



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

FUZZY LOGIKA A PODPORA ROZHODOVÁNÍ PRO INVESTIČNÍ ZÁMĚR VE STAVEBNICTVÍ

FUZZY LOGIC AND DECISION SUPPORT FOR INVESTMENT IN BUILDING PROJECTS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Tomáš Píštěk

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. Petr Dostál, CSc.

BRNO 2017

Zadání diplomové práce

Ústav: Ústav informatiky
Student: **Bc. Tomáš Pištěk**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Informační management
Vedoucí práce: **prof. Ing. Petr Dostál, CSc.**
Akademický rok: 2016/17

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Fuzzy logika a podpora rozhodování pro investiční záměr ve stavebnictví

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza současného stavu
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Cílem diplomové práce je navrhnout a vytvořit podpůrný nástroj, pro hodnocení investičního záměru do nemovitostí, který bude založen na principech fuzzy logiky. Výsledná aplikace vyhodnotí dostupné nabídky vybraných realitních kanceláří a uživatelé poskytnou doporučení ke koupi vhodné nemovitosti, v podobě seznamu ohodnocených nabídek.

Základní literární prameny:

DOSTÁL, P. Pokročilé metody rozhodování v podnikatelství a veřejné správě. Brno: CERM, 2012. 718 s. ISBN 978-80-7204-798-7.

DOSTÁL, P. Advanced Decision Making in Business and Public Services. Brno: CERM, 2011. 168 s. ISBN 978-80-7204-747-5.

HANSELMAN, D. a B. LITTLEFIELD. Mastering MATLAB. New Jersey: Pearson Education International Ltd., 2012. 852 s. ISBN 978-0-13-185714-2.

MAŘÍK, V., O. ŠTĚPÁNKOVÁ a J. LAŽANSKÝ. Umělá inteligence. Praha: ACADEMIA, 2013. 2473 s. ISBN 978-80-200-2276-9.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2016/17.

V Brně, dne 28. 2. 2017



B. Půža

doc. RNDr. Bedřich Půža, CSc.
ředitel

Stanislav Škapa

doc. Ing. et Ing. Stanislav Škapa, Ph.D.
děkan

ABSTRAKT

Tato diplomová práce se zaměřuje na návrh a vytvoření podpůrného nástroje pro hodnocení investičního záměru do nemovitostí. Hodnocení realit je založeno na principu fuzzy logiky. Návrh a ladění fuzzy modelů je provedeno pomocí programu MATLAB. Výsledná aplikace je však vyvinuta v jazyku Java.

ABSTRACT

This master's thesis focuses on the design and creation of a decision support tool for the investment property plan. The evaluation of real estate is based on fuzzy logic. The design and debugging of fuzzy models are done by using MATLAB, but the application is developed in Java.

KLÍČOVÁ SLOVA

Fuzzy logika, MATLAB, Java, jFuzzyLogic, hodnocení nemovitostí

KEYWORDS

Fuzzy logic, MATLAB, Java, jFuzzyLogic, evaluation of real estate

BIBLIOGRAFICKÁ CITACE

PÍŠTĚK, T. *Fuzzy logika a podpora rozhodování pro investiční záměr ve stavebnictví*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2017. 81 s.
Vedoucí diplomové práce prof. Ing. Petr Dostál, CSc.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 24. května 2017

.....

Podpis studenta

PODĚKOVÁNÍ

Chtěl bych poděkovat vedoucímu diplomové práce prof. Ing. Petru Dostálovi, CSc., za cenné rady, připomínky a metodické vedení této práce.

OBSAH

ÚVOD	10
1 CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ	11
1.1 Cíle práce	11
1.2 Metody a postupy zpracování	11
2 TEORETICKÁ VÝCHODISKA	12
2.1 Programovací jazyk Java.....	12
2.1.1 Java FX	12
2.1.2 Jsoup	13
2.1.3 jFuzzyLogic	13
2.1.4 Java Persistence API.....	14
2.1.5 Hibernate.....	16
2.1.6 JSON.....	17
2.2 Umělá inteligence.....	17
2.2.1 Fuzzy logika.....	18
3 ANALÝZA SOUČASNÉHO STAVU	24
3.1 Základní informace o firmě.....	24
3.1.1 Vstup na trh realit	25
3.1.2 Rentabilita nového podnikatelského záměru	25
3.1.3 Riziko konkurence	28
3.2 Situace na trhu nemovitostí a hypoték	29
3.2.1 Výhledy do budoucna	31
3.3 Analýza sledovaných atributů nemovitostí	32
3.4 Shrnutí a zhodnocení.....	35
4 VLASTNÍ NÁVRHY ŘEŠENÍ	37

4.1	Návrh řešení aplikace	37
4.2	Zdroj dat	38
4.3	Získání dat	39
4.3.1	Získání potřebných dat z realitního serveru Sreality.cz	42
4.3.2	Získání dat o vzdálenosti pomocí Google Maps API	46
4.4	Ukládání dat	47
4.4.1	Entito-relační diagram	48
4.4.2	Řešení perzistence dat v aplikaci	51
4.4.3	Aktualizace dat	51
4.5	Návrh a vytvoření fuzzy modelů	52
4.5.1	Fuzzy model pro hodnocení domů	53
4.5.2	Fuzzy model pro hodnocení bytů	62
4.5.3	Vytvoření fuzzy modelů pomocí knihovny jFuzyLogic	63
4.6	Výsledná hotová aplikace	67
4.7	Návrhy do budoucna	70
4.8	Přínosy práce	71
	ZÁVĚR	72
	SEZNAM POUŽITÉ LITERATURY	74
	SEZNAM OBRÁZKŮ	79
	SEZNAM TABULEK	80
	SEZNAM GRAFŮ	80
	SEZNAM POUŽITÝCH ZKRATEK	81

ÚVOD

Veškeré živé organismy na světě se střetávají s konkurencí, ať se již jedná o konkurenty vlastního druhu nebo o konkurenty druhů jiných, vždy je cílem vlastní přežití a předání své genetické informace potomkům. Díky náhodným mutacím v DNA může získat organismus potřebnou konkurenční výhodu. Přírozený výběr, na jehož principu je založena evoluce, je neúprosný, přežije vždy ten jedinec, který se dokáže nejlépe adaptovat.

Na firmu je do jisté míry možné také pohlížet jako na určitý druh organismu, který se potýká s konkurencí a snaží se přežít. Zatímco u živých organismů jsou změny v DNA náhodným procesem, ve firmách je tomu jinak, veškeré procesy jsou řízeny (alespoň by tak tomu mělo být). Jedna věc je však stejná, provedená změna by měla přinést konkurenční výhodu, pokud tomu tak není a změna je spíše nevýhodou, organismus nepřežije, firma zanikne.

V současné době zaznamenáváme rozmach informačních technologií, které pronikají do všech oborů podnikání. Pokročilé algoritmy a nástroje umělé inteligence mohou firmě výrazně pomoci v neúprosném konkurenčním boji. Pokud firma nedokáže držet krok s dobou a tyto technologie nevyužije ve svůj prospěch, znamená to, že se nedokázala adaptovat a čeká jí nevyhnutelný zánik.

Jedním z mnoha nástrojů umělé inteligence je také fuzzy logika. O tom, jak jí využít a vytvořit podpůrný nástroj, který firmě pomůže s novým podnikatelským záměrem, pojednává tato diplomová práce.

1 CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ

1.1 Cíle práce

Cílem diplomové práce je navrhnout a vytvořit podpůrný nástroj pro hodnocení investičního záměru do nemovitostí, který bude založen na principech fuzzy logiky. Výsledná aplikace vyhodnotí dostupné nabídky vybraných realitních kanceláří a uživateli poskytne doporučení ke koupi vhodné nemovitosti v podobě seznamu ohodnocených nabídek.

Mezi dílčí cíle pak patří analýza potřebných atributů pro tvorbu fuzzy modelu, návrh a vytvoření samotného fuzzy modelu, zajištění adekvátního zdroje dat pro hodnocení a nakonec také návrh a vytvoření grafického uživatelského rozhraní.

1.2 Metody a postupy zpracování

Aby bylo možné výše vytyčené cíle uskutečnit, bude zapotřebí použít při vypracovávání diplomové práce různé metodiky. V první kapitole bude popsána nezbytná teorie, která se k této práci váže.

Ve druhé kapitole bude představen předmět podnikání a nový investiční záměr vybrané firmy, pro kterou bude podpůrná aplikace určena. Dále zde bude popsána situace na trhu nemovitostí a hypoték. Součástí této kapitoly bude rovněž analýza atributů nemovitostí, která následně poslouží jako podklad při návrhu fuzzy modelu.

Poslední kapitola již bude věnována samotnému návrhu a vytvoření aplikace. Nejprve bude nutné provést analýzu možných zdrojů pro data a nalézt nejvhodnější způsob jejich získávání. Na základě již zmíněné analýzy potřebných atributů pak bude vytvořen Entito-relační diagram a bude zde vyřešena otázka perzistence dat. Hodnocení nemovitostí pak bude realizováno pomocí vhodně navrženého fuzzy modelu. Posledním krokem pak bude vytvoření grafického uživatelského rozhraní aplikace.

2 TEORETICKÁ VÝCHODISKA

2.1 Programovací jazyk Java

Java je objektově orientovaný programovací jazyk, který je platformě nezávislý. Jedná se o interpretovaný jazyk, kompilátor Javy nevytváří přímo spustitelný kód, ale tzn. bajt kód. Výsledné programy lze spouštět na všech platformách, pro které je vytvořen virtuální stroj – Java Virtual Machine.

Jazyk Java vychází z jazyků C a C++, odkud přebírá podobnou syntaxi a objektový model (C++). Po vzniku Javy přišla společnost Microsoft s jazykem C#, oba tyto jazyky jsou v úzkém příbuzenském vztahu. I když se zde nachází pochopitelně různé rozdíly, v zásadě platí, že programátor znající jazyk Java, bude rozumět i jazyku C# (případně se jej naučí velice snadno), totéž platí i opačně [1].

2.1.1 Java FX

Původním grafickým podsystémem jazyka Java byla sada *AWT*, která obsahuje základní množinu komponent, které umožňují vytvářet dosti omezené grafické uživatelské rozhraní. Komponenty sady AWT se označují jako těžké, poněvadž vzhled a ovládání těchto komponent závisí na dané platformě (nikoliv na jazyce Java), pracují tedy s nativním kódem. Dochází zde k převodu grafických komponent na jejich odpovídající protějšky v dané platformě, což může vést nejen k odlišnému vzhledu, ale také k odlišnému chování, specifické pro danou platformu.

