštanciu a určiť pravidlá. Pri vytváraní viacerých mechanizmov pravidiel, je zlomové pochopiť základné procesy aby sa predišlo zbytočnému preťažovaniu a nedostatkom vo výkone [13].

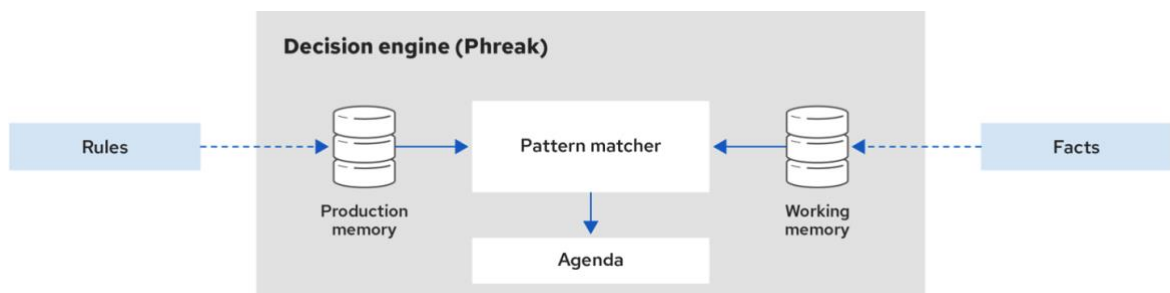
## 4.3 Mechanizmus produkčných pravidiel

Mechanizmus pravidiel riadi spracovanie, ukladanie a vyhodnocovanie dát s cieľom zaviesť zadané obchodné pravidlá alebo rozhodovacie modely. Základná funkcia spočíva v komparácií vstupných údajov alebo faktov s podmienkami pravidiel. Cieľom je určiť vhodné vykonanie pravidiel.

Medzi kľúčové komponenty (Obrázok 6) patria:

- **Pravidlá** – zahŕňajú obchodné pravidlá alebo rozhodnutia DMN. Každé pravidlo vyžaduje minimálne spúšťacie podmienky a zodpovedajúce akcie.
- **Skutočnosti** – Predstavujú údaje vstupujúce alebo meniace sa v rámci mechanizmu pravidiel. Priradujú k podmienkam pravidiel na vykonanie príslušných pravidiel.
- **Produkčná pamäť** – miesto, kde sú pravidlá uložené v mechanizme pravidiel Drools.
- **Pracovná pamäť** – pamäť, kde sú uložené fakty v rámci mechanizmu pravidiel Drools.
- **Agenda** - v tomto mieste sa registrujú a organizujú aktivované pravidla pripravené na vykonanie.

Keď používateľ alebo automatizovaný systém vytvorí alebo upraví údaje súvisiace s pravidlami (angl. rules) v systéme Drools, tak sa informácie zaradia do pracovnej pamäte (angl. working memory) mechanizmu ako jeden alebo viac faktov (angl. facts). Následne sa porovnávajú tieto fakty s podmienkami pravidiel uloženými v produkčnej pamäti (angl. production memory), aby sa identifikovali pravidlá vhodné na vykonanie. Po splnení podmienok sa spúšťa motor pravidiel (angl. pattern matcher), kde sa zapisujú pravidlá do agendy. Usporiadajú sa prioritné alebo konfliktné pravidlá pripravené na vykonanie [12].



Obrázok 6 Prehľad komponentov mechanizmu pravidiel [12]

## 5 VELKÝ JAZYKOVÝ MODEL

Velký jazykový model (LLM) je algoritmus hlbokého učenia, ktorý je schopný učiť sa z veľkej množiny textových dát a používať túto znalosť na generovanie nových textov. Tento model môže byť trénovaný na rôzne úlohy, ako je automatizované písanie článkov, tvorba kódu, obchodné správy a mnoho ďalších. LLM je závislý na veľkých zbierkach údajov pri vykonávaní svojich funkcií. Tieto zbierky údajov môžu obsahovať až 100 miliónov alebo viac parametrov, z ktorých každý reprezentuje premennú, ktorú jazykový model využíva na generovanie nového obsahu. Taktiež využívajú techniku nazývanú transferové učenie, ktorá umožňuje prenášať znalosti a skúsenosti získané pri riešení jednej úlohy na inú s podobným charakterom. Táto schopnosť umožňuje týmto modelom efektívne riešiť rozličné jazykové problémy, ako napríklad odpovedanie na otázky, klasifikáciu textu, sumarizáciu a generovanie textu [14].

### 5.1 Mechanizmus LLM

Veľké jazykové modely sú trénované na enormnom množstve dátových súborov ako sú text, obrázky, videá, reč a štruktúrované údaje. Týmto spôsobom sa snažia pochopiť a zvládnuť náročnosť ľudského jazyka. Čím viac parametrov a údajov LLM využíva, tým lepší je jeho výkon. Vyžadujú si značné množstvo zdrojov vrátane dát, výpočtových kapacít a techník. Pred trénovaním musia tieto modely absolvovať fázu, kedy sa učia základné lingvistické úlohy a funkcie. Táto fáza predtréningu vyžaduje najviac údajov a výpočtových zdrojov [15].

Cieľom tejto fázy je naučiť model vysokú úroveň funkcií, ktoré sa potom môžu uplatniť v jemnom ladení na konkrétne úlohy.

Proces trénovania veľkého jazykového modelu zahŕňa niekoľko krokov:

- Predbežné spracovanie textových údajov, ktoré ich konvertuje na číselnú reprezentáciu vhodnú pre vstup do modelu.
- Náhodné inicializovanie parametrov modelu.
- Vkladanie číselnej reprezentácie textových údajov do modelu.
- Použitie stratovej funkcie na meranie rozdielu medzi výstupmi modelu a skutočným ďalším slovom vo vete.
- Optimalizácia parametrov modelu s cieľom minimalizovať chyby.

- Opakovanie celého procesu, kým výstupy modelu nedosiahnu prijateľnú úroveň presnosti [16].

Cieľom potréningovej fázy je zdokonaľiť schopnosti modelu pre špecifické úlohy alebo prípady použitia. Na rozdiel od predtréningu, doladenie vyžaduje menej údajov a energie, pretože model už má základné lingvistické znalosti. Celkovo veľké jazykové modely potrebujú veľké množstvo údajov a výpočtových zdrojov na svoje tréningovanie [15].

Príklad jazykového modelu:

- Generative Pretrained Transformer 4 (GPT-4) – vyvinutý spoločnosťou OpenAI.
- Robustly Optimized BERT Approach (RoBERTa) – vyvinutý spoločnosťou Facebook AI.
- Text-to-Text Transfer Transformer (T5) – vyvinutý spoločnosťou Google.
- Megatron-Turing – vyvinutý spoločnosťou NVIDIA [16].

## 5.2 Výhody a nevýhody LLM

Medzi výhody patria:

- Jazykové modely umožňujú organizáciám automatizovať rôzne procesy, ako je generovanie textu, predikcia, klasifikácia a podobne.
- Využitím chatbotov a virtuálnych asistentov, môžu poskytovať personalizované služby zákazníkom.
- Zvyšujú presnosť úloh spracovávaním veľkého množstva údajov.
- Tieto kľúčové hodnoty jazykových modelov prispievajú k zlepšeniu efektivity procesov, personalizácii zákazníckych služieb a presnosti úloh [15].

Medzi nevýhody patria:

- Nedokážu skutočne rozumieť jazyku. Chýba im schopnosť porozumieť kontextu a významu obsahu, napríklad nie sú schopné rozumieť matematike.
- Nerozumejú faktom a nemajú samostatné režimy pre vyhľadávanie informácií alebo tvorivé písanie.
- Nerozumejú spôsobom správania, emóciám alebo etike.

- Je důležité uvedomiť si tieto obmedzenia a používať veľké jazykové modely s rozumnosťou a kritickým myslením [17].

## **II. PRAKTICKÁ ČÁST**

## 6 ROZHODOVACÍ MODUL

V tejto kapitole je vysvetlené fungovanie rozhodovacieho mechanizmu. Ďalej sa zaoberá zdrojom dát, kde sú predstavení experti a je vysvetlené získavanie vstupných informácií. Nasleduje oboznámenie sa s reprezentáciou dát a so spôsobom, ako priviesť poznatky do produkčných pravidiel. Okrem toho je vysvetlená metóda hodnotenia podľa skóre, ktorá hodnotí expertmi vybrané rámce a je ukázaný príklad ocenenia testovacieho podniku. Nakoniec je priblížená konkrétna implementácia a vizuálne zobrazenie celého procesu.

### 6.1 Zdroj dát

Cieľom je na základe poznatkov, ktoré poskytol uznávaný zdroj znalostí, integrovať do vývoja expertného systému rôznorodé pohľady a špecializované odborné znalosti investorov, úspešných podnikateľov, inkubátorov univerzitných iniciatív (ako sú centrá alebo platformy na šírenie technológií) a finančných analytikov. Títo odborníci prispievajú do expertného systému cennými znalosťami a metodikami. Okrem toho sa niektoré faktory spresňujú vstupmi z relevantných zdrojov znalostí, ako je vedecká literatúra alebo pohľady odborných organizácií, aby sa zvýšila účinnosť systému.

#### 6.1.1 Prehľad expertov

V záujme zachovania dôvernosti a rešpektovania súkromia zúčastnených osôb sú experti označení výlučne ich pozíciami, a nie ich konkrétnymi menami.

- Expert 1 – Projektový manažér a mentor v inkubátore s viac ako 15-ročnou praxou, ktorý má skúsenosti s úspešnými aj neúspešnými podnikateľskými iniciatívami.
- Expert 2 – Spoluzakladateľka a CBO technologickej startupovej spoločnosti, ktorá má hlboké znalosti marketingu a osobnú skúsenosť s faktormi ovplyvňujúcimi úspech podnikania.
- Expert 3 – Spoluzakladateľ a CEO, ktorý zastupuje veľkého investora so záujmom o investíciu do technologických startupov v štádiu "pred ziskom" a "pred príjmom".
- Expert 4 – Finančný analytik a mentor, ktorý zastupuje veľkého investora so záujmom o investíciu do finančno-technologických startupov v štádiu seed a pre-seed.



### 6.1.2 Získavanie vedomostí

Na získanie poznatkov od expertov sa použila technika neštruktúrovaného rozhovoru. Táto technika sa použila, aby experti mohli vyjadriť svoj názor slobodne a bez prvoplánových odpovedí, čím sa znížila vnímaná zaujatosť. Počas začiatku rozhovoru alebo pred rozhovorom boli experti informovaní o cieľoch expertného systému, ale nedostali žiadnu nápovedu o tom, aká metodika alebo faktory sa majú použiť. Rozhovor začal prvou otázkou, ktorá znela: „Čo je podľa vašich skúseností najdôležitejším faktorom, ktorý prispieva k úspechu začínajúcich podnikov?“. Týmto spôsobom sa mohol odborník vyjadriť k faktorom, ktoré boli preň najdôležitejšie. Následne sa dialóg rozvinul o špecifikáciu a vysvetlenie dôležitosti skutočnosti, ktorú je potrebné zohľadniť. Experti často poukazyvali na metódy a stupnice merania a hodnotenia určitých faktorov, ako je napríklad úroveň technologickej pripravenosti (TRL) alebo pomer medzi hodnotou životnosti zákazníka (LTV) a nákladmi na získanie zákazníka (CAC). Diskutovalo sa aj ohľadom rôznych metrík s cieľom objasniť najvhodnejší spôsob štruktúrovaného získavania poznatkov. Na záver rozhovoru boli respondentovi predložené vstupy predchádzajúcich expertov, aby na ne vyjadril svoj názor, čím sa naslepo prehodnotili predtým získané poznatky. Príkladom je situácia, kedy bola pôvodne navrhnutá subjektívna hodnota medzi 1 a 10 pre meranie LTV v rámci trakcie, no expert 4 upozornil na výhody použitia pomeru medzi LTV a CAC s cieľom dosiahnuť vyššiu objektivnosť. Na záver sa vyvinula metainformácia o dôležitosti faktorov s ohľadom na ich označenie expertmi. Ďalšie analyzované vzťahy medzi faktami slúžia na lepšie porozumenie argumentácie, ktorá má byť zahrnutá do bázy znalostí a jej reprezentácie.