Odstranění výše zmíněných nedostatků přinesla technologie *Swing*. Toto rozhraní má, až na několik málo výjimek, lehké komponenty, které jsou napsány v jazyce Java. Celkový vzhled i chování tedy kompletně závisí na rozhraní Swing, což poskytuje očekávané chování napříč platformami. Grafický vzhled a chování komponent lze pak oddělit od business logiky, poněvadž za vše je odpovědný kód napsaný v jazyku Java [1].

Nejnovější grafickou architekturou je však *JavaFX*, která pracuje rovněž s lehkými komponentami, přesto se však technologie Swing bude ještě několik let hojně používat. JavaFX je architektura, která umožňuje vytvářet bohaté klientské aplikace, které fungují napříč různými platformami. Vzhled JavaFX aplikací lze snadno

přizpůsobit, samotná implementace aplikace je od grafického návrhu oddělena, grafici se tak mohou soustředit na design uživatelského rozhraní pomocí *FXML* (obdoba XML) a vývojáři na back-end logiku. Grafický návrh lze však provést i bez psaní kódu, k tomuto účelu slouží Scene Builder, který výsledný FXML soubor vygeneruje na základě grafického návrhu. Tento návrh se vytváří skládáním jednotlivých grafických komponent. Rozhraní JavaFX je plně integrovanou součástí JRE a JDK [1, 2].

2.1.2 Jsoup

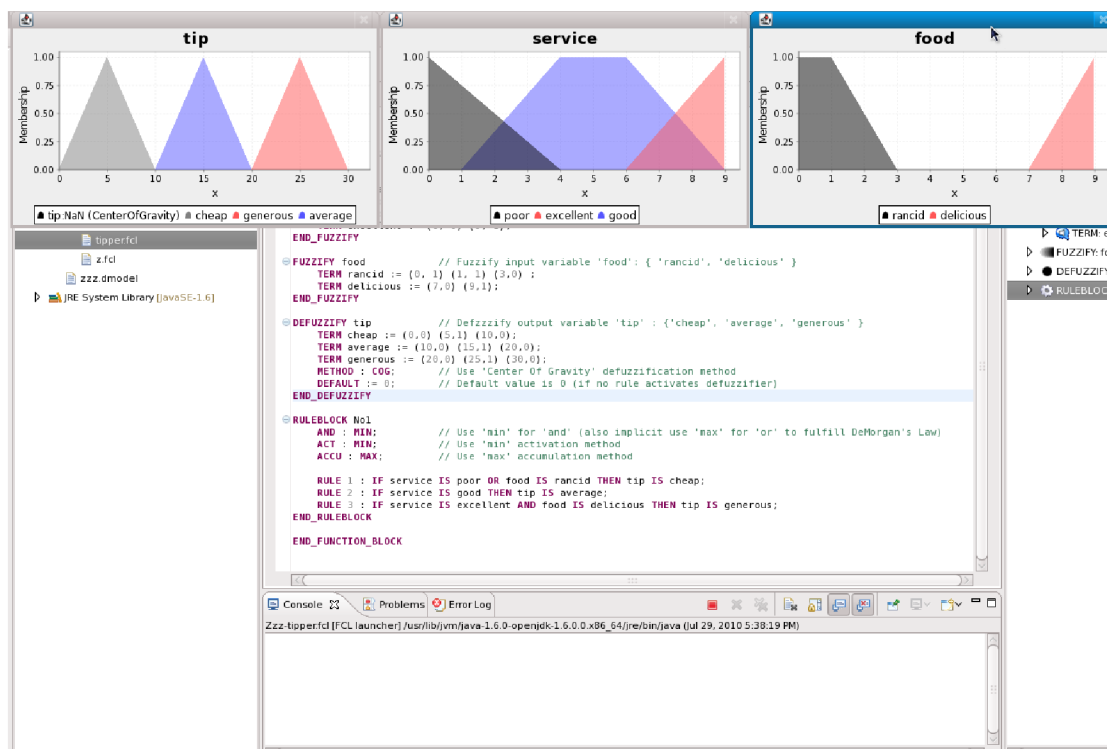
Jsoup je open source knihovna, určená pro jazyk Java, která poskytuje propracované API rozhraní pro extrakci a manipulaci s daty, uloženými v *HTML dokumentech*. V zjednodušeném podání lze Jsoup charakterizovat jako knihovnu pro parsování HTML dokumentů. Za tímto účelem využívá technologie, jakými jsou *DOM*, *CSS* a *jquery* [3].

Funkcionalitu této knihovny lze tedy shrnout do následujících třech bodů [3]:

- Parsovat HTML dokumenty z URL adresy, textového řetězce nebo souboru
- Nalézt a extrahovat data pomocí DOM nebo CSS selektorů
- Manipulovat s HTML elementy, atributy a textem

2.1.3 jFuzzyLogic

Jedná se o open source knihovnu, vyvíjenou pro jazyk Java, která implementuje průmyslové standardy pro zjednodušení vývoje fuzzy systémů. Velkou výhodou této knihovny je, že poskytuje základní programovací elementy *normy IEC 61131-7*, lze tedy využít výhod jak otevřeného softwaru, tak standardizace. Usnadnění a především urychlení vývoje fuzzy systémů je možné díky použití *standardního jazyka FCL*. Nabízený plugin pro vývojové prostředí Eclipse usnadňuje psaní a testování FCL kódu (kód zbarvuje a graficky ukazuje funkce členství) viz *obrázek 2.1* [4, 5].



Obrázek 2.1: Ukázka pluginu pro vývojové prostředí Eclipse

(Zdroj: [4])

2.1.4 Java Persistence API

Java Persistence API (JPA) je specifikací, která řeší v jazyce Java persistenci dat. V jazyce Java se pracuje s objekty, jejichž životnost je úzce svázána s během aplikace. Po skončení používání aplikace se veškerá využívaná paměť uvolní a je poskytnuta jiným aplikacím. Používané objekty tedy po skončení běhu aplikace doslova zmizí, k tomu, aby se uchovaly, je potřeba je uložit do nějakého persistentního úložiště.

V dnešní době je mnoho dat, ať se již týkají obchodní logiky apod., ukládáno do databází. Současným trendem je (i když se stále více rozšiřuje používání NoSQL databází) hojně využívání databází, založených na relačním modelu. Zjednodušeně řečeno, v relačních databázích jsou data ukládána do tabulek, které jsou tvořeny řádky a sloupci. Každý záznam je jednoznačně identifikovatelný pomocí primárního klíče (speciální sloupec), který musí být unikátní. Vztahy mezi tabulkami jsou pak řešeny pomocí cizích klíčů, které umožňují spojení záznamů jednotlivých tabulek [6].

Výše uvedený popis si však nekladl za cíl podrobně pospat fungování relačních databází, ale pouze poukázat na nesoulad mezi objekty v jazyce Java a výše uvedenými

pojmy. K vyřešení tohoto problému se používá tzv. **Objektově-Relační mapování** (ORM). Tato technologie umožňuje namapovat objekty, používané v objektově orientovaných programovacích jazycích, do relační databáze, čímž umožňuje vzájemný převod mezi odlišným reprezentováním entit reálného světa. Při ukládání objektů do relačních tabulek jsme však vystaveni těmto problémům [7]:

- **Dědičnost** – běžná věc v objektově orientovaných programovacích jazycích, v relačních databázích se však s něčím takovým nesetkáme (alespoň ne v těch standardizovaných).
- **Identita** – v relačních databázích je identita (a tedy „stejnost“) výhradně definována a určena pomocí primárního klíče, v jazyce Java však rozeznáváme dva druhy porovnávání objektů, tedy i dva druhy identity. Jednak pomocí metody „equals“, ale také pomocí operátoru „==“.
- **Asociace** – v objektově orientovaných jazycích jsou vazby mezi objekty reprezentovány jednosměrnou referencí, v relačních databázích je však vazba mezi entitami uskutečněna pomocí cizího klíče. Pro obousměrný vztah je tak potřeba v jazyce Java definovat asociaci dvakrát.
- **Datová navigace** – přístup k datům v jazyce Java je diametrálně odlišný od způsobu, jakým se to provádí v relačních databázích. V Javě přejdeme z jedné asociace na další průchodem sítí objektů. Tento způsob získávání dat však není efektivní ve světě relačních databází, kde chceme minimalizovat počet SQL dotazů. Než začneme procházet síť objektů, nejprve načteme několik entit přes spojení pomocí operací JOIN a provedeme výběr cílových entit.
- **Granularita (Zrnitost)** – někdy se můžeme setkat s případem, kdy objektový model má více tříd, než je počet odpovídajících tabulek v databázi. Jinými slovy, objektový model je daleko „přesnější“, než model relační.

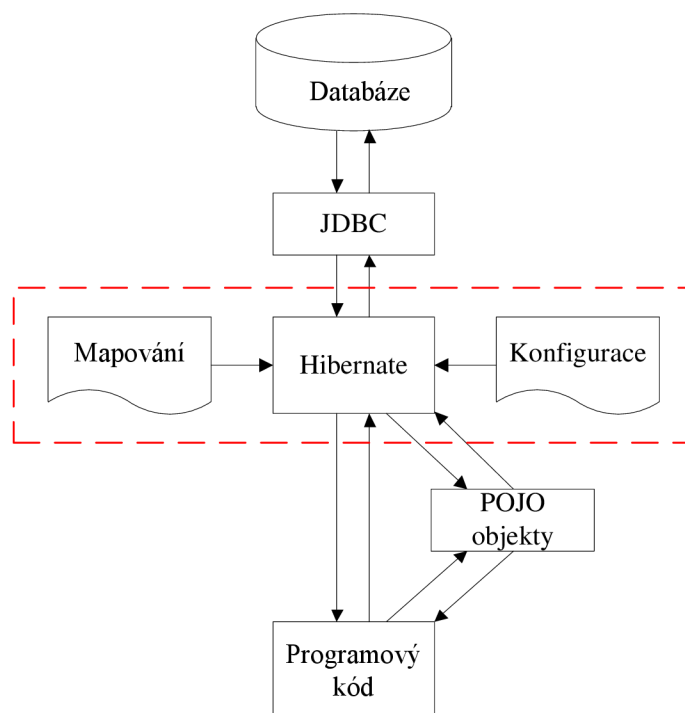
Jak již bylo nastíněno hned v úvodu, JPA je pouze specifikací, existuje však několik frameworků, které tuto specifikaci implementují, jako příklady lze uvést Hibernate, TopLink a Java Data Objects (JDO) [6].

2.1.5 Hibernate

Hibernate je framework, který zjednodušuje používání relačních databází v Java aplikacích. Poskytuje dva druhy programového rozhraní, jedná se o rozhraní „nativního Hibertane“ a standard Javy EE (Java Persistence API).

V závislosti na situaci a konkrétním problému, se pro některá řešení hodí ORM mapování, které poskytuje např. Hibernate, a pro některá je zase vhodnější použít tradiční postup v podobě přímého přístupu přes Java Database Connectivity (JDBC) API.

Role frameworku Hibernate v Java aplikaci je znázorněna na **obrázku 2.2**, jak je patrné, základním cílem je ukládat POJO objekty do databáze. Hranice toho frameworku je zvýrazněna červenou čárkovanou čarou. Údaje o mapování zjednodušeně řečeno představují informaci, které tabulky se vztahují k jakým objektům. Mapování lze provést jednak prostřednictvím anotací nebo pomocí XML souboru. Je nutné však podotknout, že kromě ORM mapování, lze také použít pro perzistenci dat NoSQL databáze, poněvadž Hibernate podporuje i OGM mapování (Object-Graph Mapping) [8].



Obrázek 2.2: Role frameworku Hibernate v java aplikaci

(Zdroj: Upraveno dle [8])

2.1.6 JSON

JSON je standardizovaný formát pro výměnu dat. Mezi jeho výhody patří snadná čitelnost a zapisovatelnost jak člověkem, tak také strojevě. Tento formát je odlehčenou alternativou k formátu XML, kde nezanedbatelné procento obsahu tvoří pouze samotné značky. Vlastností XML je tedy velká „upovídánost“, což není zrovna ideální pro přenos dat. Gramatická jednoduchost formátu JSON nahrává vysoké interoperabilitě. Tato skutečnost je také jedním z důvodů, proč se stal JSON de facto formátem webu [9, 10].

2.2 Umělá inteligence

Některé živé organismy disponují velice důležitou vlastností, která jim umožňuje efektivně reagovat v nelehkém a složitém prostředí, přičemž jim zároveň i pomáhá k dosažení svých cílů. Tato vlastnost se nazývá inteligence. Mnozí lidé si pak od nepaměti kladli otázku, jestli lze vytvořit umělý systém, který by vykonával takovou činnost, kterou bychom u živých organismů označili za inteligentní.

Významný pokrok v tomto oboru však započal teprve ve 30. letech minulého století, zasloužily se o to významné výsledky v matematické logice a v teorii algoritmů. Po druhé světové válce pak započal prudký rozmach výpočetní techniky, což umožnilo tomuto oboru dále nabírat na významnosti.

Existuje nesčetné množství pokusů o to, jak co nejlépe definovat pojem umělá inteligence. Není se ovšem čemu divit, poněvadž i samotný pojem inteligence, u živých organismů, nemá jednotnou definici a ani jednotné její měření [11].

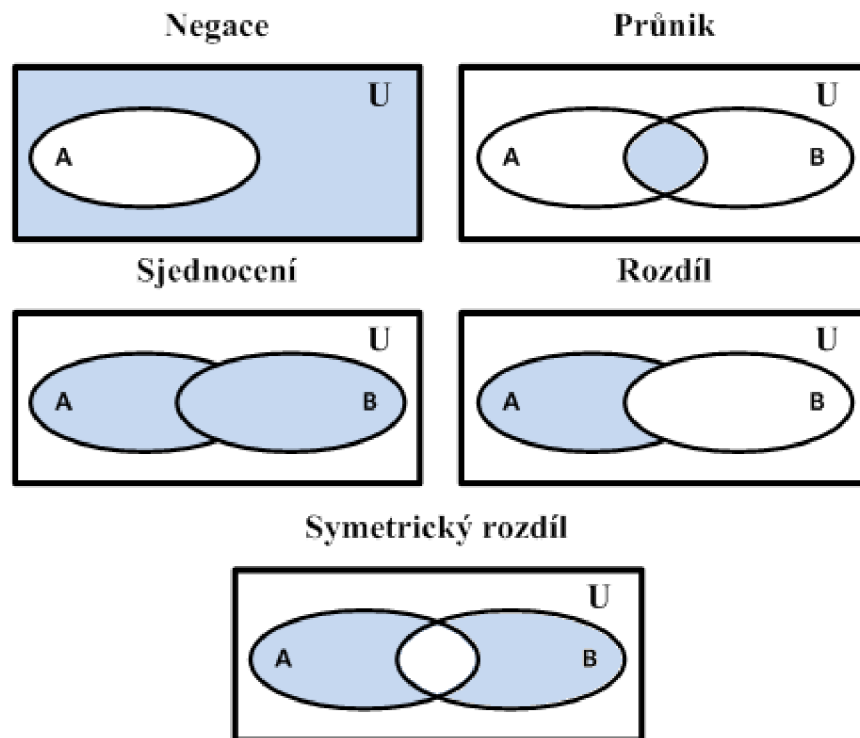
Z celé řady pokusů o definici umělé inteligence zde bude citována jedna, kterou lze zařadit mezi ty nejlépejší a formuloval ji Marvin Minsky: „*Umělá inteligence je věda o vytváření strojů nebo systémů, které budou při řešení určitého úkolu užívat takového postupu, který – kdyby ho dělal člověk – bychom považovali za projev jeho inteligence*“ [11, s. 17].

Mezi nástroje umělé inteligence, kromě neuronových sítí, genetických algoritmů a jiných přístupů k řešení problémů, patří také fuzzy logika [12].

2.2.1 Fuzzy logika

2.2.1.1 Klasické množiny

V teorii množin známe pojmy *množina* a *prvek*. Necht' existuje množina A , pak zápis $x \in A$ značí, že x je prvkem množiny A a naopak zápis $x \notin A$ značí, že prvek x do množiny A nepatří. Množina je charakterizována prvky, které obsahuje. Mezi dvě důležité množiny patří *univerzum* U , kde se jedná o množinu, která obsahuje veškeré prvky a dále pak *prázdná množina* \emptyset , která neobsahuje žádný prvek. Podle teorie množin lze provádět mezi množinami operace, jakými jsou negace, průnik, sjednocení, rozdíl a symetrický rozdíl. Uvedené operace jsou graficky znázorněny na *obrázku 2.3* [12].



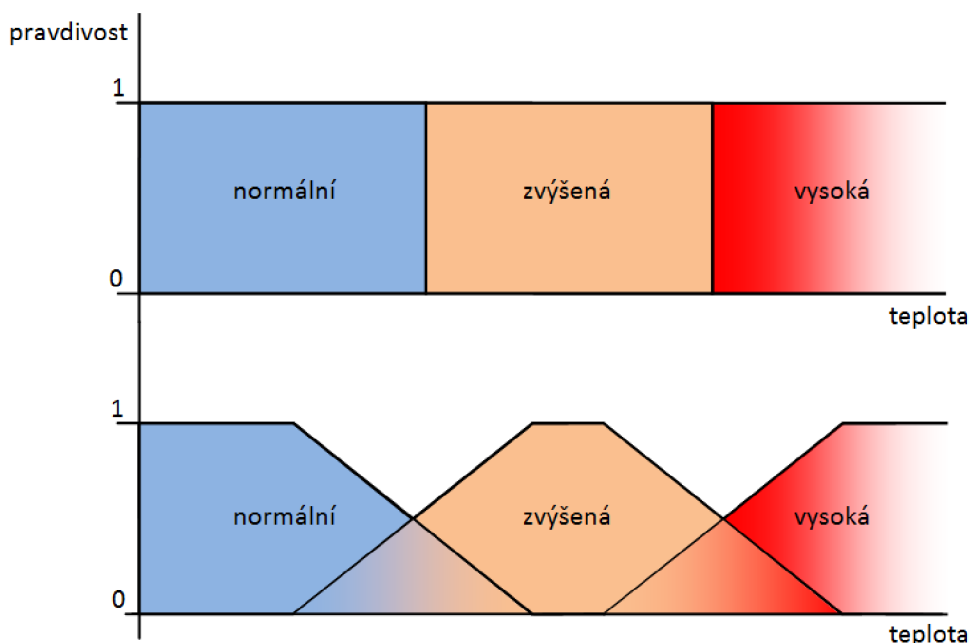
Obrázek 2.3: Grafické znázornění operací mezi množinami

(Zdroj: [12])

Existuje několik způsobů, jak definovat množinu A , jedním z nich je vyjmenování všech prvků. Pro lepší zobecnění klasických množin na fuzzy množiny je však lepší definovat množinu A pomocí charakteristické funkce μ_A . Rozdíl mezi definicí množiny pomocí charakteristické funkce a výrokem je minimální. Charakteristická funkce vrací hodnotu 1 nebo 0 , výrok nám poskytuje hodnotu *pravda* nebo *nepravda* [12].

$$\mu_A(x) = \begin{cases} 1 & \text{pro } x \in A \\ 0 & \text{pro } x \notin A \end{cases}$$

Pokud chceme popsat reálný svět, můžeme s teorií množin narazit na různá úskalí, ne vždy je možné odpovědět na otázku striktně pouze *ano* nebo *ne*, někdy se může odpověď nacházet někde „*mezi*“ nebo můžeme říci „*spíše ano*“ nebo „*spíše ne*“. Další problém se může ukrývat ve vyhodnocování jednoznačných a přesných údajů. Jako příklad lze uvést vyhodnocení teploty vody. Pokud rozdělíme teplotu vody podle intervalu do tří různých pásem a charakterizujeme je výroky, že je *normální*, *zvýšená* a *vysoká*, nastává problém s hranicí intervalů. Bude-li např. hranice mezi pravdivostí výroků *normální* a *zvýšená* 70 °C, pak pro hodnotu 69,999 °C bude pravdivý výrok *normální*, zatímco pro 70,001 °C bude již pravdivý výrok *zvýšená*. Tato situace však neodpovídá skutečnému vnímání světa. V tomto konkrétním případě by bylo lepší, kdyby charakteristická funkce množiny nenabývala hodnot 1 nebo 0, ale kdyby nabývala spojitéch hodnot mezi 0 a 1, což již odpovídá hodnocení výroků ve smyslu fuzzy logiky [13].