### 6.1.3 Hodnotenie faktorov

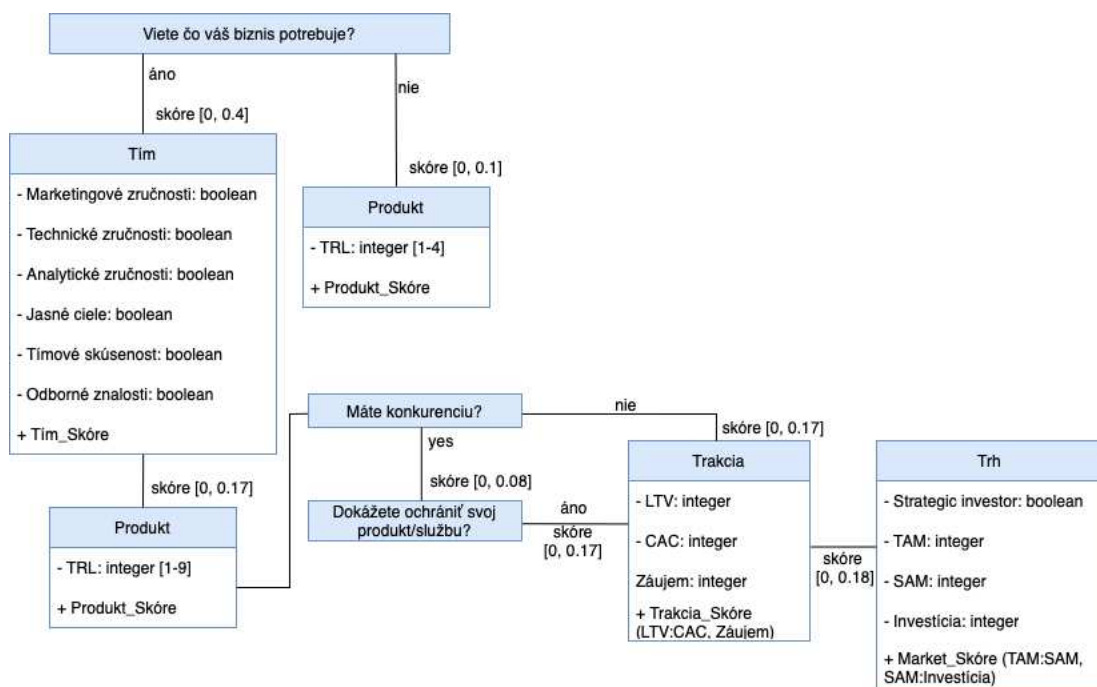
Pre 75 % odborníkov bola kľúčovou faktorovou zložkou tím. Medzi odbornými investormi sa na prvé miesto dostávala trhová príležitosť, avšak len jeden z nich dal prednosť tomuto faktoru. Pre ostatných odborníkov trhová príležitosť bola na druhom mieste. Medzi kľúčové čiastkové faktory trhových príležitostí patrila ich trakcia. Nasledoval produkt a jeho technologická pripravenosť. Konkurenčné prostredie sa považovalo za menej významný faktor, pričom len jeden expert explicitne upozornil na obranyschopnosť trhu. Najviac expertov si vybralo tím ako najdôležitejší faktor, a jeden z kľúčových finančno-technologických investorov dokonca uviedol, že až 40% celkového úspechu by hľadal v tíme. Práve preto je tím neoddeliteľnou a kľúčovou súčasťou, ktorá zohráva najvýznamnejšiu úlohu v dosiahnutí celkového úspechu.

## 6.2 Reprezentácia dát

V systéme založenom na pravidlách, je báza znalostí reprezentovaná pomocou rámcov a binárnych stromov (Obrázok 7), ktoré slúžia na organizáciu a štruktúrovanie informácií. Skladá sa z dvoch typov znalostí:

- 1) Procedurálna znalosť, ktorá popisuje spôsoby riešenia problémov.
- 2) Štruktúrálna znalosť, ktorá sa zaoberá vzťahmi medzi faktami.

Medzi rámce patria objekty ako Tím, Produkt, Duševné vlastníctvo a Trh, ktoré sú kľúčovými prvkami analytického procesu. Na druhej strane, binárne stromy ako „Viete čo váš biznis potrebuje?“ a „Máte konkurenciu?“ slúžia na usmerňovanie procesu.



Obrázok 7 Reprezentácia znalostí pomocou rámcov a binárnych stromov

Znalostiam je potrebné zaviesť príslušnú technickú formu, aby bola možná ich zrozumiteľná a efektívna reprezentácia vo forme produkčných pravidiel.

### 6.2.1 Produkčné pravidlá

Pri definovaní produkčných pravidiel je kľúčové pochopenie vstupných znalostí a ich následná efektívna konverzia do technickej podoby. Rovnako je dôležité určiť parameter „*salience*“, čo predstavuje formu priority. Pravidlá s vyššou hodnotou *salience* majú vyššiu prioritu pri zoraďovaní v aktivačnom fronte. V kontexte práce s predikciou startupu budú

základnými entitami inštancia triedy Dôkaz (angl. Evidence), Pravdepodobnosť (angl. Probability), Hypotéza (angl. Hypothesis) a Záver (angl. Conclusion). Tieto Java objekty budú reprezentovať a manipulovať s údajmi v rámci mechanizmu pravidiel Drools.

**Dôkaz (Evidence):** Táto trieda reprezentuje vstup od užívateľa. Ide o informácie alebo údaje, ktoré sú použité na podporu alebo vyvrátenie určitej hypotézy či tvrdenia. Zväčša to sú merateľné dáta, skúsenosti alebo výsledky experimentov. V technickej forme je vnímaný ako podmienková časť alebo LHS pravidla. Trieda sa skladá z atribútu popis (angl. description) a z atribútu hodnota (angl. value). Atribút popis obsahuje statickú hodnotu preddefinovaných otázok smerujúcich k používateľovi. Na druhej strane atribút hodnota nesie dynamickú informáciu, ktorú nám poskytol používateľ. Tento atribút môže nadobudnúť informáciu vo formáte áno/nie alebo čísla buď z definovanej škály alebo z škály otvorenej.

Napríklad:

- `Evidence(description == "Do you know what your business needs?", value == "yes")`
- `Evidence(description == "What is your Product Readiness Level?", value == "7")`
- `Evidence(description == "How much money do you need to enter a market?", value == "10000")`

**Hypotéza (Hypothesis):** Táto trieda predstavuje predpoklad alebo teóriu, ktorá sa navrhuje ako vysvetlenie konkrétneho produkčného pravidla. V technickej forme je vnímaná ako akciovú časť alebo RHS pravidla. Trieda obsahuje atribút popis (angl. description), ktorý slúži ako vysvetlenie, ktoré má byť pre používateľa pridanou hodnotou.

Napríklad:

- `Hypothesis h = new Hypothesis("You should consider intellectual property protection for your product/service.");`

**Pravdepodobnosť (Probability):** Táto trieda reprezentuje mieru alebo stupeň, ktorý vyjadruje ako je istá alebo neistá určitá udalosť alebo stav. V technickej forme bude reprezentovať akciovú časť RHS a rovnako aj podmienkovú časť LHS. V prípade LHS sa používa pri aktualizácií priori pravdepodobností. Trieda obsahuje atribút pravdepodobnosť (angl. probability) a atribút popis (angl. description). Atribút pravdepodobnosť je vyjadrený

číselne v rozsahu od 0 do 1, kde hodnota blížiaci sa k 0 znamená nemožnosť a hodnota blížiaci sa k 1 znamená istotu. Atribút popis vyjadruje informáciu o názve pravdepodobnosti, s ktorou je možné pracovať.

Napríklad:

- `Probability p = new Probability(0.7, "P(Traction|LTV:CAC)");`
- `Probability(description == "P(Traction|LTV:CAC)")`

**Záver (Conclusion):** Táto trieda sa vzťahuje na akciu, ktorá sa vykoná v prípade, keď je splnená podmienka pravidla. V technické forme bude reprezentovať akciovú časť RHS. Táto trieda obsahuje atribút ocenenie (angl. rate), ktorý vyjadruje pravdepodobnú hodnotu úspešnosti podniku v rozsahu od 0 do 100 a vysvetlenia (angl. explanations), ktorý obsahuje popis hypotézy, ktorá nastala.

Napríklad:

- `response.appendExplanation(new Hypothesis("Consider expanding your marketing skills").getDescription());`

### 6.3 Metóda hodnotenia podľa skóre

Metóda hodnotenia podľa skóre (angl. Scorecard Valuation Method) je spôsob určenia hodnoty startupu v rannej fáze a v počiatočnom štádiu financovania. Táto metóda je založená na porovnávaní s podobnými startupmi v rovnakom štádiu vývoja, s cieľom stanoviť odhadovanú hodnotu spoločnosti ešte pred získaním investície.

Kľúčovým prvkom tejto metódy je zhromaždenie údajov o priemernom ocenení iných startupov, ktoré boli financované v podobnom regióne a v podobnej fáze vývoja. Tieto údaje slúžia ako základ pre stanovenie hodnoty testovaného startupu.

#### 6.3.1 Zohľadnenie všetkých faktorov

Metóda hodnotenia podľa skóre zdôrazňuje potrebu zohľadnenia jedinečných faktorov, ktoré robia daný startup výnimočným. Tieto faktory môžu zahŕňať unikátne vlastnosti produktu alebo služby, skúsenosti tímu a ďalšie konkurenčné výhody. Kľúčovým krokom je priradenie hodnôt faktorom, ktoré boli identifikované expertmi. Medzi tieto oblasti patria:

- konkurenčné prostredie (0% až 8%)
- produkt (0% až 17%)

- trhov prležitost' (0% ař 35%)
- tm (0% ař 40%)

Vzhľadom na to, ře rmce trhov prležitost' a tm obsahuj dielcie faktory, je potrebn im priradiť hodnotu. Hodnoty faktoru trhov prležitost':

- strategick investor (0 ař 5%)
- trakcia (0 ař 17%)
- risk (0 ař 13%)

Hodnoty faktoru tm:

- marketingov zrunosti (0 ař 12%)
- technick zrunosti (0 ař 8%)
- analytick zrunosti (0 ař 5%)
- jasn ciele (0 ař 5%)
- tmov skusenosti (0 ař 5%)
- odborn znalosti (0 ař 5%)

### 6.3.2 Vpoet hodnotenia

Na vyhodnotenie, o koľko je testovac startup cennej neř priemer, je potrebn vykonať niekoľko krokov. Prvm krokom je priradiť kařdej funkcii hodnotenie na řkle od 1 do 10, čím sa vyjadruje jej uspenosť. Potom sa tieto hodnotenia preved na pravdepodobnosti od 0 do 1, aby bolo mořn vyjadriť ich relativnu vhu.

Nsledne sa tieto pravdepodobnosti vynsobia hodnotami prslunch faktorov alebo dielch faktorov, čo umořn zskať celkov hodnotu pre kařd hodnotiacu asť. Tieto vstupy sa potom stj, čím sa zska celkov percento, ktoré poskytne informciu o tom, či je hodnota startupu nad alebo pod priemerom. Hodnotenie niřie ako 50 % naznauje, ře je startup podpriemern, zatiaľ čo hodnotenie vyřie ako 50 % naznauje, ře je nadpriemern.

Nakoniec je mořn finlne ocenenie startupu porovnať s priemernou hodnotou, ktor bola zisten na zaiatku hodnotiaceho procesu. Tmto spsobom sa zska subjektvne, ale zroveň rozumn ocenenie testovacieho startupu.