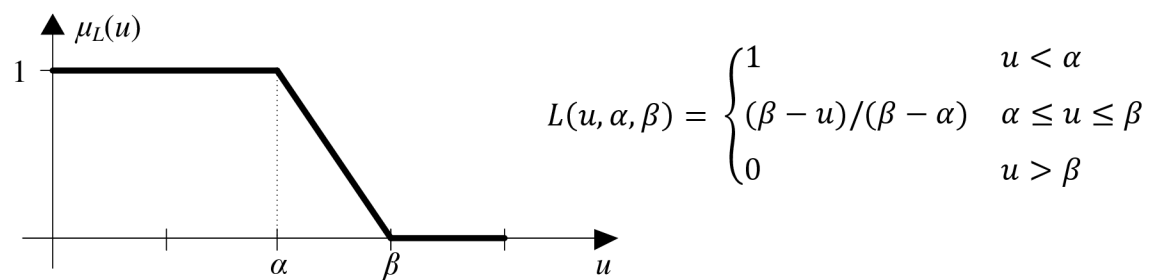
Obrázek 2.4: Demonstrace hranice mezi pravdivostí výroků na příkladu vyhodnocení teploty vody

(Zdroj: Upraveno dle [13])

2.2.1.2 Fuzzy množiny

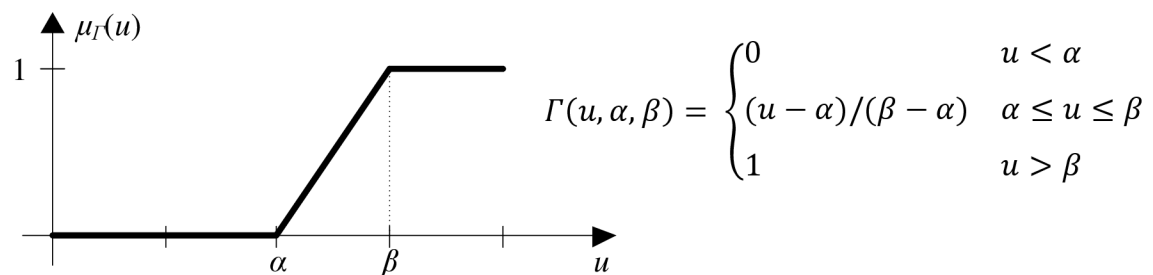
V teorii fuzzy množin se používá pro klasickou množinu výraz *ostrá množina*. Ve *fuzzy množině* však nemusí platit pouze dvě možnosti, tak jak je tomu v ostré množině (prvek do množiny patří nebo nepatří), ale prvek může do *fuzzy množiny* patřit pouze *částečně*.

Ostrá množina C je definována pomocí charakteristické funkce $\mu_C: U \rightarrow \{0, 1\}$, která univerzum mapuje do *dvouprvkové množiny*. Tato funkce je však v teorii fuzzy množin zobecněna a nazývá se *funkce členství*, případně *funkce příslušnosti*. Mapování zde již neprobíhá pouze do dvouprvkové množiny, ale univerzum se mapuje na *celý interval* $[0, 1]$. Funkci příslušnosti μ_F fuzzy množiny F pak zapisujeme jako funkci $\mu_F: U \rightarrow [0, 1]$. Funkce příslušnosti může nabývat různých tvarů, nejčastěji se však používají, z důvodu snadného výpočtu, funkce členství sestavené z lomených přímk, kterými jsou *L-funkce*, *Γ-funkce*, *Λ-funkce* a *Π-funkce* [12].



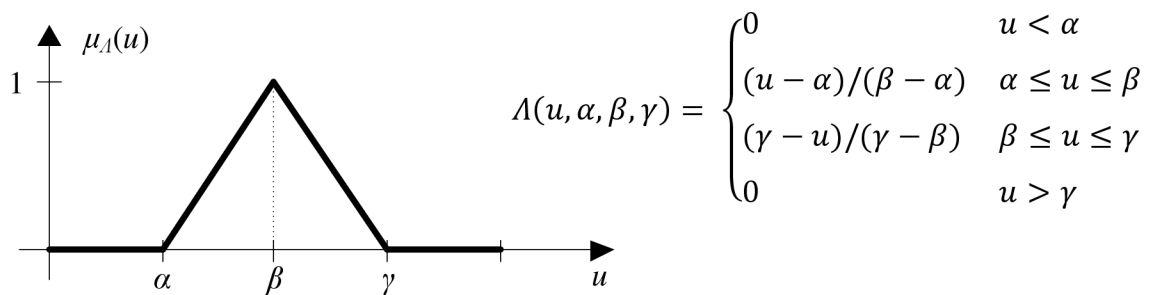
Obrázek 2.5: Průběh a definice *L-funkce*

(Zdroj: [12])



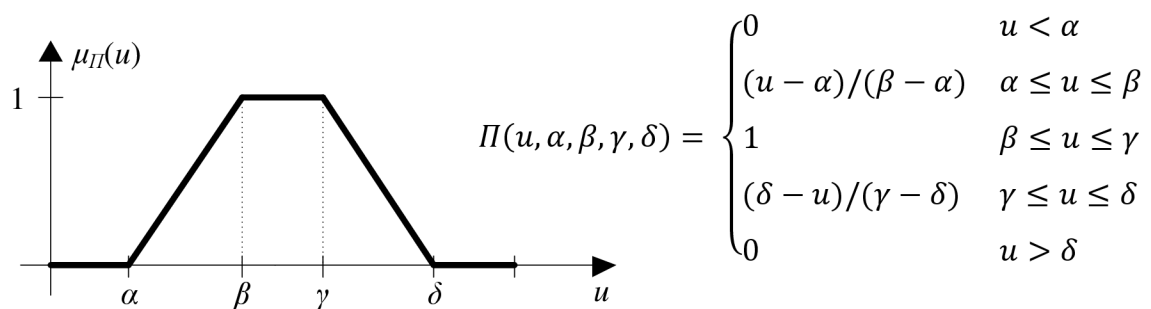
Obrázek 2.6: Průběh a definice *Γ-funkce*

(Zdroj: [12])



Obrázek 2.7: Průběh a definice Λ -funkce

(Zdroj: [12])



Obrázek 2.8: Průběh a definice Π -funkce

(Zdroj: [12])

2.2.1.3 Základní operace

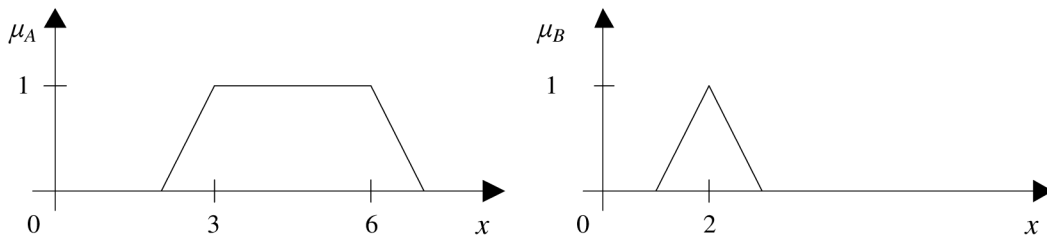
Pro operace sčítání, odčítání, násobení a dělení používá fuzzy logika jiná pravidla, postup u těchto operací je dle [14] následující:

- **sčítání:** $[a, b] + [c, d] = [a + c, b + d]$,
- **odčítání:** $[a, b] - [c, d] = [a - d, b - c]$,
- **násobení:** $[a, b] \times [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$,
- **dělení:** $[a, b] / [c, d] = [\min(a/c, a/d, b/c, b/d), \max(a/c, a/d, b/c, b/d)]$.

U ostrých množin byly uvedeny operace průniku, sjednocení a doplňku. Stejné operace lze provádět i s fuzzy množinami, platí zde však odlišná pravidla a postupy. Vyhodnocování logických operací je dle [12] následující:

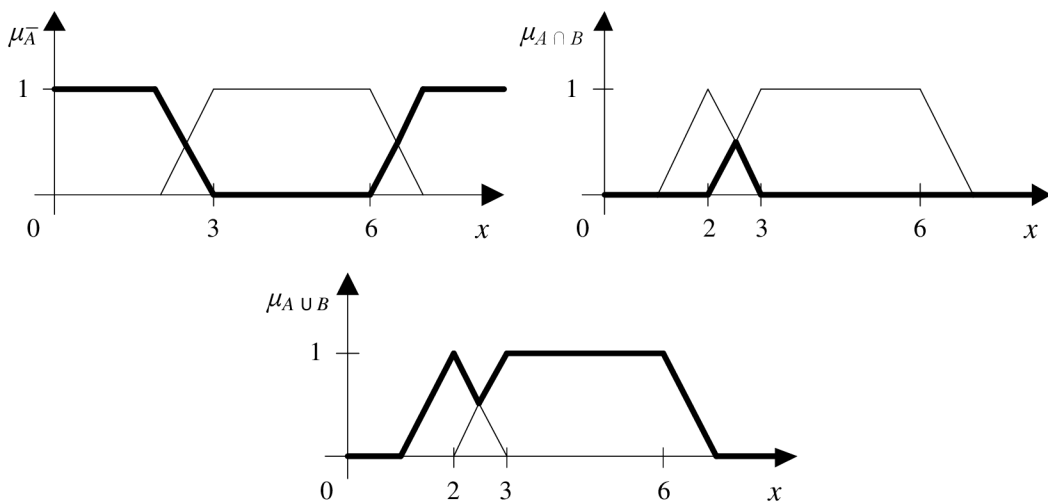
- **průnik:** $\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x))$,
- **sjednocení:** $\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x))$,
- **doplňk:** $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$.

Na **obrázku 2.9** jsou zachyceny funkce členství μ_A a μ_B , grafické znázornění výše popisovaných operací je pak zobrazeno na **obrázku 2.10**.



Obrázek 2.9: Funkce členství μ_A a μ_B

(Zdroj: [14])



Obrázek 2.10: Grafické znázornění operací doplňku (nahore vlevo), průniku (nahore vpravo) a sjednocení (dole)

(Zdroj: [14])

2.2.1.4 Tvorba systému s fuzzy logikou

Na *obrázku 2.11* jsou zachyceny základní kroky postupu při tvorbě systému s fuzzy logikou.



Obrázek 2.11: Postup při tvorbě systému s fuzzy logikou

(Zdroj: [14])

Prvním krokem je fuzzifikace a jedná se o „převedení reálných proměnných na jazykové proměnné. Definování jazykových proměnných vychází z lingvistické proměnné, např. u proměnné riziko lze zvolit následující atributy: žádné, velmi nízké, nízké, střední, vysoké, velmi vysoké riziko“ [14, s. 11].

V druhém kroku se vytvoří báze pravidel, která jasně definuje chování systému. „Pravidla fuzzy logiky představují expertní systém. Každá kombinace atributů proměnných, vstupujících do systému a vyskytujících se v podmínce <Když> <Potom>, představuje jedno pravidlo. Pro každé pravidlo je třeba určit stupeň podpory, tj. váhu pravidla v systému z“ [14, s. 12].

Pravidla si vytváří každý uživatel sám, během optimalizace systému je možné pravidla měnit, poněvadž správný požadovaný výsledek je dán právě dobře definovanými pravidly.

Posledním krokem je defuzzifikace, převádí se zde výsledek předchozí operace, kterým byla jazyková proměnná, na reálné hodnoty. „Cílem defuzzifikace je převedení fuzzy hodnoty výstupní proměnné tak, aby slovně co nejlépe reprezentovala výsledek fuzzy výpočtu“ [14, s. 12].

3 ANALÝZA SOUČASNÉHO STAVU

3.1 Základní informace o firmě

Aplikace bude vyvinuta pro firmu, která se v oboru stavebnictví pohybuje na trhu bezmála již 10 let. Za tuto dobu si firma stačila získat spousty spokojených zákazníků, což mj. také dokládají mnohé pozitivní reference a doporučení. Pro udržení stávající kvality služeb, při růstu počtu zákazníků, se firma rozhodla pořídit si před dvěma lety vlastní prodejnu stavebních materiálů. Expandování firmy však tímto činem neskončilo, posledním aktuálním krokem je vstup na trh realit.

Hlavní podnikatelskou činností firmy je však pro zatím stále stavba a rekonstrukce rodinných domů a bytů. Ve výsledku se tak jedná celkově o jakékoliv stavební práce, včetně venkovních zemních prací, jakými jsou bagrování, pokládka zámkové dlažby apod. Základní činnost tak lze rozdělit do tří ucelených bodů, kterými jsou kamnářství, rekonstrukce a stavební činnost.

Kamnářství:

- Renovace a stavba kachlových kamen
- Prodej a montáž krbových vložek
- Stavba teplovodních, teplovzdušných a akumulčních krbů
- Stavba a renovace komínů

Rekonstrukce:

- Rekonstrukce starých domů a bytů
- Revitalizace zahradních staveb a komunikací (chodníky)

Stavební činnost:

- Stavba rodinných domů a garáží
- Stavba drobných staveb
- Fasádnické a zednické práce (včetně zateplení budov)
- Sádkartónářské práce
- Obkladačské práce

- Tesařské práce
- Vodoinstalace

3.1.1 Vstup na trh realit

V současné době firma rozšiřuje svou činnost. V oboru stavebnictví neustále roste konkurence, z důvodu stability a nezávislosti na poptávkách zákazníků, se firma rozhodla vstoupit na realitní trh a podnikat s nemovitostmi.

Hlavní myšlenka spočívá v koupi nové nemovitosti, její následné rekonstrukci a prodeji. Pro tuto činnost splňuje firma veškeré předpoklady. Na trhu působí již několik let, po tuto dobu získávala praktické zkušenosti při realizaci mnoha projektů. Nabytí „know-how“ tak lehce může zúročit při realizaci rekonstrukce vlastní nemovitosti. Skutečnost, že firma provozuje svou vlastní prodejnu stavebních materiálů, je pak jakýmsi „bonusem“, poněvadž zde dochází k synergickému efektu. Při rekonstrukci nemovitosti je potřeba stavební materiál, který by bylo jinak nutné pořídit u jiného prodejce stavebních materiálů. Jelikož si každý prodejce přidává ke zboží svou vlastní marži, jedná se pro firmu o úsporu v nákladech za materiál, navíc se tím i zvýší do jisté míry obrat vlastní prodejny, což pak v konečném důsledku následně vede k lepší vyjednávací pozici s dodavateli. Při dostatečném odběru určitého zboží od dodavatele je možné vyjednat lepší nákupní cenu a tím je možné ještě více ušetřit, případně dosahovat vyššího zisku při prodeji zákazníkům prodejny.

3.1.2 Rentabilita nového podnikatelského záměru

Podstata nového podnikatelského záměru je již zřejmá, otevřenou otázkou však stále zůstává, zda se tato nová činnost firmě vyplatí. Nikoho zjevně nepřekvapí, že samotné vyjádření firmy na tuto otázku je kladné. Tím je myšleno, že firma předpokládá a má spočítáno, že se jí investice do nemovitostí vyplatí, bude schopná prodat nemovitost po rekonstrukci se ziskem, jinak by se pochopitelně do tohoto investičního záměru nepouštěla. Oprostěme se však nyní od stanoviska firmy a rozeberme celou problematiku rentability investičního záměru podrobněji.

Každý člověk, který vlastní nějakou nemovitost a rozhodne se jí prodat, chce na této transakci co nejvíce vydělat, jedná se o přirozenou lidskou vlastnost. Většina lidí se

tak snaží těsně před prodejem nemovitosti podniknout kroky, které povedou k navýšení její tržní ceny. V tom, jaké kroky by to přesně měly být, však panuje určitá neshoda. Zatímco některé zdroje uvádějí, že i s minimálními úpravami lze zvýšit hodnotu objektu, jiné se k těmto úpravám vyjadřují spíše skepticky. Výčet některých méně nákladných úprav je dle [15] následující:

- **Podlaha** – plovoucí podlaha může vyvolat modernější dojem než koberec.
- **Strop a osvětlení** – starý lustr nahradit podhledem s bodovým osvětlením, navodí opět dojem moderního stylu.
- **Dveře (vnitřní i vchodové)** – bezpečnostní vchodové dveře navodí dobrý dojem již při vstupu do objektu. Vnitřní dveře mohou být poškrábané nebo nemoderní, totéž se týká i klik. Dále se doporučuje obložit kovové zárubně.
- **Koupelna a toaleta** – je zapotřebí, aby na potenciální zájemce o nemovitost udělala tato místa co nejlepší dojem. V konečném důsledku se však nemusí jednat o celkovou rekonstrukci, ale pouze o výměnu dílčích věcí, jakými jsou např. záchodová místa, umyvadlo, baterie apod.
- **Kuchyňská linka** – doporučuje se vyměnit poškrábanou kuchyňskou desku a případně opravit poškozená dvířka skříněk.

Od celkové rekonstrukce např. odrazuje autor tohoto článku [16], upozorňuje zde na skutečnost, že pokud nemáme dobré známé nebo tip na opravdu spolehlivou firmu, mohou být náklady spojené s investicí do nemovitosti vyšší, než bude její navýšená hodnota. Ve výsledku tak lze na rekonstrukci ještě prodělat. Upozorňuje na situaci, že někdy je výhodnější provést korekci svého očekávaného výdělku a prodat objekt ve stávajícím stavu bez rekonstrukce. Podle autora kupující raději na nemovitosti něco ušetří a provede si rekonstrukci sám podle svých představ. Do nemovitosti se před prodejem údajně vyplatí vynaložit pouze nepřilíš vysoké investice, jakými jsou například vymalování a nezbytné opravy.

Jiný zdroj [17] zase uvádí, že pokud je nemovitost určena k rekonstrukci, nelze dopředu přesně odhadnout, kolik výsledná přestavba bude stát, tím tedy ani nelze určit, zda se koupě, za případnou nižší cenu, vyplatí. K standardním kritériím, která se berou v potaz při určování ceny nemovitosti, jakými jsou např. lokalita, občanská vybavenost

apod., patří také stavebně-technický stav. Porovnáme-li ceny dvou identických bytů, které se budou nalézat ve stejné lokalitě, z nichž jeden se bude nacházet ve stavu před rekonstrukcí a druhý po rekonstrukci, rozdíl v ceně bude činit až 20%. Jako příklad výhodné investice u bytového domu je zde uvedeno provedení nové fasády, včetně zateplení a výměny oken, která pro jednu bytovou jednotku vychází průměrně na 120 tis. Kč. Výsledné zhodnocení se u tohoto modelového příkladu uvádí jako dvojnásobek investice.

Je zřejmé, že některé investice do rekonstrukce se vyplatí více, některé méně či vůbec. Pokusem o formulaci toho, které investice se nejvíce zhodnotí, se zabývá článek [18]. Autor zde opět zdůrazňuje, že si je potřeba pohlídat stavební firmu a dohlížet na kvalitu a cenu všech prováděných prací, jinak investice nebude rentabilní. Zároveň zde vysvětluje, že je větší zájem o méně vybavené nemovitosti, poněvadž si většina lidí zařídí vše podle svého vkusu. Dále upozorňuje, že u věcí, které nejsou lidem tolik na očích, jako například vnitřní rozvody, se nevyplatí investovat do nadstandardních materiálů, poněvadž to při koupi málo kdo adekvátně ohodnotí. Naopak, vzhledové parametry objektu, jakými jsou např. fasáda, garáž a střecha, jsou při prodeji nesmírně důležité a kupující je ochoten tyto náležitosti při koupi cenově zohlednit. Souhrn nejvíce a nejméně zhodnocujících investic je uveden v *tabulce 3.1*.

Tabulka 3.1: Souhrn zhodnocujících a nezhdnocujících investic

(Zdroj: [18])

Nejvíce zhodnocující se investice	Nejméně zhodnocující se investice
+ Fasáda	- Nadstandardní vybavení
+ Střecha	- Drahé obklady a dlažby
+ Pěstěná zahrada	- Nábytek na míru
+ Prostorná garáž	- Nadměrné stavební náklady

Veškeré výše uvedené informace byly zaměřené především na běžného člověka, který nedisponuje patřičnými znalostmi, aby byl schopen provést rekonstrukci, před plánovaným prodejem, svépomocí. U mnoha zdrojů lze nalézt zmínku o tom, že pokud se má revitalizace objektu vyplatit, je potřeba u stavební firmy hlídat náklady a kvalitu

prováděné práce. Firma, představená v této práci, se však v oboru stavebnictví pohybuje. Z toho důvodu podobné problémy zcela odpadají, poněvadž by odbytou prací uškodila pouze sama sobě. Zároveň dovede, díky dlouholetým zkušenostem, lépe odhadnout náklady celé přestavby. Dále je potřeba zdůraznit skutečnost, že firma, realizující revitalizaci, bude zároveň vlastníkem objektu, to znamená, že nárůst konečné prodejní ceny nemusí nutně být vyšší, než prováděné stavební úpravy. Jinými slovy, firmě stačí pouze takový zisk, který by získala, kdyby tutéž práci prováděla u některého zákazníka. To znamená, že při tvorbě konečné prodejní ceny disponuje mnohem větším manipulačním prostorem, než běžný člověk. Obrovská výhoda však zde ještě spočívá v tom, že při koupi nemovitosti má „zakázku“ jistou, v ostatních případech musí soupeřit s ostatními konkurenčními firmami na trhu. V rámci konkurenceschopnosti pak firma musí snižovat cenu své nabídky, čímž zároveň snižuje svůj zisk.