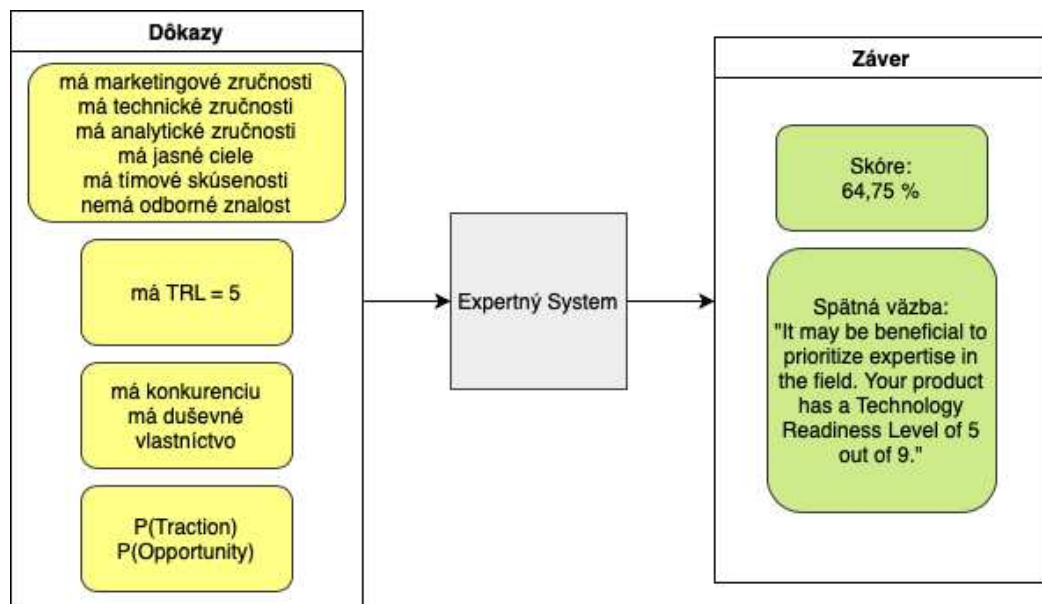
Naprklad:

- konkurenčné prostredie:  $1 * 8\% = 8\%$
- produkt:  $0.7 * 17\% = 11.9\%$
- trhova prilezitost:  $1 * 5\% + 0.6 * 17\% + 0.5 * 13\% = 21.7\%$
- tim:  $0.7 * 25\% + 1 * 5\% + 0 * 5\% + 0 * 5\% = 22.5\%$

Spolu tieto hodnoty dosahuju 64.1%, co znamena, ze startup ma o 14.1% lepsi vysledok ako priemer v danej oblasti. Ak je hodnota priemernej spoločnosti v rovnakom štadiu 1 000 000 dolarov, je mozne predpokladat, ze hodnota testovacieho startupu moze dosiahnut 1 141 000 dolarov, co je o 14.1 % viac.

## 6.4 Implementacia

Implementacna cast hodnotiaceho modulu zahrna definovanie spravnych vstupnych otazok a ich nasledne vyhodnotenie v systeme riadenia obchodnych pravidiel Drools. Nasledujuci obrazok (Obrazok 8) ilustruje, ako by tento proces mohol vyzerat.



Obrazok 8 Priklad procesu expertneho systemu

Pravidla mozu nadobudat bud deterministicku alebo pravdepodobnostnu formu. Binarne stromy sa používaju pri otazkach, ktoré maju proceduralnu povahu, ako napriklad: „Viete, co vasa firma potrebuje?“. Tuto otazku mozno preformulovat ako: „Opiste potreby vasho podniku z hladiska zručnosti a pristupu na trh“, co pomaha znit pocet falosne pozitivnych vysledkov. Položenim otvorenej otazky koncovemu používatelovi mozno poskytnut

negatívne odpovede bez negatívneho vnímania. V tomto scenári sa akýkoľvek iný vstup používateľa ako prázdny alebo „nie“ interpretuje ako „áno“.

Otázka „Viete, čo vaša firma potrebuje?“ je procedurálna, pretože ak nie sú známe obchodné potreby, používateľ nemôže odpovedať na otázky o tíme alebo trhu, pretože mu chýbajú znalosti na správne vyhodnotenie skutočností. Ak nie sú známe obchodné potreby, používateľ zodpovedá kópiu rámca produkt s maximálnou úrovňou TRL 4. Rovnaký princíp platí pre otázku „Máte konkurenciu?“, keďže experti hovoria „neexistuje startup bez konkurencie, existuje len zlý výskum“. Práve preto by odpoveď „nie“ znehodnotila akékoľvek znalosti o duševnom vlastníctve.

Rámce majú atribúty, ktoré korešpondujú s faktami a ich otázkami, operáciami alebo vzťahmi medzi faktami. Tím a produkt sú reprezentované ako rámce, zatiaľ čo trhové príležitosti boli pre jednoduchosť rozdelené na traktiu a príležitosť. Konkurenčné prostredie bolo reprezentované pomocou binárnych stromov.

Tabuľka 1 Charakteristika faktov a operácií pre reprezentáciu znalostí (podľa zohľadnenia faktorov v kap. 6.3.1)

Kľúčové slovo	Dátový typ / Výstup	Vstup / Otázka
<b>Tím 40%</b>		
Marketingové zručnosti - 12%	boolean	Máte marketingové zručnosti?
Technické zručnosti - 8%	boolean	Máte technické zručnosti?
Analytické zručnosti - 5%	boolean	Máte analytické zručnosti?
Jasné ciele - 5%	boolean	Má vaša spoločnosť jasné ciele, ktoré sú v súlade s hodnotami členov vášho tímu?
Tímové skúsenosti - 5%	boolean	Má váš tím predchádzajúce skúsenosti so spoluprácou?
Odborné znalosti - 5%	boolean	Má váš tím interné odborné znalosti alebo externú pomoc v odvetví vášho podnikania?
Skóre	[0, 0.4]	Zručnosti + Ciele + Skúsenosti + Znalosti
<b>Produkt 17%</b>		

TRL	integer	Aká je vaša úroveň technologickej pripravenosti?
Skóre	[0, 0.17]	$(TRL / 9) * 17\%$
<b>Konkurenčné prostredie 8%</b>		
Duševné vlastníctvo	boolean	Dokážete ochrániť svoj produkt/službu?
<b>Trakcia 17%</b>		
LTV	integer	Aká je životná hodnota vášho potenciálneho zákazníka?
CAC	integer	Aké sú potenciálne náklady na získanie zákazníka?
Záujem	integer	Od 1 do 10, aká je vaša miera záujmu?
$P(\text{Trakcia} \text{LTV}:\text{CAC})$	$p [0.8, 0.2]$	Pomer > 5
$P(\text{Trakcia} \text{Záujem})$	$p [0.95, 0.05]$	Pomer > 5
Skóre	[0, 0.17]	$P(\text{Trakcia}) * 17\%$
<b>Trhová príležitosť 18%</b>		
Strategický Investor - 5%	boolean	Máte strategického investora?
TAM	integer	Aký je váš TAM do 5 rokov?
SAM	integer	Aký je váš SAM do 5 rokov?
Investícia	integer	Koľko peňazí potrebujete na vstup na trh?
$P(\text{Príležitosť} \text{SAM}:\text{Investícia})$	$p [0.8, 0.2]$	Pomer > 10
$P(\text{Príležitosť} \text{TAM}:\text{SAM})$	$p [1.0, 0.0]$	Pomer > 10
Skóre	[0, 0.18]	$\text{Investor} + P(\text{Príležitosť}) * 13\%$

Rámec tímu udáva pravdepodobnosť na základe pridelených hodnôt pravdepodobnosti jednotlivým vstupom, ako je uvedené v Tabuľka 1. Tieto hodnoty pravdepodobností sa sčítajú a určujú výstup. V prípade rámca produkt sa pracuje s TRL, ktorá môže nadobúdať hodnoty od 1 do 9. V tomto prípade sa hodnoty 7 a vyššie aplikujú ako 100%, keďže odporúčanie od experta bolo, že ak startup dosiahne technologický zrelostný stupeň (TRL) 80%, má zelenú na vstup na trh. Na výpočet pravdepodobnosti úspechu pre rámce Trakcia a Trhová príležitosť sa použili pravdepodobnostné pravidlá využívajúce Bayesovskú



inferenciu. Na odvodenie posteriorných pravdepodobností rámcov sa použili rozhodovacie tabuľky. V rámci trakcie je možné vypočítať posteriornú pravdepodobnosť  $P(\text{Trakcia}|\text{LTV}:\text{CAC}|\text{Záujem})$  (Tabuľka 4) kombináciou závislých pravdepodobností  $P(\text{Trakcia}|\text{LTV}:\text{CAC})$  (Tabuľky 2) a  $P(\text{Trakcia}|\text{Záujem})$  (Tabuľky 3) prostredníctvom rozhodovacej tabuľky. Táto tabuľka spracúva extrémne pravdepodobnosti a váži stredné skóre smerom nadol.

Tabuľka 2 Pravdepodobnosť  $P(\text{LTV}:\text{CAC}|\text{Trakcia})$ 

<b>P (LTV:CAC Trakcia)</b>			
Trakcia	dobrá	zlá	Pravdepodobnosť
LTV:CAC > 5	0.8	0.2	LS = 4.0
LTV:CAC < 5	0.2	0.8	LN = 0.25

Tabuľka 3 Pravdepodobnosť  $P(\text{Záujem}|\text{Trakcia})$ 

<b>P (Záujem Trakcia)</b>			
Trakcia	dobrá	zlá	Pravdepodobnosť
Záujem > 5	0.95	0.05	LS = 19.0
Záujem < 5	0.05	0.95	LN = 0.05

Pravdepodobnosť záujmu o produkt má väčší vplyv na výsledok ako pomer LTV a CAC, pretože s počtom záujemcov sa znižuje riziko. Ak má produkt LTV len 1 dolár, je možné povedať, že nie je príliš dobrý. Naopak, ak má o produkt záujem 1 milión ľudí, potom je jeho trakcia dobrá. Avšak, ak obchodný model s LTV 1 milión dolárov má len jedného záujemcu, je možné povedať, že trakcia nie je dobrá, pretože podnik závisí od jediného kupujúceho, čo výrazne zvyšuje riziko.

Tabuľka 4 Pravdepodobnosť  $P(\text{Trakcia}|\text{LTV:CAC}|\text{Zájem})$ 

	<b>P (Trakcia LTV:CAC Zájem)</b>		
	dobrá	zlá	Pravdepodobnosť
$P(\text{Trakcia} \text{LTV:CAC}) = \text{dobrá}$ , $P(\text{Trakcia} \text{Zájem}) = \text{dobrá}$	0.9	0.1	LS = 9.0
$P(\text{Trakcia} \text{LTV:CAC}) = \text{dobrá}$ , $P(\text{Trakcia} \text{Zájem}) = \text{zlá}$	0.4	0.6	LN = 0.67
$P(\text{Trakcia} \text{LTV:CAC}) = \text{zlá}$ , $P(\text{Trakcia} \text{Zájem}) = \text{dobrá}$	0.6	0.4	LS = 1.5
$P(\text{Trakcia} \text{LTV:CAC}) = \text{zlá}$ , $P(\text{Trakcia} \text{Zájem}) = \text{zlá}$	0.1	0.9	LN = 0.11

V rámci trhovej príležitosti je možné vypočítať posteriornú pravdepodobnosť  $P(\text{Príležitosť}|\text{SAM:Investícia (TAM:SAM)})$  prostredníctvom rozhodovacej tabuľky 5.

Tabuľka 5 Pravdepodobnosť  $P(\text{SAM:Investícia}|\text{Príležitosť})$ 

Príležitosť	<b>P (SAM:Investícia Príležitosť)</b>	
	dobrá	zlá
SAM:Investícia > 10	0.8	0.2
SAM:Investícia < 10	0.2	0.8

Pravdepodobnosť úspechu na trhu (Tabuľka 6) zohľadňuje pomer medzi veľkosťou adresovateľného trhu a potrebnými investíciami na jeho dosiahnutie (SAM:Investícia) a význam pomeru celkového adresovateľného trhu ku veľkosti trhu, ktorý je momentálne dostupný (TAM:SAM). Pomer SAM:Investícia vyšší ako 10 sa považuje za pozitívny indikátor, nakoľko zabezpečuje vysokú návratnosť v krátkodobom horizonte. TAM:SAM funguje ako dôležitý faktor, ktorý ovplyvňuje konečnú pravdepodobnosť úspechu, pričom vyšší pomer môže zvýšiť šance na úspech, najmä v situáciách, kde pomer SAM:Investícia indikuje nízku hodnotu, avšak potenciál budúcej príležitosti je vysoký.

Tabuľka 6 Pravdepodobnosť  $P(\text{Príležitosť}|\text{SAM:Investícia}(\text{TAM:SAM}))$ 

	<b>P (Príležitosť SAM:Investícia(TAM:SAM))</b>		
	dobrá	zlá	Pravdepodobnosť
$P(\text{Príležitosť} \text{SAM:Investícia})$ = dobrá, TAM:SAM > 10	0.9	0.1	LS = 9.0
$P(\text{Príležitosť} \text{SAM:Investícia})$ = dobrá, TAM:SAM < 10	0.5	0.5	LN = 1.0
$P(\text{Príležitosť} \text{SAM:Investícia})$ = zlá, TAM:SAM > 10	0.4	0.6	LS = 0.67
$P(\text{Príležitosť} \text{SAM:Investícia})$ = zlá, TAM:SAM < 10	0.1	0.9	LN = 0.11

Produkčné pravidlá boli implementované v deterministickom formáte pomocou postupného reťazenia a v pravdepodobnostnom formáte pomocou spätného reťazenia. Pri postupnom reťazení pravidlá nezáviseli na žiadnych ďalších faktoroch a boli spúšťané podľa ich priority. Na druhej strane, pri použití spätného reťazenia, čo umožnilo aktualizáciu pravdepodobností nastala závislosť na vytvorení pravdepodobností v pracovnej pamäti.

## 6.5 Zhrnutie implementácie

Definovanie úspešnosti alebo neúspešnosti expertného systému s použitím metódy hodnotenia podľa skóre je náročnou úlohou. Táto metóda závisí na správnej interpretácii vstupov a odbornom posúdení faktorov v jednotlivých oblastiach. Otázky sú formulované pre rýchle pochopenie a ľahké zodpovedanie používateľom s ohľadom na svoje vedomosti o konkrétnom podniku. Posúdenia v tomto prípade vykonávajú spoluzakladatelia startupov a investiční experti, čo môže predvídateľne signalizovať určitú spoľahlivosť či expertízu. Znalosti a skúsenosti expertov sú teda kľúčovými faktormi, ktoré určujú relevanciu a úspech systému.

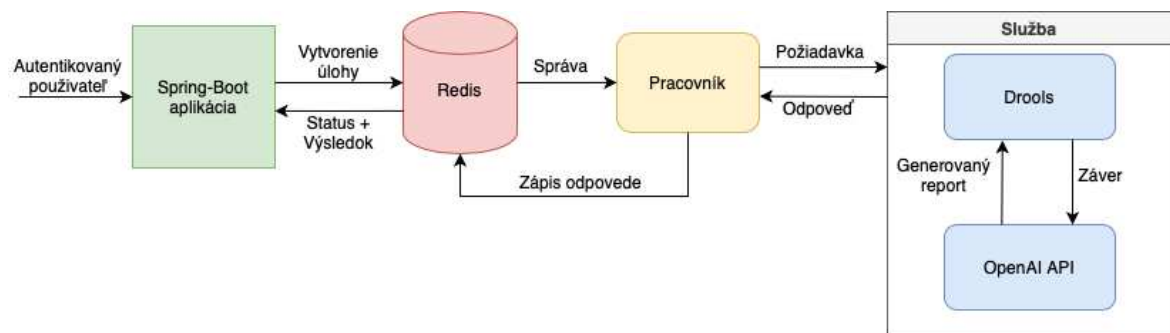
Výsledky systému sú najpresnejšie pri extrémnych hodnotách. Pri nedostatku informácií o podniku alebo nejasnosti pri vstupe sa úspešnosť systému znižuje, pretože používateľ nemôže vyjadriť svoju nedostatočnú presnosť v odpovedi. Stredné hodnoty z intervalu od 3 do 6 na škále od 1 do 10 majú nižšiu úspešnosť kvôli znižovaniu stredných hodnôt nadol.

Na zhodnotenie presnosti znalostného systému by bolo nevyhnutné využiť set historických a aktuálnych dát a následne vyvodiť záver na základe úspešnosti predchádzajúcich

hodnotení v porovnání s aktuálním stavem testovacího startupu. Čas ukáže konkrétní přesnost expertního systému, každopádně v rámci této diplomové práce byl vytvořený automatizovaný hodnotiaci proces s relativnou odchýlkou, založený na expertních znalostí.

## 7 SERVEROVÁ ČASŤ

Účelom serverovej časti systému je slúžiť ako základná stavba aplikácie, ktorá poháňa funkčnosť aplikácie. Rieši úlohy, ako je spracovanie obchodnej logiky, správa perzistencie údajov a integrácia s externými systémami. Kotlin je vybraný ako hlavný programovací jazyk pre jeho výkonné funkcie, stručnú syntax a bezproblémovú kombináciu s jazykom Java bez potreby špeciálnych nastavení. Sekundárne bol zvolený programovací jazyk Java, predovšetkým s ohľadom na integráciu Drools a manipuláciu s obchodnou logikou. Druhý Framework Spring Boot je vybraný na zefektívnenie procesu vývoja aplikácií, vzhľadom na jeho bohatý ekosystém knižníc a modulov, ktoré zjednodušujú základné funkcie, ako je vytváranie RESTful služieb, riadenie zabezpečenia, správa dát pomocou JPA a reaktívna práca s Redisom. Z pohľadu architektúry vyzerá serverová časť nasledovne (Obrázok 9).



Obrázok 9 Architektúra serverovej časti

### 7.1 Zabezpečenie pomocou JWT

Kľúčovou súčasťou vývoja modernej aplikácie je jej zabezpečenie. Medzi najznámejšie prístupy patrí zabezpečenie pomocou JSON Web Token (JWT), ktorý prenáša token pomocou HTTP hlavičky. Token JWT sa skladá z hlavičky, užitočného zaťaženia obsahujúceho prenášané údaje a podpisu. Umožňuje bezpečne autentifikovať a autorizovať používateľov pomocou generovania a overovania tokenov. V aplikácií sa generujú dva druhy tokenov: Prístupový token a token pre obnovenie.

Oba typy tokenov v sebe obsahujú informácie ako email používateľa, čas vytvorenia, expiračný čas a podpisový kľúč. Prístupový token má platnosť 2 minúty, zatiaľ čo token pre obnovenie expiruje už po 15 minútach.

Pri autentifikácii sa využíva trieda `AuthenticationManager`, ktorá spracúva vstupné údaje používateľa. V prípade úspešnej autentifikácie sa vyhledá používateľ v H2 databáze. Táto

databáza slúži ako úložisko pre používateľské údaje, ako sú emailové adresy, heslá a ďalšie. Ak sa používateľ nájde v databáze, aplikácia vygeneruje prístupový a obnovovací token. Tieto tokeny sú následne uložené do kolekcie ConcurrentHashMap, ktorá slúži ako dočasné úložisko pre aktívne tokeny. Výstupom celého procesu autentifikácie je informácia o vygenerovaných tokenoch a základné informácie o používateľovi, ako sú meno, priezvisko a emailová adresa.

Proces obnovy tokenu sa začína získaním tokenu zo vstupu metódy refreshToken(). Z tokenu sa extrahuje emailová adresa, ktorá slúži ako identifikátor používateľa v databáze. V prípade, že je získaný token platný a používateľ s danou emailovou adresou existuje v databáze, systém generuje nový prístupový a obnovovací token. Novo vygenerované tokeny sú následne uložené do kolekcie ConcurrentHashMap, ktorá slúži na uchovávanie aktívnych tokenov v pamäti. Výstupom tejto metódy sú informácie o vygenerovaných tokenoch, ktoré sú následne vrátené na výstup.

Pri inicializácii aplikácie je nevyhnutné vytvoriť administrátorského používateľa pomocou súboru data.sql, ktorý slúži ako testovací používateľ pre prihlásenie sa do systému. Je reprezentovaný entitou User a vyhľadáva sa v H2 databáze pomocou JPA repozitára.

Zabezpečenie sa zvyšuje nastavením konkrétnych zdrojov, ktoré majú povolené pristupovať k aplikácii. Konfiguračné atribúty portu a hostiteľa (angl. host) sa získavajú zo súboru premenných prostredia .env.

## 7.2 Konfigurácia Drools

Správna konfigurácia je nevyhnutná pre úspešný chod každého projektu s Drools. Je nevyhnutné, aby bol v adresári resources a podadresári META-INF prítomný súbor kmodule.xml, ktorý definuje KieBase a KieSession a konfiguruje tieto bázy znalostí a relácie. Tento súbor by mohol vyzerat' nasledovne.

```
<?xml version="1.0" encoding="UTF-8"?>
<kmodule xmlns="http://www.drools.org/xsd/kmodule">
  <kbase name="startup-valuation" packages="rules">
    <ksession name="kSession-rules"/>
  </kbase>
</kmodule>
```

Nasledujúcim krokom konfigurácie je vytvorenie Spring komponenty DroolsRuleEngine, ktorá po úspešnom skompilovaní vytvára novú KieSession z KieContainera. KieSession

obsahuje prispôsobený poslucháč udalostí a globálnu premennú, ktorá obsahuje konečný výsledok. Do tejto KieSession sa potom vložia dôkazy, ktoré prišli ako vstup do Spring aplikácie, a spustí sa metóda fireAllRules(), ktorá vykoná pravidlá na základe týchto vstupov.

## 7.3 Redis

Integrácia Redisu do Spring Boot aplikácie predstavuje efektívny spôsob optimalizácie operácií vyžadujúcich neblokujúce a asynchrónne spracovanie. Využitie Redisu umožňuje výrazné zvýšenie odozvy a škálovateľnosti aplikácie, najmä v prípade časovo náročných operácií, ako je spustenie produkčných pravidiel nad používateľovým vstupom a volanie externých koncových bodov v našom prípade OpenAI. Vzhľadom na to, že tieto operácie môžu dokopy trvať až desiatky sekúnd, je použitie Redisu vhodné.

### 7.3.1 Konfigurácia

Konfigurácia spočíva v nastavení objektu LettuceConnectionFactory, ktorému nastavíme hodnoty portu, hostiteľa a pre vyššiu úroveň zabezpečenia aj heslo. Konfiguračné parametre sú načítané zo súboru premenných prostredia .env, čo zjednodušuje údržbu a nasadenie aplikácie.

### 7.3.2 Vydavateľ, Konzument a Pracovník

V kontexte aplikácie bolo implementované asynchrónne spracovanie pomocou Redis fronty, ktoré sa skladá z troch hlavných komponentov: Vydavateľa, Odberateľa a Pracovníka.

**Vydavateľ:** PublishRedisQueueService je zodpovedný za publikovanie správ do Redis fronty. Pri príprave správy sa generuje jedinečný identifikátor pomocou funkcie randomUUID(). Následne sa vytvorí objekt typu Pair, ktorý obsahuje identifikátor a vstupný objekt vo formáte JSON. Tento objekt sa potom publikuje do zoznamu pod určitou témou pomocou metódy leftPush(topic, message).