Závěrem lze objektivně konstatovat, že nový podnikatelský záměr může být pro firmu rentabilní a rovněž jí může přinést lepší stabilitu v podobě předem jistých zakázek. V textu sice zazněly názory autorů, které poukazyvaly na touhu lidí, provést si rekonstrukci podle svých představ, naproti tomu se však někteří ztotožňují s faktem, že někteří kupující preferují raději již zrekonstruovanou nemovitost, protože se obávají příliš vysokých dodatečných nákladů. V zásadě lze tedy konstatovat, že revitalizovaný objekt si vždy svého kupce najde, poněvadž každý člověk preferuje něco jiného.

3.1.3 Riziko konkurence

Riziko konkurence od firem, které se zabývají jinou činností podnikání, než je stavebnictví, je nízké. Takové firmy by si musely na stavební práce někoho najmout, což je drahé, dostaly by se do podobné pozice jako řadový člověk, který prodává nemovitost. Pro tyto firmy je zde malý potenciální prostor pro zisk a vysoké riziko, že se vynaložená investice nevrátí, tento druh podnikání se tak stává pro tyto subjekty zcela neatraktivním.

Jediné riziko hrozí ze strany stavebních firem. Je potřeba si však uvědomit, že popisovaný druh podnikání sebou přináší poměrně vysoké nároky na finanční prostředky. Pro velké firmy, které si to mohou dovolit, je tento druh podnikání nezajímavý, protože jim z toho nebude plynout dostatečný výnos, navíc by bylo nutné

na tuto práci alokovat lidské zdroje, které jsou potřebné u jiných velkých (a zároveň výnosnějších) projektů.

Střední firmy přinášejí z hlediska konkurence nejvyšší riziko, potřebují však disponovat dostatečným volným kapitálem na koupi a rekonstrukci nemovitosti. Ještě jednou zde zdůrazněme slovo volný kapitál, poněvadž když si takový subjekt vezme úvěr, tak jako firma dostane vyšší úrok, což se již nemusí vyplatit. Buď musí ručit nějakým svým majetkem, nebo nemovitostí, kterou mají zájem koupit a následně se ziskem prodat. Tato situace však již sebou přináší komplikované vyvázání z úvěru (zástavní právo banky).

Malé firmy nedisponují dostatečným finančním kapitálem, museli by si vzít úvěr, víceméně tak pro ně platí podobné podmínky, jaké byly popisovány výše.

Konkurenty se zde však stávají nejen firmy s podobným podnikatelským záměrem, ale také široká veřejnost, jednoduše veškerí lidé, kteří mají zájem o nové bydlení. Dále pak osoby či subjekty, které chtějí investovat peníze za účelem spekulace. V rámci konkurenčního boje zde existuje možnost nabídnutí vyšší částky, rivalita je zde však poměrně nízká, poněvadž přeplácení je do maximální výše výdělku. Jedná se většinou o poctivé soupeření.

3.2 Situace na trhu nemovitostí a hypoték

V roce 2016 se obchodu s nemovitostmi mimořádně dařilo. Na realitním trhu poptávka po nemovitostech převyšovala nabídku. Za hlavní příčinu lze označit úrokové sazby hypoték, které již po dlouhou dobu vykazovaly klesající trend a v listopadu 2016 **průměrná úroková sazba** klesla dokonce, s hodnotou **1,77%**, na své historické minimum. Vývoj průměrných úrokových sazeb, za sledované období 2013 – 2016, je zaznamenán v **grafu 3.1** [19, 20].

Jen pro zajímavost, listopad 2016 byl zároveň i nejúspěšnějším měsícem, co se týče počtu sjednaných úvěrů a i jejich celkové výše. V tomto měsíci bylo sjednáno 14 386 hypotečních úvěrů, s celkovým objemem 29,683 miliard korun. V meziměsíčním srovnání, s měsícem říjnem téhož roku, tak byl zaznamenán nárůst o 4 902 hypotečních úvěrů a celkový objem hypoték narostl o 10,898 miliard korun [20].

FINCENTRUM HYPOINDEX LISTOPAD 2016



Graf 3.1: Vývoj průměrných úrokových sazeb

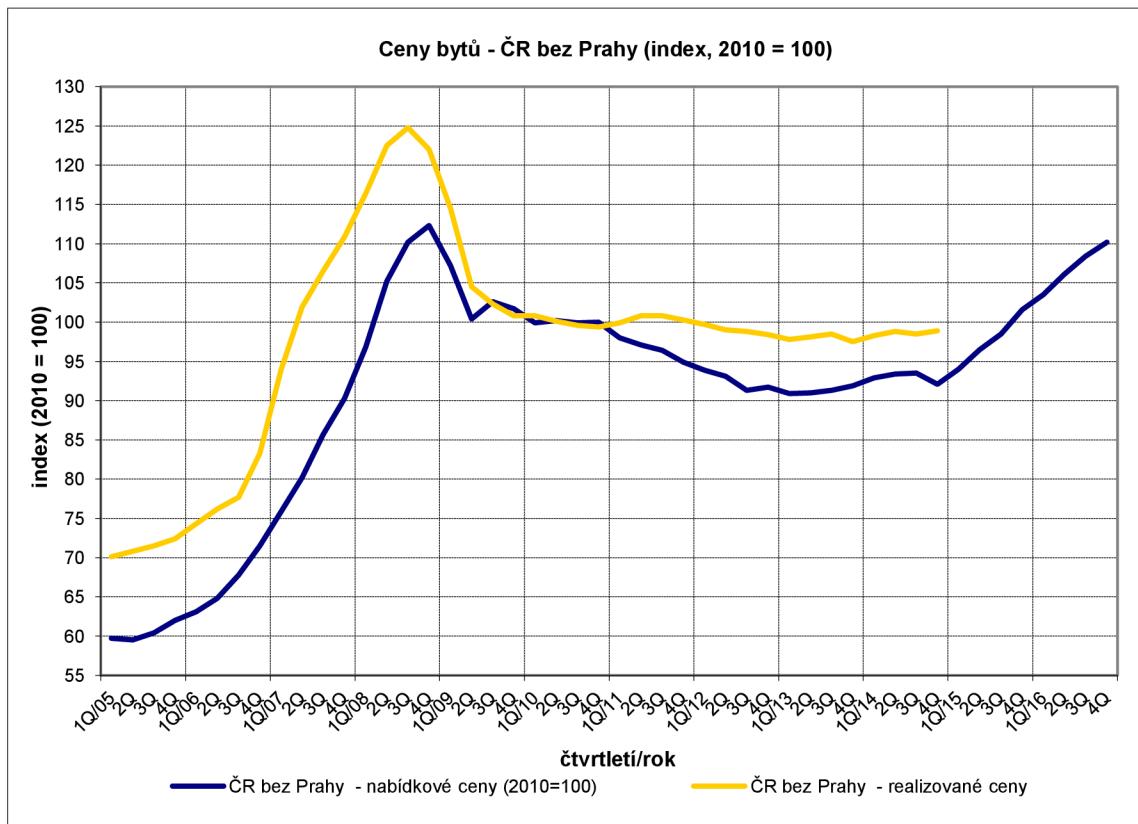
(Zdroj: [20])

Nízké úrokové sazby hypoték bezesporu stojí za zvýšenou poptávkou na realitním trhu, na tuto skutečnost však také reagují ceny nabízených nemovitostí. Nedostatečná nabídka má za následek růst cen bytů i rodinných domů. Lidé jsou při nízké úrokové sazbě více ochotni přejít z nájemního bydlení do vlastního, poněvadž se výše měsíčního nájmu může blížit splátce hypotečního úvěru [19].

Vývoj cen bytů v ČR, bez hlavního města Prahy, je znázorněn v *grafu 3.2*, i když je rostoucí trend cen bytů bez výjimky platný pro celou ČR, situace v Praze je přece jen mírně atypická. Z důvodů možného zkreslení byl tedy zvolen graf, kde nejsou uvažovány hodnoty hlavního města. Ceny nemovitostí jsou zde podstatně vyšší, než je celorepublikový průměr. Podíváme-li se i na tzv. index návratnosti bydlení, zjistíme, že mezi jednotlivými kraji panují docela velké rozdíly [21].

„Jednoznačně nejhůře z těchto statistik vychází hlavní město, kde je sice průměrný příjem jednoznačně nejvyšší v ČR, ale ceny bytů jsou zde neporovnatelně vyšší, než je celorepublikový průměr. Index návratnosti bydlení v metropoli dosahuje hodnoty

10,55násobku ročního příjmu. Nad celorepublikovým průměrem je ještě Jihomoravský kraj v čele s Brnem a indexem na hodnotě 6,54násobku ročního příjmu“ [21].



Graf 3.2: Vývoj cen bytů v ČR (bez hlavního města Prahy)

(Zdroj: [22])

3.2.1 Výhledy do budoucna

Většina odhadů se ztotožňuje s tímtež názorem, ceny nemovitostí budou v roce 2017 i nadále růst. Vysvětlení spočívá v tom, že ceny nezávisí pouze na úrokových sazbách hypotečních úvěrů, ale odvíjejí se od celkového hospodářského vývoje [23].

Ministerstvo financí České republiky provedlo v listopadu 2016 průzkum makroekonomických prognóz, jehož výsledky vycházejí z predikce 19 institucí: „Cílem průzkumu makroekonomických prognóz (tzv. Kolokvia), který provádí MF ČR, je zjistit názor relevantních institucí na budoucí vývoj české ekonomiky a vyhodnotit základní tendence, s nimiž zúčastněné instituce ve svých předpovědích počítají“ [24]. Na základě výsledků, nacházejících se ve zveřejněném dokumentu [25], lze usuzovat, že hospodářský vývoj ČR v roce 2017 bude podobný, jako tomu bylo v roce 2016. Míra

nezaměstnanosti by měla i nadále klesat, je potřeba však upozornit, že její hodnota se již pohybuje na hranici přirozené míry nezaměstnanosti, proto je další pokles již dosti omezený [25].