**Pracovník:** QueueWorker je komponenta, ktorá odoberá a spracováva správy z Redis fronty. Keďže táto operácia môže byť asynchrónna, odberateľ používa metódu leftPop(topic) na získanie správy. Následne sa správa extrahuje a vykoná sa ohodnotenie podniku, pričom výsledok sa serializuje do formátu JSON. Nakoniec je výsledok uložený

späť do Redis databázy pod kľúčom úlohy. Záznam v databáze Redis by mohol vyzerat' nasledovne.

```
{"rate":7.56,"explanation":"Further analysis of your business is necessary because your current success probability is only 7.56 out of 100. This means that there are significant challenges or weaknesses in your business that need to be addressed in order to improve your chances of success. By conducting a more in-depth analysis, you can identify areas for improvement, develop strategies to overcome obstacles, and ultimately increase your likelihood of achieving success in the future."}
```

**Konzument:** `ConsumeRedisQueueService` je zodpovedný za získanie stavu a výsledku hodnotenia z Redis databázy. Služba v databáze vyhľadáva správy pomocou metódy `hasKey(jobId)`. Ak je správa nájdená, získa sa pomocou metódy `get(jobId)`. Výsledkom je stav úlohy a ohodnotenie startupu. Po úspešnom ukončení úlohy je stav označený ako `DONE` pričom evaluácia obsahuje deserializovanú odpoveď z databázy. V prípade stavov `IN_PROGRESS` a `INVALID_JOB_ID` je evaluácia označená ako nulová.

## 7.4 REST API

Každá moderná Spring aplikácia obsahuje rozhranie REST API, ktoré umožňuje vystaviť zdroje a funkcie štruktúrovaným spôsobom. Podporuje to škálovateľnosť a interoperabilitu aplikácie a uľahčuje bezproblémovú komunikáciu medzi systémami bez ohľadu na ich technológie. Aplikácia bude pracovať synchronne pomocou modelu MVC a asynchrónne pomocou modelu WebFlux, ktoré sú poskytované ekosystémom Spring. Manipulovať bude s objektmi vo formáte JSON.

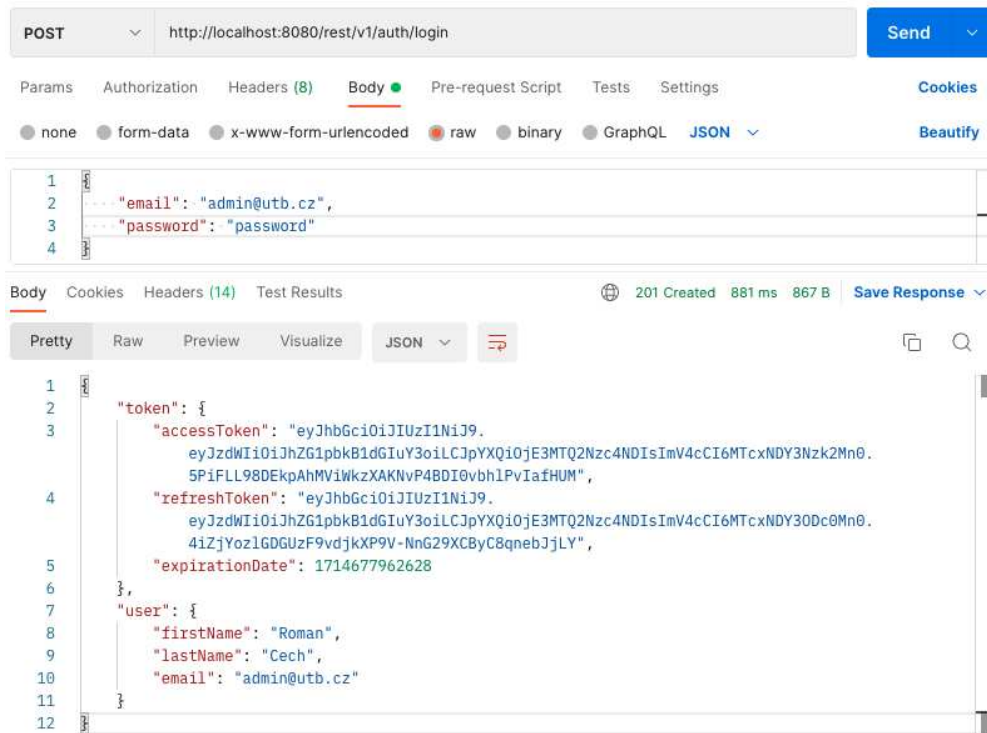
### 7.4.1 Spring MVC

Riadi sa tradičným synchronným modelom, ktorý je založený na spracovaní požiadaviek a odpovedí používajúci blokujúce vstupy a výstupy. V našom prípade bude vhodným prístupom pre prihlásenie, odhlásenie a obnovovanie tokenu, pretože je odpoveď nevyhnutná pre ďalšie fungovanie systému.

**Koncový bod:**

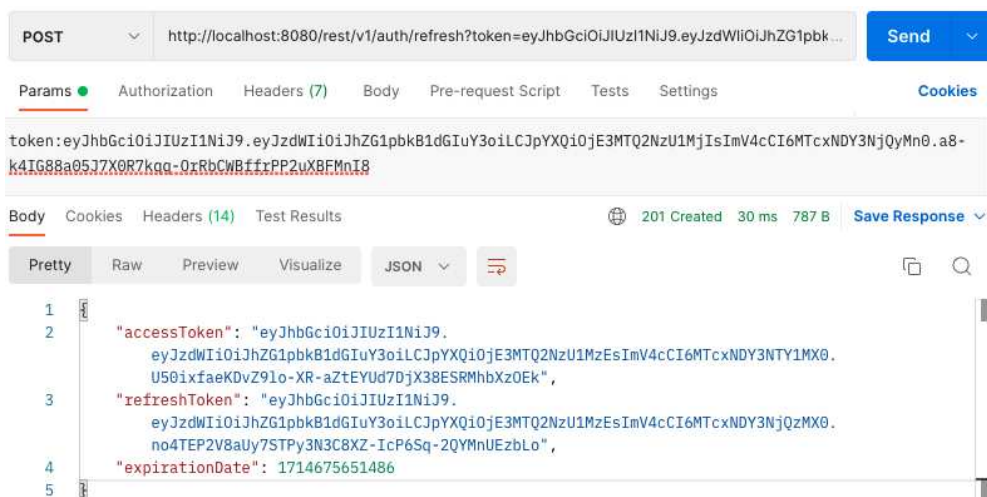
- `"/rest/v1/auth/login"` reprezentuje službu na prihlásenie, ktorá konzumuje objekt `AuthenticationRequest` a produkuje `AuthenticationResponse` (Obrázok 10).





Obrázok 10 Úspešná odpoveď po prihlásení

- `"/rest/v1/auth/refresh"` reprezentuje službu na obnovenie tokenu, ktorá konzumuje obnovovací token a produkuje objekt typu Token (Obrázok 11).



Obrázok 11 Úspešná odpoveď po obnovení tokenu

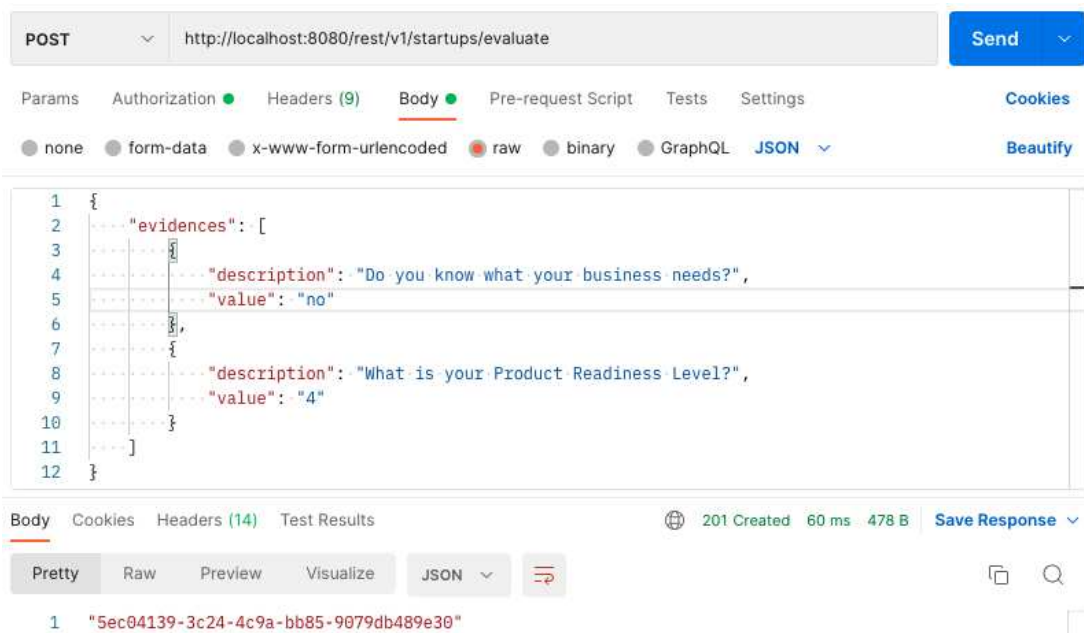
- `"/rest/v1/auth/logout"` reprezentuje službu na odhlásenie, ktorá konzumuje prístupový token a vracia prázdnu odpoveď. Tento koncový bod vyžaduje na svoje správne fungovanie platný prístupový token.

## 7.4.2 Spring WebFlux

Riadi sa reaktívnym modelom, ktorý je založený na reaktívnych princípoch a využíva neblokujúce vstupy a výstupy. Zároveň je navrhnutý pre vysokú súbežnosť a škálovateľnosť s malým počtom vlákien. V našom scenárii bude zastrešovať biznisovú funkciu hodnotenia a to znamená, že sa bude starať o vytvorenie úlohy, jej spracovanie a v neposlednom rade aj o vrátenie výsledku konzumentovi. Koncové body budú vyžadovať platný prístupový token v HTTP hlavičke.

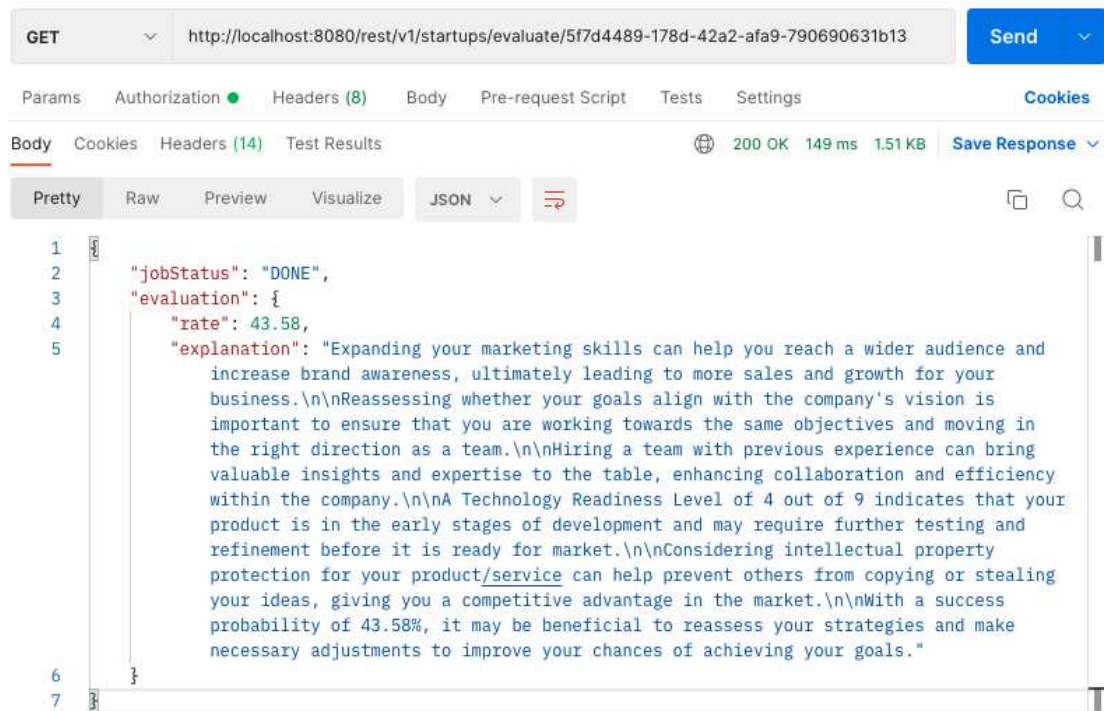
### Koncový bod:

- `"/rest/v1/startups/evaluate"` reprezentuje službu, ktorá konzumuje objekt `StartupValuationRequest` a produkuje jedinečný identifikátor úlohy (Obrázok 12).



Obrázok 12 Úspešná odpoveď po publikovaní správy do fronty

- `"/rest/v1/startups/evaluate/{jobId}"` reprezentuje službu, ktorá konzumuje jedinečný identifikátor úlohy a produkuje prispôsobený typ `Pair`, kde kľúč obsahuje stav úlohy a hodnota obsahuje objekt `StartupValuationResponse` (Obrázok 13).



Obrázok 13 Úspešná odpoveď ohodnotenia startupu

## 7.5 Kontejnerizácia

Kontejnerizácia sa stala kľúčovým prístupom pri nasadzovaní moderných aplikácií, pretože ponúka zjednodušené a konzistentné prostredie na rôznych platformách. V kontexte našej aplikácie v kombinácii Spring Boot a Redis zjednodušuje proces zapuzdrením backendovej služby a dátového úložiska v pamäti Redis do samostatných kontajnerov.

services:

startup-valuation-be:

image: startup-valuation-be-container:latest

restart: unless-stopped

depends\_on:

- redis

env\_file:

- .env

ports:

- "8080:8080"

redis:

image: redis:latest

ports:

- "6379:6379"

Táto konfigurácia Docker Compose definuje dve služby: `startup-valuation-be` pre časť implementovanú v Spring Boot a `redis` pre úložisko Redis. Každá služba je izolovaná vlastným kontajnerom, čím sa zabezpečuje škálovateľnosť a nezávislosť. Závislosti podporuje smernica `depends_on`, ktorá zabezpečuje, že služba backendu sa spustí až po inicializácii úložiska Redis.

## 7.6 Jazykový model GPT

V dynamicky rozvíjajúcej sa oblasti spracovania prirodzeného jazyka a umelej inteligencie sa veľké jazykové modely stali kľúčovými. Zvyšujú produktivitu a poskytujú nákladovo efektívne riešenia. Integrácia jazykového modelu je preto prínosná, hlavne pri funkcionalite ako je generovanie slovných reportov. Práve pre schopnosť chápať a generovať text bol zvolený model GPT-3.5. Z technického hľadiska je hlavnou výhodou moderný výkon, škálovateľnosť či jednoduchá integrácia.

### 7.6.1 Požiadavka na OpenAI API

Aby bolo možné pristupovať ku zdroju `"/chat/completion"`, je potrebné vygenerovať OpenAI API kľúč. Pre získanie kľúča je nevyhnutné sa zaregistrovať a vybrať si predplatné, ktoré ponúka prístup k API. Tento kľúč slúži ako obsah autorizačnej hlavičky v podobe Bearer tokenu a je získavaný zo súboru premenných prostredia `.env`.

Povinné atribúty tela požiadavky sú `role` a `model`. Správa obsahuje atribúty `role` (odosielateľ správy) a `content` (samotná správa). Atribút `model` obsahuje identifikátor modelu. V našom prípade je pre testovacie účely použitá verejná verzia `"gpt-3.5-turbo"` z dôvodu cenovej výhodnosti oproti verzií `"gpt-4-turbo"`.

Nad rámec povinných atribútov rozhranie, API akceptuje niekoľko voliteľných parametrov. Aby model nevygeneroval príliš náhodnú odpoveď, je nutné nastaviť teplotu vzorkovania medzi 0 a 1. Pre dosiahnutie optimálneho výsledku je zvolená hodnota 0.2.

Príklad správne vytvorenej požiadavky na OpenAI API vo formáte cURL vyzerá nasledovne (Obrázok 14).

```
1 curl https://api.openai.com/v1/chat/completions \  
2 -H "Content-Type: application/json" \  
3 -H "Authorization: Bearer $OPENAI_API_KEY" \  
4 -d '{  
5   "model": "gpt-3.5-turbo",  
6   "temperature": 0.2,  
7   "messages": [  
8     {  
9       "role": "user",  
10      "content": "Generate explanation for: $prompt"  
11    }  
12  ]  
13 }'
```

Obrázok 14 Príklad požiadavky na OpenAI

## 7.6.2 Odpoveď z OpenAI API

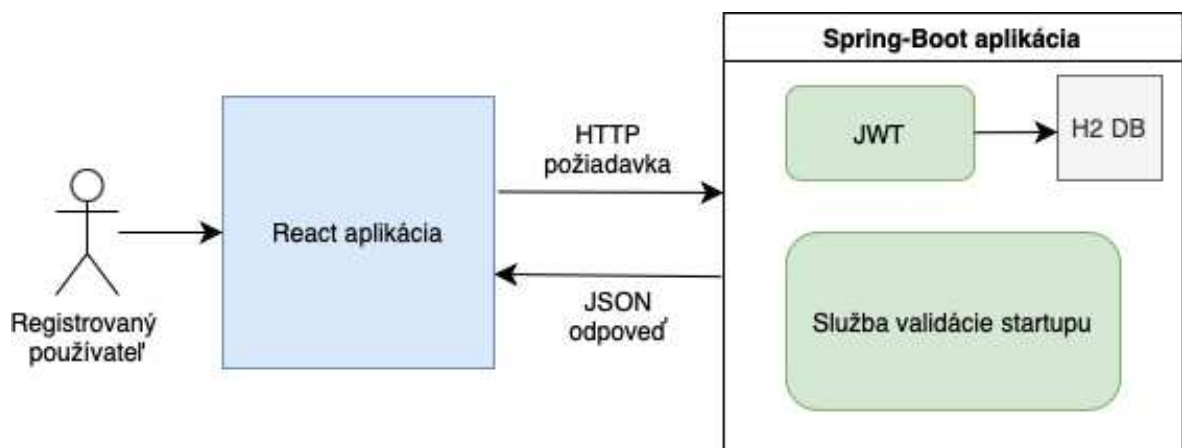
Ak je požiadavka vytvorená správne je vrátená štruktúrovaná odpoveď vo formáte JSON (Obrázok 15). V odpovedi je pre klientsku časť relevantná informácia atribútu obsah (angl. content), ktorá obsahuje generovaný slovný report.

```
"id": "chatcmpl-9KteXLipiwCEgmR1lbFevTyfMABiw",  
"object": "chat.completion",  
"created": 1714767517,  
"model": "gpt-3.5-turbo-0125",  
"choices": [  
  {  
    "index": 0,  
    "message": {  
      "role": "assistant",  
      "content": "A Technology Readiness Level (TRL) of 4 indicates that your product is in the early stages of development and testing. At this level, the basic technological components have been validated in a laboratory environment, and the feasibility of the product concept has been demonstrated. However, further development and testing are still required to demonstrate the product's functionality in a relevant environment. This stage typically involves conducting initial field tests and refining the product design based on feedback and data collected. It is important to continue advancing the technology to higher TRL levels to ensure that it is ready for commercialization and widespread adoption."  
    },  
    "logprobs": null,  
    "finish_reason": "stop"  
  }  
],  
"usage": {  
  "prompt_tokens": 27,  
  "completion_tokens": 118,  
  "total_tokens": 145  
},  
"system_fingerprint": "fp_a450710239"
```

Obrázok 15 Príklad odpovede z OpenAI

## 8 KLIENSKÁ ČASŤ

Účelom klientskej časti je poskytnúť užívateľsky prívetivé rozhranie, ktoré umožní interakciu medzi používateľom a aplikáciou a zároveň zabezpečí efektívnu komunikáciu medzi serverom a klientom. Pre dosiahnutie tohto cieľa je zvolený hlavný programovací jazyk TypeScript a javascriptový framework React. TypeScript je vybraný pre svoju schopnosť poskytnúť statické typovanie a lepšiu bezpečnosť a údržbu kódu. React je zvolený pre svoju výkonnosť, flexibilitu a schopnosť vytvoriť interaktívne a dynamické užívateľské rozhranie. Na realizáciu používateľského rozhrania bola zvolená kombinácia technológií Ant Design a Tailwind CSS. Z pohľadu architektúry vyzerá klientská časť nasledovne (Obrázok 16).



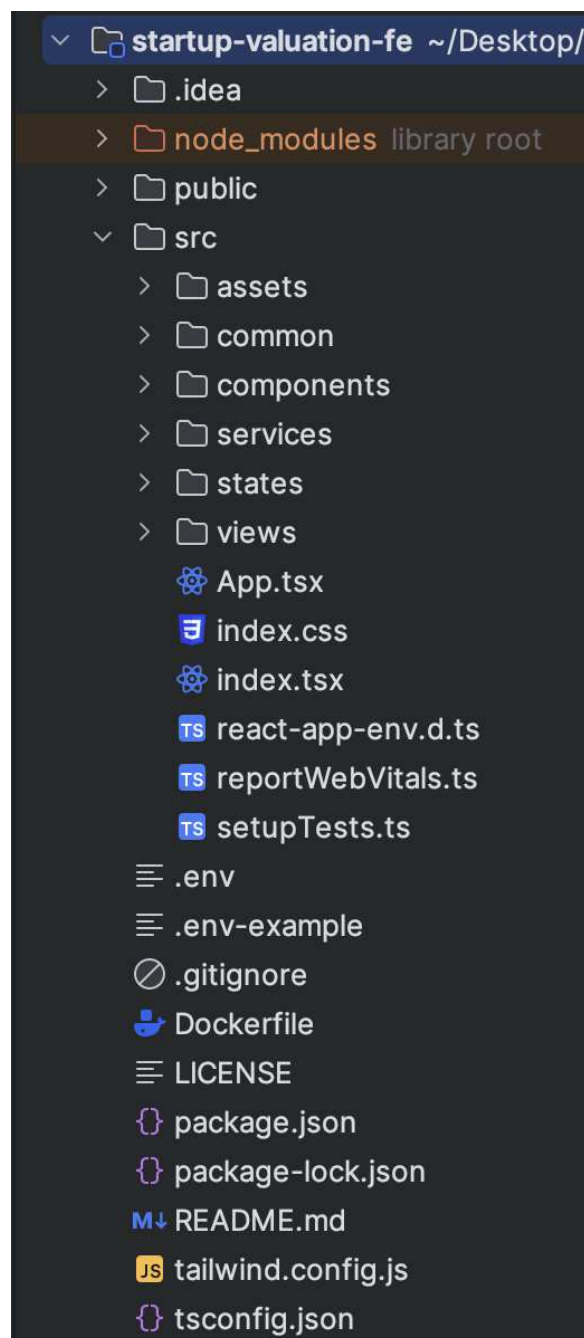
Obrázok 16 Architektúra klientskej časti

### 8.1 Štruktúra projektu

Správna štruktúra projektu (Obrázok 17) je pre frontendovú aplikáciu kľúčová hlavne pre udržiavateľnosť a škálovateľnosť. Každá takáto aplikácia zvyčajne obsahuje niekoľko dôležitých adresárov a súborov.

- `node_modules` – obsahuje nainštalované závislosti projektu. Tento adresár je automaticky generovaný po inštalácii balíčkov pomocou nástroja npm.
- `public` – obsahuje verejne prístupné aktíva ako hlavný HTML súbor, ikony a ďalšie.
- `src` – obsahuje všetok zdrojový kód aplikácie vrátane komponentov, stránok a ďalších častiach aplikácie.

- .env – definuje premenné prostredia a zahrňa senzitivne informácie, ktoré nie sú zverejňované. Vzorová verzia tohoto súboru sa nachádza v súbore .env-example.
- .gitignore – špecifikuje súbory a priečinky, ktoré chce aby Git ignoroval.
- Dockerfile – obsahuje pokyny na vytvorenie Docker obrazu pre aplikáciu React. Zapuzdruje aplikáciu a umožňuje prenosnosť aplikácie, čo pomáha pri jej distribúcií.
- tailwind.config.js – konfiguruje cesty ku všetkým šablónam na použitie Tailwind CSS.