Nízká míra nezaměstnanosti sebou přináší u lidí větší ochotu utrácet a zadlužovat se. Strach ze ztráty zaměstnání je potlačen, poněvadž lidé věří, že si rychle a snadno najdou zaměstnání nové. Hospodářský růst tak bude stále lidi utvrzovat v tom, že je vhodný čas vzít si hypoteční úvěr a pořídit si vlastní bydlení. Poptávka po nemovitostech bude tedy stále vysoká a to se musí zákonitě odrazit také v jejich ceně [23].

Další prognóza se týká výše úrokových sazeb hypotečních úvěrů. Již v říjnu 2016 vydala Česká národní banka doporučení, které výrazně omezuje půjčky na 100% hodnoty nemovitosti, maximální limit byl stanoven na 95%. I když se jedná pouze o doporučení, nikoliv nařízení (ČNB nemá pro zatím pravomoc určovat parametry hypotečních úvěrů), ČNB může bankám, které doporučení nebudou respektovat, zvýšit povinné rezervy. V dubnu roku 2017 by měl stanovený limit klesnout dokonce na hodnotu 90%. Podíl doporučených hypotečních úvěrů v pásmu 80 – 90% bude činit 15% [23, 26, 27].

V lednu 2017 však vláda ČR schválila návrh novely zákona, podle kterého by ČNB získala zákonnou pravomoc určovat parametry hypotečních úvěrů. V případě schválení novely zákona Poslaneckou sněmovnou, Senátem a podpisu prezidenta republiky, by zákon měl začít platit od července 2017 [27].

Úrokové sazby hypotečních úvěrů tak budou, podle předpovědi, v roce 2017 postupně růst. Velkou měrou se na tom podepíše také zákon 257/2016 Sb., o spotřebitelském úvěru, který nabyl účinnosti již v prosinci 2016. Tento zákon mj. ukládá bankám povinnost přísnější kontroly finanční situace klienta. Klient zase získává zákonem stanovenou 14 denní lhůtu na rozmyšlenou, během které nemůže banka nabídku změnit ani odvolat. Další změny jsou v podobě předčasného splacení hypotéky atd. [28].

3.3 Analýza sledovaných atributů nemovitostí

Vzhledem k účelu nemovitosti, který byl již upřesněn výše, byly vybrány následující atributy, jejichž hodnoty budou mít při výběru nejvyšší váhu a na jejich základě bude

provedeno rozhodnutí o případné koupi. Důležité je však poukázat na podstatné rozdíly mezi požadavky a váhou jednotlivých atributů v závislosti na typu objektu. Zatímco u rodinného domu je prostor pro rekonstrukci takřka neomezený, u bytu má člověk do jisté míry „svázané ruce“, alespoň co se vnějších úprav týče. Výčet následujících atributů tak může nabývat různého významu pro odlišné typy objektů.

Stav objektu

Je požadováno, aby se výsledný objekt nacházel v takovém stavu, aby jej bylo možné revitalizovat. Tímto tedy z výběru automaticky vypadávají veškeré novostavby (s výjimkou nedokončených) a objekty nacházející se po rekonstrukci. V zásadě tedy platí, čím horší stav objektu, tím lépe (samozřejmě do určité míry). Ideálními kandidáty se tak stávají objekty před rekonstrukcí.

Energetická náročnost budovy

U tohoto atributu v zásadě platí podobné požadavky, jak již bylo naznačeno v předchozím případě. Vyšší energetická náročnost budovy dává prostor pro akční zásah ze strany investora, aplikací tepelné izolace se hodnota nemovitosti jistě zvýší. Samo o sobě však pouhé zateplení objektu a následný prodej nemusí být dostatečně rentabilní, proto je požadována i vhodná kombinace s jinými faktory. Tento atribut bude hrát vyšší roli u rodinných domů, poněvadž u bytů je realizace tepelné izolace objektu, z hlediska jednotlivce, do jisté míry omezená.

Vzdálenost objektu od sídla firmy

K celkovým nákladům na revitalizaci objektu je rovněž nutné také přičíst náklady na přepravu. Vysoká hodnota nákladů snižuje výsledný zisk pro investora, z hypoteticky dostupné množiny objektů je tak potřeba odfiltrovat ty, jejichž vzdálenost od sídla firmy je příliš velká. Volba maximální vzdálenosti však není úplně tak jednoduchá, logicky je výhodnější co možná nejnižší. Co když se však potenciálně výhodný objekt bude nacházet těsně za stanovenou hranicí? Vzdálenost je tedy volena tak, aby byl pokryt dostatečný prostor a zvýšila se tak i pravděpodobnost na vyskytnutí se výhodné koupě. S rostoucí vzdáleností je však požadována výhodnější kombinace ostatních sledovaných atributů.

Cena objektu

Požadavek je zřejmý, pořídit vhodnou nemovitost za nejnižší možnou cenu, bude však panovat diametrálně odlišná cena např. u velkého rodinného domu s pozemkem a jednopokojovým bytem. Akceptovatelná hodnota tohoto atributu je tak silně závislá na kombinaci ostatních atributů.

Plocha objektu

Užitná a zastavěná plocha jsou atributy, které výrazně ovlivňují cenu nemovitosti. Je pochopitelné, že větší plocha objektu bude mít za následek ochotu vynaložit více finančních prostředků, to stejné platí i opačně, při menší ploše objektu bude muset být jeho cena nižší.

Číslo podlaží v domě

Byt nacházející se ve vyšším podlaží může přinášet jisté komplikace, přesun potřebného materiálu pro rekonstrukci bude zajisté se stoupajícím pořadovým číslem patra náročnější. I když by bylo možné namítnout, že s využitím výtahu je v podstatě jedno, v kterém patře se byt nachází, je potřeba konstatovat, že ne vždy je možné výtah k těmto účelům využívat a nelze také opomenout to, že někde nemusí být výtah vůbec. Váha tohoto atributu však bude nižší, rozhodně nebude mít na rozhodování takový vliv, jakým je např. podobný atribut - vzdálenost objektu od sídla firmy.

Vybavení objektu a příslušenství

I když investor, vzhledem k záměru, jak chce objekt využít, nemá přímé požadavky na vybavení objektu, je potřeba tyto údaje do rozhodovacího procesu zahrnout a počítat s nimi. Vybavení objektu a jeho příslušenství má podstatný vliv na celkovou cenu. Téměř identické byty, nacházející se v podobné lokalitě, budou mít určitě rozdílnou cenu, pokud jeden bude s balkónem a druhý bez balkónu. Podobných příkladů by šlo zajisté najít více, zde je výčet některých údajů, které budou mít vliv na cenu nemovitosti a tím i působit na rozhodování:

- Balkón
- Lodžie
- Garáž

- Sklep
- Výtah

Inženýrské sítě

Přítomnost inženýrských sítí je velice důležitá věc, která má zásadní vliv na cenu nemovitosti. Dodatečné zavedení je dosti nákladné, nicméně absence např. vodovodu, pokud je přítomný vlastní pitný zdroj vody, lze tolerovat. Totéž platí pro napojení na kanalizaci, tento problém lze řešit pomocí septiku nebo domovní čističky odpadních vod. Výčet sledovaných atributů je následující:

- Elektrická přípojka
- Vodovod
- Vlastní zdroj pitné vody
- Kanalizace

Umístění objektu

Opět se jedná o atribut, který je pro investora relativně nevýznamný, ovlivňuje však zásadně pořizovací cenu nemovitosti. Objekt nacházející se v centru města bude zajisté dražší, než typově stejný objekt, nacházející se někde na okraji města.

Vlastnictví

Jedná se o atribut, který se týká především bytů. Investorem je preferováno osobní vlastnictví nemovitosti.

3.4 Shrnutí a zhodnocení

Popisovaná firma se pohybuje v oboru stavebnictví, hlavní podnikatelskou činností je rekonstrukce a stavba nemovitostí. Spolu s touto činností provozuje také svou vlastní prodejnu stavebních materiálů. Obě dvě tyto aktivity spolu vytváří synergický efekt, s jehož pomocí dokáže lépe čelit konkurenčnímu prostředí a dosahovat vyšších zisků.

Nejnovějším podnikatelským záměrem firmy je investování do nemovitostí. Podstata spočívá v koupi vhodné nemovitosti, její následné rekonstrukci a prodeji.

V současné době panuje na trhu situace, že poptávka po nemovitostech převyšuje nabídku. Tato skutečnost má za následek růst cen nemovitostí.

V roce 2016 byly zaznamenány historicky nejnižší úrokové sazby u hypotečních úvěrů. ČNB však začala vydávat pro banky jistá doporučení, které mají za následek mírné zvyšování úrokových sazeb. Návrh novely zákona, který by měl v případě schválení nabýt platnosti v létě roku 2017, zvyšuje pravomoc ČNB, umožní ji určovat parametry hypotečních úvěrů.

Podnikatelský záměr firmy byl podroben objektivnímu posouzení. Výsledkem bylo konstatování, že firma je schopná využít své dlouholeté zkušenosti a investice do vhodné nemovitosti jí může přinést patřičný zisk. Důvodem je zejména skutečnost, že firmě stačí k dosažení rentability prodat nemovitost s navýšením vůči kupní ceně o tolik, kolik by inkasovala za podobnou práci u některého svého zákazníka.

Firma tedy disponuje veškerými potřebnými prostředky pro úspěšné podnikání s nemovitostmi, nechybí jí důležité „know-how“ a ani potřebný volný kapitál. Jediným pomyslným úskalím se tak stává výběr vhodné nemovitosti. Je zřejmé, že samotné posouzení, zda je ta či ona nabídka výhodná, nebude tím pravým problémem. Ten se ukrývá především ve vyhledání výhodné nabídky z celkového dostupného nabízeného množství nemovitostí. Projít veškeré dostupné nabídky všech realitních kanceláří zabere bezesporu určitý čas. I když nové inzeráty nemusí přibývat nikterak rapidním tempem, neustálá kontrola nově přidaných nabídek si opět vyžádá svůj čas.