Obrázok 17 Štruktúra projektu

Hlavný adresár src obsahuje súbor *App.tsx* a *index.tsx*. Súbor *App.tsx* definuje koreňový komponent aplikácie. Tento súbor nastavuje celkovú štruktúru aplikácie prostredníctvom hlavičky, obsahu a päty. Používa knižnicu *react-router-dom* ktorá prináša funkcie smerovania. Vstupným bodom React aplikácie je súbor *index.tsx*, ktorý je zodpovedný za vykreslenie koreňového prvku *App* do DOM a nastavuje potrebné konfigurácie, napríklad pre knižnicu *Recoil*. Okrem toho src obsahuje vnorený adresár *assets*, *common*, *components*, *services*, *states* a *views*.

### **Assets:**

Adresár *assets* obsahuje statické súbory *DownArrow.svg* a *UnderLine.svg*, ktoré sú následne vykreslené na stránke *Home*.

### **Common:**

- *Types.tsx* definuje všetky dátové typy používajúce ako odpoveď z volania API servera, konkrétne *IAuthResponse*, *IToken*, *IEvidence*, *IStartupValuationResponse* a *IPairResponse*.
- *SurveyJSON.tsx* obsahuje špecifikáciu dotazníka, ktorý je vykreslený stránkou *Survey*. Tento JSON bol dizajnovaný vo webovom rozhraní *surveyjs.io*, ktorá umožňuje vytvárať vlastné dotazníky.
- *Util.tsx* implementuje utility, ktoré slúžia na získavanie a ukladanie informácií o tokenoch z lokálneho úložiska a utilitu na transformáciu údajov na objekty *IEvidence*.

### **Components:**

- Adresár *components* obsahuje súbory určené na opakované použitie alebo tie, ktoré nezaberajú celú obrazovku.
- *AuthorizedDropDown.tsx* slúži ako rozbaľovacie menu profilu používateľa. Tento komponent sa vykreslí len v prípade, že je používateľ prihlásený. Obsahuje meno, priezvisko a email, ako aj tlačidlo pre odhlásenie.
- *Container.tsx* poskytuje kontajnerovú zložku, ktorá sa používa ako obsah každej stránky s rovnakými štylizáčnymi prvkami a obmedzeniami výšky a šírky.



- *Footer.tsx* a *Header.tsx* sú statické informačné komponenty. *Header.tsx* obsahuje navigačný systém a využíva komponentu *AuthorizedDropDown* pre interakciu s používateľom. Tieto súbory predstavujú hlavnú hlavičku a päť aplikácie a sú integrované v koreňovom komponente *App.tsx*.
- *Logout.tsx* je zodpovedný za odhlásenie používateľa zo systému. Po kliknutí používateľa na tlačítko “Log Out” v komponente *AuthorizedDropDown* sa tento súbor aktivuje a zobrazí sa komponenta *Spinner*, ktorá indikuje prebiehajúce odhlásenie.
- *MobileNavigation.tsx* sa stará o responzivitu aplikácie. V prípade, že používateľ otvoril aplikáciu na mobilnom zariadení je mu hlavička zobrazená tak, aby bola prispôbená jeho potrebám.
- *Spinner.tsx* slúži na zobrazenie animovaného spinnera na obrazovke po prihlásení, odhlásení a vyplnení dotazníka.

#### Services:

- *ApiService.tsx* slúži ako rozhranie na vykonávanie HTTP požiadaviek na server API. Zapuzdruje metódy *login(email, password)*, *logout()*, *evaluation(evidence)* a *getEvaluation(jobId)*. Táto abstrakcia pomáha pri správe API centralizovaným spôsobom.
- *AxiosInstance.tsx* slúži na jednotné vytvorenie inštancie *axios*, ktorá sa ďalej používa v rámci *ApiServices.tsx*. Zároveň nastavuje interceptory na zachytávanie odpovede v prípade, že vypršala platnosť prístupového tokenu. Ak sa tak stalo, obnoví token použitím metódy *refreshToken()* a zopakuje pôvodnú požiadavku.

#### States:

Adresár *states* obsahuje stavové atómy, ktoré slúžia na správu stavov. Atóm je jednotka stavu, ktorú môžu komponenty upravovať a čítať pomocou knižnice *Recoil*.

- *authAtom.tsx* definuje atóm, ktorý slúži používateľovi zostať prihlásený, pretože zdieľa jeho stav medzi komponentami. Tento atóm je definovaný jedinečným kľúčom “auth”.

- *evaluationAtom.tsx* definuje atóm, ktorý slúži na uloženie odpovede a prenos stavov medzi stránkou Survey a Evaluation. Tento atóm je definovaný jedinečným kľúčom “evaluation”.

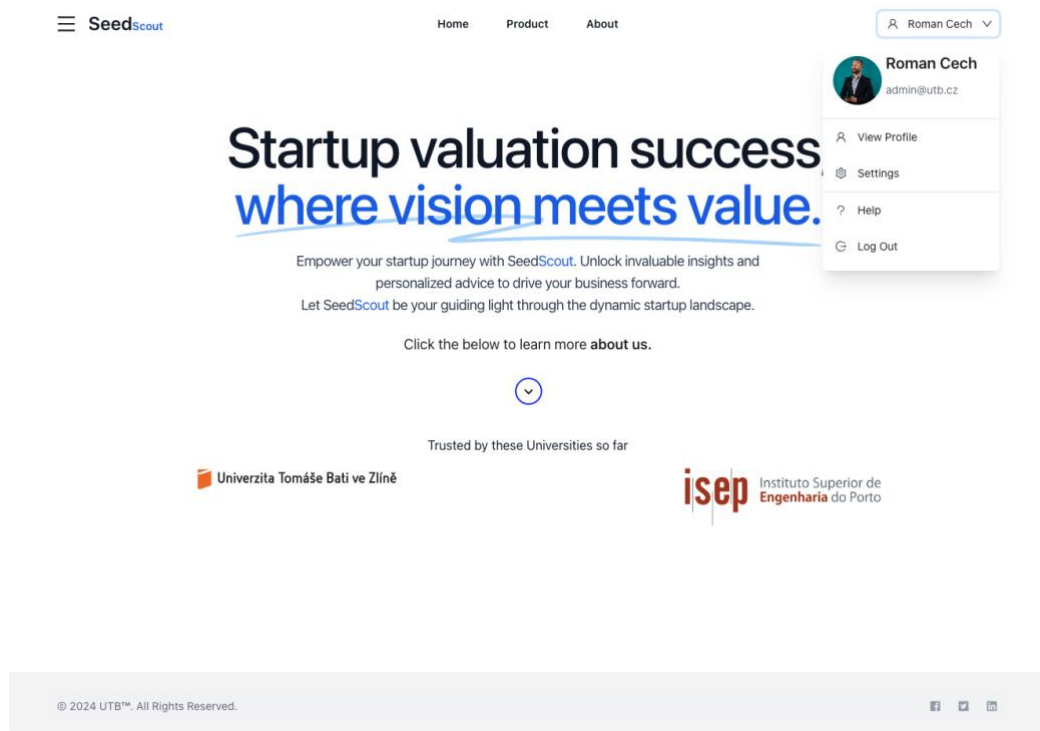
### Views:

Adresár Views obsahuje komponenty najvyššej úrovne pre rôzne stránky alebo pohľady v aplikácií. Každá stránka môže obsahovať viacero menších komponentov z adresára components.

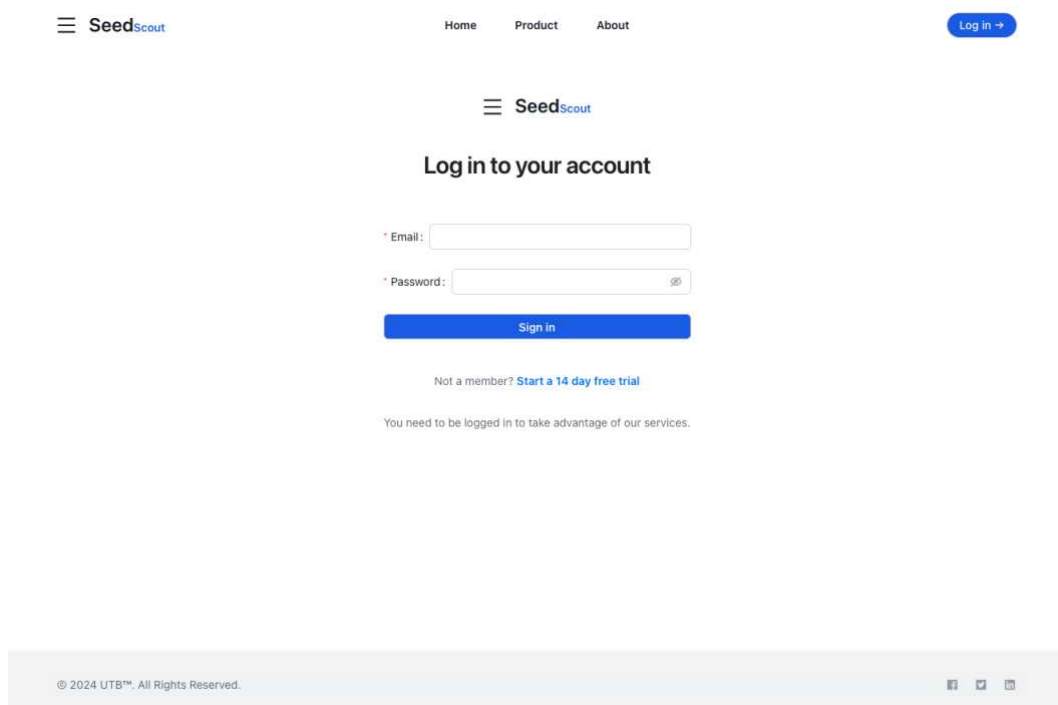
- *About.tsx* zobrazuje informácie o aplikácii SeedScout a poskytuje používateľovi popis aplikácie, definuje cieľových používateľov a na záver pozýva na účasť.
- *Evaluation.tsx* je zodpovedný za vykreslenie finálnej odpovede (Obrázok 21) v rámci ktorej zobrazuje percentuálne ohodnotenie a vysvetlenie ohodnotenia. Zároveň použitím knižnice react-to-print poskytuje používateľovi možnosť si ohodnotenie vo formáte PDF stiahnuť.
- *Home.tsx* je domovskou obrazovkou aplikácie (Obrázok 18) a je navrhnutý tak, aby čo najviac interagoval s používateľom a on stránku neopustil. Obsahuje stručné informácie a navrhuje používateľovi aby sa o aplikácii dozvedel viac na stránke “About”.
- *Login.tsx* je zodpovedný za vykresľovanie prihlasovacej stránky (Obrázok 19) a je kľúčový pre funkčnosť prihlásenia do aplikácie. Po úspešnom prihlásení je používateľ automaticky presmerovaný na stránku “Evaluate”, kde môže začať využívať aplikáciu.
- *PageNotFound.tsx* slúži na vykreslenie stránky, ktorá informuje používateľa o tom, že požadovaná stránka nebola nájdená. Nabáda ho, aby sa vrátil na domovskú stránku.
- *Survey.tsx* zabezpečuje vykresľovanie prispôbeného formulára (Obrázok 20) použitím knižnice survey-core, ktorý je určený na zhromaždenie údajov od používateľa. Po dokončení spustí proces hodnotenia a naviguje používateľa na stránku “Evaluation”, kde výsledok zobrazí.

## 8.2 Rozloženie stránky

Táto podkapitola obsahuje zobrazenie stránok, ktoré boli popísané v predošlej podkapitole.



Obrázok 18 Úvodná stránka a profil používateľ



Obrázok 19 Prihlásenie do aplikácie

Market

14. Do you have a strategic investor? \*

No Yes

15. What's your TAM within 5 years? \*

1000000

16. What's your SAM within 5 years? \*

10000

17. How much money do you need to enter a market? \*

1000

Previous Complete

© 2024 UTB™. All Rights Reserved. [f](#) [t](#) [i](#)

Obrázok 20 Stránka pred zaslaním dotazníka

43.58%

Expanding your marketing skills can help you reach a wider audience and increase brand awareness, ultimately leading to more sales and growth for your business. Reassessing whether your goals align with the company's vision is important to ensure that you are working towards the same objectives and are on the same page with the rest of the team. This alignment can lead to better communication, teamwork, and ultimately, success. Hiring a team with previous experience can bring valuable insights and expertise to the table, enhancing collaboration and efficiency within the company. This can lead to better decision-making, problem-solving, and overall performance. A Technology Readiness Level of 4 out of 9 indicates that your product is in the early stages of development and may require further testing and refinement before it is ready for market launch. Your current success probability of 43.58% out of 100 suggests that there is room for improvement in your strategies, processes, or product offerings. By identifying areas of weakness and implementing changes, you can increase your chances of success and achieve your desired outcomes.

Download PDF

© 2024 UTB™. All Rights Reserved. [f](#) [t](#) [i](#)

Obrázok 21 Hodnotiaca stránka

## ZÁVER

Cieľom diplomovej práce bolo vytvoriť expertný systém na ocenenie úspechu startupu a získanie spätnej väzby v oblastiach, ktoré je potrebné zlepšiť. Zároveň bolo cieľom vytvoriť používateľsky prívetivé rozhranie na interakciu s používateľmi.

V teoretickej časti práce bolo prezentované pochopenie a kategorizácia znalostí a taktiež ich reprezentácia, ktoré boli aplikované pri praktickom návrhu znalostnej bázy a technickej formy pravidiel. Zároveň boli vysvetlené komponenty expertného systému a jeho architektúra pre hlbšie pochopenie fungovania tohoto systému. V neposlednom rade bol vysvetlený jazyk pravidiel a mechanizmus systému Drools, ktorý bol kľúčový pri implementácií znalostnej bázy.

V praktickej časti bola predstavená konkrétna reprezentácia znalostí a zvolená metóda hodnotenia podľa skóre (angl. Scorecard Valuation Method). Keďže je oceňovanie v počiatočnom štádiu výrazne zložité a subjektívne, je použitá pravdepodobnosť úspechu namiesto pevnej hodnotovej metriky, ktorú je možno optimalizovať podľa osvedčených vzorov úspešných startupov a odborných štandardov. Na základe toho bol vytvorený expertný systém použitím Drools, ktorý odhaduje pravdepodobnosť úspechu startupu a odvodzuje podnikateľovi spätnú väzbu v oblastiach, ktoré je potrebné zlepšiť.

Celý systém sa skladá zo serverovej a klientskej časti. Serverová časť systému bola implementovaná v jazyku Kotlin a Java s využitím populárneho frameworku Spring Boot. Pomocou modulu Spring Web bolo vystavené REST API pre synchrónnu a reaktívnu komunikáciu, ktorá je zabezpečená pomocou JWT tokenu. Klientská časť systému bola implementovaná v jazyku TypeScript, pomocou frameworku React. Pre prehľadnejšie používateľské rozhranie boli využité dizajnové prvky z Ant Design a Tailwind CSS. Posledným krokom bolo zdrojový kód optimalizovať do vhodnej štruktúry, aby bolo zabezpečené dodržanie známych štandardov a postupov. Okrem toho bolo pridanie kontejnerizácie významným krokom s cieľom zjednodušiť distribúciu aplikácie a zabezpečiť konzistentné prostredie.

Výstupom práce je systém, ktorý dokáže účinne riešiť problémy a výrazne zlepšovať rozhodovacie procesy podnikateľov v oblasti začínajúcich startupov. Neposkytuje finálne rozhodnutie používateľa a ani investičné rady. Jeho úlohou je automatizovať proces ocenenia podniku, odhadnúť pravdepodobnosť úspechu a poskytnúť personalizovanú spätnú väzbu.

Táto diplomová práca by mala byť prínosom hlavne pre čitateľov, ktorí sú začínajúci podnikatelia, alebo majú záujem začať podnikat' a majú jasnú predstavu o produkte a trhu, na ktorom sa chcú uchýtiť. Táto skupina čitateľov potrebuje automatizovaný systém na vyhodnotenie svojho startupu a získanie konštruktívnej spätnej väzby, ktorú je systém schopný poskytnúť.

**ZOZNAM POUŽITEJ LITERATURY**

- [1] SEBT, Vahid Mohammad. Risk assessment modeling for knowledge based and startup projects based on feasibility studies: A Bayesian network approach. *ScienceDirect[online]*. 2021 [cit. 2024-04-30]. Dostupné z: <https://www.sciencedirect.com/science/article/abs/pii/S0950705121002550>
- [2] NURSADAAH, Maulidyah a FATUROHMAN Taufik. The Application of Risk-Based New Venture Technique for Startup Valuation. *Scholar[online]*. 2022 [cit. 2024-04-30]. Dostupné z: [http://buscompress.com/uploads/3/4/9/8/34980536/riber\\_11-1\\_05\\_t22-088\\_166-183.pdf](http://buscompress.com/uploads/3/4/9/8/34980536/riber_11-1_05_t22-088_166-183.pdf)
- [3] FESTEL Gunter., WUERMSEHER Martin a CATTANEO Giacomo. Valuation of Early Stage High-tech Start-up Companies. *Scholar[online]*. 2013 [cit. 2024-04-30]. Dostupné z: <https://ijb.cyut.edu.tw/var/file/10/1010/img/859/V183-3.pdf>
- [4] LASKOWSKI Nicole. Artificial Intelligence. *TechTarget[online]*. 2023 [cit. 2024-02-06]. Dostupné z: <https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence>
- [5] AKERKAR, Arvind Rajendra a SAJJA, Srinivas Priti. *Knowledge-based system: Introduction to Knowledge-Based Systems*. USA: Jones & Bartlett Learning, 2009. ISBN: 9781449612948. s.1-23.
- [6] WILDAN Joynathan. Knowledge Acquisition, Representation and Reasoning. *Academia[online]*. [cit. 2024-02-10] Dostupné z: [https://www.academia.edu/36366950/Knowledge\\_Acquisition\\_Representation\\_and\\_Reasoning](https://www.academia.edu/36366950/Knowledge_Acquisition_Representation_and_Reasoning)
- [7] Knowledge Representation. *DevX[online]*. 2024 [cit. 2024-02-12]. Dostupné z: <https://www.devx.com/terms/knowledge-representation/?fbclid=IwAR1Onc8Dz-DPBOg5i8KGoDgOjMFNQua3xu1zJoyuN95nZTaW0CPGzwl8uU4>
- [8] Techniques of Knowledge Representation. *javatpoint[online]*. [cit. 2024-02-16]. Dostupné z: <https://www.javatpoint.com/ai-techniques-of-knowledge-representation>
- [9] GULAVANI Sampada a KULKARNI Raja. A review of knowledge based systems in medical diagnosis. *ResearchGate[online]*. [cit. 2024-04-30]. Dostupné z: [https://www.researchgate.net/publication/228388965\\_A\\_REVIEW\\_OF\\_KNOWLEDGE\\_BASED\\_SYSTEMS\\_IN\\_MEDICAL\\_DIAGNOSIS](https://www.researchgate.net/publication/228388965_A_REVIEW_OF_KNOWLEDGE_BASED_SYSTEMS_IN_MEDICAL_DIAGNOSIS)

- [10] GUPTA, Itisha a NAGPAL, Garima. *Artificial Intelligence and Expert systems: Expert Systems*. USA: Mercury Learning and Information LLC. 2020. ISBN: 978-1-68392-507-1. s. 71-86.
- [11] ORAKZAI Rahmat. What are Expert Systems?. *Baeldung[online]*. 2023 [cit. 2024-02-13]. Dostupné z: <https://www.baeldung.com/cs/expert-systems>
- [12] Drools. *Drools[online]*. [cit. 2024-02-17]. Dostupné z: <https://www.drools.org/>
- [13] SALATINO, Mauricio; DE MAIO, Mariano; ALIVERTI, Esteban. *Mastering jboss drools 6: Drools Runtime*. Birmingham: Packt Publishing. 2016. ISBN: 978-1-78328-862-5. s. 41-60.
- [14] LAKE Rebecca. What is Large Language Model (LLM)? *Investopedia[online]*. 2023 [cit. 2024-04-30]. Dostupné z: <https://www.investopedia.com/large-language-model-7563532>
- [15] Large Language Models (LLM) : Everything you need to know. *DataScientest[online]*. 2023 [cit. 2024-04-30]. Dostupné z: <https://datascientest.com/en/large-language-models-llm-everything-you-need-to-know>
- [16] ROUSE Margaret. Large Language Model (LLM). *Technopedia[online]*. 2024 [cit. 2024-04-30]. Dostupné z: <https://www.techopedia.com/definition/34948/large-language-model-llm>
- [17] Large Language Model (LLM). *Microsoft[online]*. [cit. 2024-04-30]. Dostupné z: <https://microsoft.github.io/Workshop-Interact-with-OpenAI-models/llms/>



**ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK**

BN	Bayesian network
AI	Artificial Intelligence
KBS	Knowledge Based System
LHS	Left Hand Side
RHS	Right Hand Side
BRMS	Business Rule Management System
BRE	Business Rules Engine
DMN	Decision Model and Notation
IDE	Integrated Development Environment
JVM	Java Virtual Machine
DRL	Drools Rule Language
LLM	Large Logical Model
GPT	Generative Pre-trained Transformer
CBO	Chief Business Officer
CEO	Chief Executive Officer
TRL	Technology Readiness Level
LTV	Lifetime Value of Customer
CAC	Customer Acquisition Cost
TAM	Total Addressable Market
SAM	Serviceable Addressable Market
JPA	Java Persistence API
JWT	JSON Web Token
HTTP	Hypertext Transfer Protocol
cURL	client Uniform Resource Locator
JSON	JavaScript Object Notation

REST	Representational State Transfer
API	Application Programming Interface
MVC	Model View Controller
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
DOM	Document Object Model
PDF	Portable Network Graphics

**ZOZNAM OBRÁZKOV**

Obrázok 1 Architektúra znalostných systémov .....	17
Obrázok 2 Príklad inferenčného reťazca [10].....	21
Obrázok 3 Fungovanie mechanizmu skrz postup Match-Fire [10] .....	22
Obrázok 4 Postupné reťazenie [10] .....	23
Obrázok 5 Spätné reťazenie [10] .....	24
Obrázok 6 Prehľad komponentov mechanizmu pravidiel [12].....	27
Obrázok 7 Reprezentácia znalostí pomocou rámcov a binárnych stromov .....	34
Obrázok 8 Príklad procesu expertného systému .....	38
Obrázok 9 Architektúra serverovej časti .....	45
Obrázok 10 Úspešná odpoveď po prihlásení .....	49
Obrázok 11 Úspešná odpoveď po obnovení tokenu .....	49
Obrázok 12 Úspešná odpoveď po publikovaní správy do fronty .....	50
Obrázok 13 Úspešná odpoveď ohodnotenia startupu .....	51
Obrázok 14 Príklad požiadavky na OpenAI .....	53
Obrázok 15 Príklad odpovede z OpenAI .....	53
Obrázok 16 Architektúra klientskej časti.....	54
Obrázok 17 Štruktúra projektu .....	55
Obrázok 18 Úvodná stránka a profil používateľ.....	59
Obrázok 19 Prihlásenie do aplikácie .....	59
Obrázok 20 Stránka pred zaslaním dotazníka.....	60
Obrázok 21 Hodnotiaca stránka.....	60

**ZOZNAM TABULIEK**

Tabuľka 1 Charakteristika faktov a operácií pre reprezentáciu znalostí (podľa zohľadnenia faktorov v kap. 6.3.1) .....	39
Tabuľka 2 Pravdepodobnosť $P(\text{LTV:CAC} \text{Trakcia})$ .....	41
Tabuľka 3 Pravdepodobnosť $P(\text{Záujem} \text{Trakcia})$ .....	41
Tabuľka 4 Pravdepodobnosť $P(\text{Trakcia} \text{LTV:CAC} \text{Záujem})$ .....	42
Tabuľka 5 Pravdepodobnosť $P(\text{SAM:Investícia} \text{Príležitosť})$ .....	42
Tabuľka 6 Pravdepodobnosť $P(\text{Príležitosť} \text{SAM:Investícia}(\text{TAM:SAM}))$ .....	43