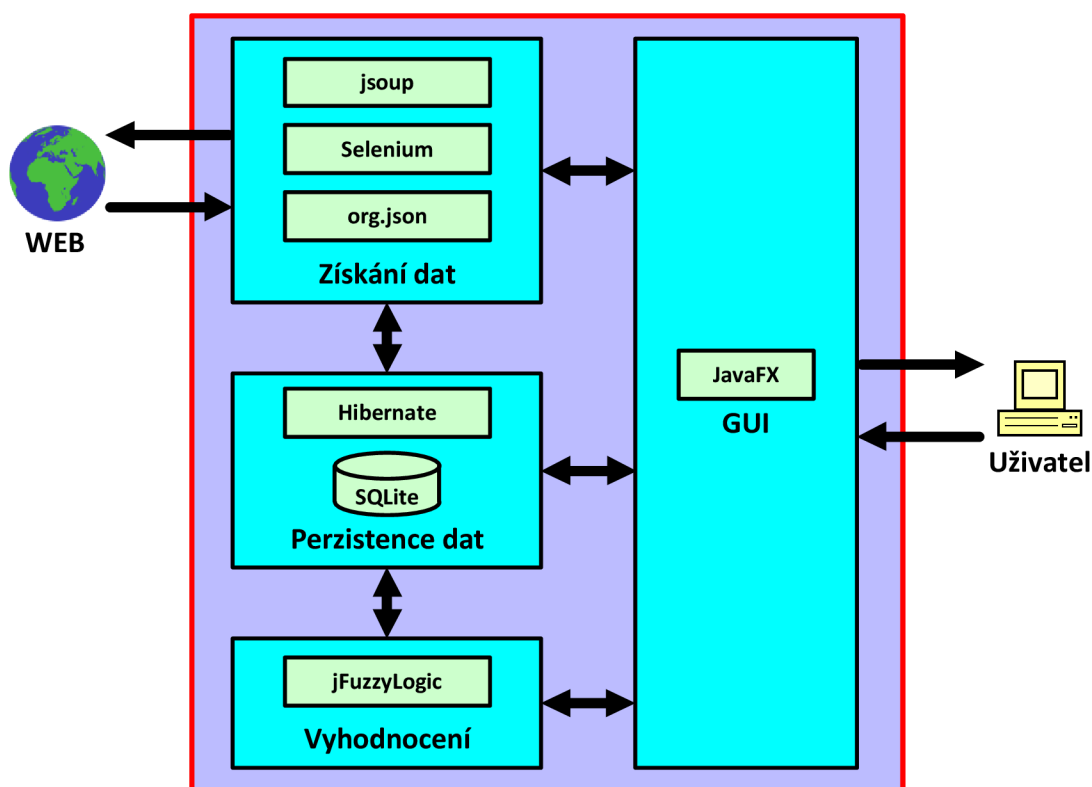
Ideální situace by tak nastala, kdyby byl k dispozici podpůrný nástroj, který by neustále sledoval nově vystavené nabídky a zároveň je dokázal ohodnotit podobným způsobem, jako by to udělal člověk.

4 VLASTNÍ NÁVRHY ŘEŠENÍ

4.1 Návrh řešení aplikace

Tato kapitola se bude věnovat výslednému návrhu a implementaci celé aplikace. Problém lze dekomponovat do čtyř ucelených částí, tento fakt se bude odrážet i v rozložení jednotlivých kapitol.

Za první část lze označit vývoj modulu, který bude mít na starost komunikaci s webem, který bude obsahovat inzeráty týkající se nemovitostí (realitní servery nebo přímo webové stránky konkrétních realitních kanceláří). Odtud se budou získávat potřebná data, která budou předána dalšímu modulu, který bude mít již na starost samotné vyhodnocování. Záznamy budou ukládány do databáze, o tuto skutečnost se postará modul pro perzistenci dat. Aby bylo možné prezentovat výsledky uživateli, poslední dekomponovanou částí bude vytvoření přívětivého grafického uživatelského rozhraní. Návrh celé aplikace je, v podobě diagramu, znázorněn na *obrázku 4.1*.



Obrázek 4.1: Schéma návrhu aplikace

(Zdroj: Vlastní zpracování)

4.2 Zdroj dat

Aby bylo možné jednotlivé nemovitosti hodnotit a porovnávat, je potřeba o nich získat patřičné informace. Výčet konkrétních atributů byl podrobně rozebrán v kapitole o analýze sledovaných atributů, předmětem této kapitoly tak bude obecné získávání dat.

Většina realitních kanceláří vystavuje inzerce nemovitostí také na svém webu, toto medium se tedy jeví jako vhodnou volbou pro čerpání dat. Pomocí vhodného parseru je možné data získat z dokumentů HTML. Jedním z těchto HTML parserů je také knihovna *Jsoup*.

Problém je skryt v odlišném zpracování jednotlivých webů realitních kanceláří, každý web má svou unikátní stromovou strukturu a liší se také obsahem jednotlivých HTML dokumentů. Potřebná data se tak mohou skrývat pod jiným označením např. *Technologie výstavby – cihlová* vs. *Druh objektu – cihlová* nebo *Počet pokojů* vs. *Velikost*, porovnání výčtu některých atributů z dvou webů je znázorněno jako ukázka na *obrázku 4.2*. Dále je samozřejmě potřeba počítat s odlišným HTML kódem, parsování, i stejných atributů, tak bude docíleno s použitím různého řetězce HTML elementů.

Základní vlastnosti		INFORMACE K NABÍDCE	
Technologie výstavby:	cihlová	Číslo zakázky	1776
Stav objektu:	novostavba	Datum aktualizace	16.12.2016
Poloha objektu:	samostatný	Poloha objektu	řadový
Umístění objektu:	kraj obce	Druh objektu	cihlová
Celkový počet podlaží:	2	Umístění objektu	centrum obce
Počet pokojů:	5 a více pokojů	Velikost	5 a více pokojů

Obrázek 4.2: Porovnání některých atributů z dvou různých webů

(Zdroj: Upraveno dle [29, 30])

Z uvedených informací jasně vyplývá, že pro každý podporovaný web by bylo nutné rovněž navrhnout unikátní modul, který by zajistil správné získávání dat. Vzhledem k počtu realitních kanceláří, které na trhu figurují, by kromě samotného vytvoření, byla také pracná a neefektivní správa velkého množství těchto modulů. Jakákoliv změna na podporovaném webu by znamenala také úpravu a aktualizaci příslušného modulu,

nehledě také na fakt, že některé realitní kanceláře na svých webech používají v inzerátech pouze souvislý text, jinými slovy, pro důležité informace není použita žádná strukturovaná tabulka, získání potřebných informací by tak bylo příliš složité.

Ideálním řešením by bylo, kdyby se informace nacházely na jednom místě, kde je již zajištěna jejich integrace. Taková řešení lze nalézt v podobě *realitních serverů*, jakými jsou např. *Sreality.cz*, *Reality.iDNES.cz* nebo *HyperReality.cz*. Pro inzerci je používají i samotné realitní kanceláře, tyto realitní servery zpravidla obsahují i určitý druh importního rozhraní, který je patřičně zdokumentován. S použitím můstku tak lze vložit novou inzerci na realitní server z pohodlí vlastního realitního softwaru.

Většina realitních kanceláří inzeruje současně na více realitních serverech, při návštěvě těchto serverů tak lze narazit na stejné opakující se nabídky. Po konzultaci s firmou, pro kterou bude aplikace určena, byl jako zdroj dat vybrán realitní server *Sreality.cz*, který je v současné době nejnavštěvovanějším realitním serverem a navíc její firma rovněž prozatím využívá pro hledání vhodné nemovitosti pro nový podnikatelský záměr. Do budoucna je však počítáno s tím, že aplikace bude podporovat i další servery. I když se hodně nabídek opakuje, stále je možné narazit na unikátní nabídky a tím i zvýšit pravděpodobnost nalezení nemovitosti, která splňuje požadované parametry.

4.3 Získání dat

Způsob, jakým budou získávána data, se značně odvíjí od použitých technologií na daném realitním severu. Pro získání dat byl vybrán realitní server *Sreality.cz* [31], který je *Single Page Aplikací (SPA)*, která využívá *JavaScriptový framework* od *Google* – *AngularJS*. Tato technologie funguje ve zkratce tak, že se pomocí JavaScriptu mění základní kostra HTML dokumentu. Jak již název napovídá, vše se děje na jedné stránce, obsah je průběžně měněn na základě interakce uživatele. JavaScript běží na klientské straně, server pouze poskytuje na vyžádání potřebná data, většinou ve formátu *JSON*.

Použití knihovny *Jsoup* zde tedy nepřichází v úvahu, poněvadž pro zobrazení obsahu je potřeba spustit JavaScript, který vytváří konečný HTML dokument, což tato knihovna (přímo) neumožňuje. Jedinou věcí, kterou by tak bylo možné parsovat, by byla základní holá kostra HTML dokumentu, která neobsahuje žádný užitečný obsah, což pochopitelně nemá žádný význam.

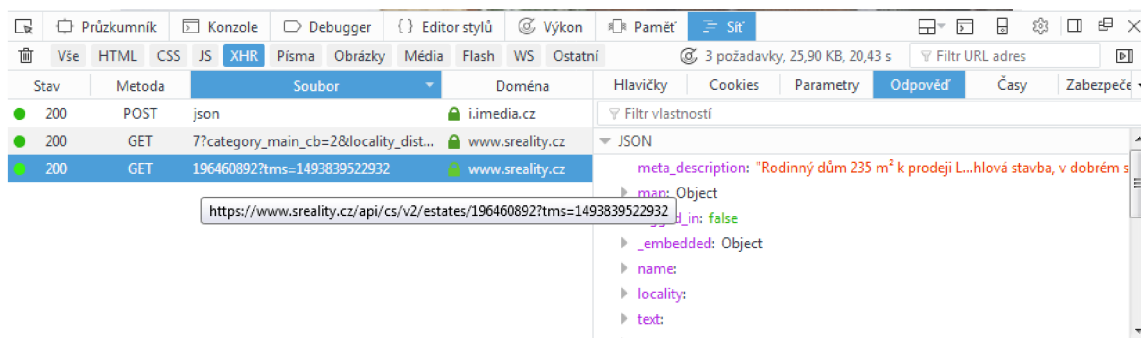
Pro korektní získání dat je potřeba mít k dispozici HTML kód již kompletně vyrenderované stránky, tuto činnost běžně provádí každý webový prohlížeč. V prostředí jazyka Java je tedy nutné mít k dispozici komponentu, která umožní simulovat uživatelskou interakci s webovými stránkami a bude umět spustit JavaScript, tento problém lze vyřešit pomocí frameworku *Selenium* [32], který je určen pro testování webových aplikací. Pomocí tohoto nástroje lze v jazyku Java napsat testy, které simulují veškerou uživatelskou činnost. Na realitním serveru Sreality.cz by tedy bylo možné vyplnit a odeslat formulář s požadavky pro vyhledávání, následně obdržet výsledek v podobě seznamu nemovitostí a tento seznam postupně procházet. Z webové stránky s popisem již konkrétní nemovitosti by pak už bylo možné snadno parsovat potřebná data, protože aby byla automatizace uživatelské interakce vůbec možná, musí tento framework poskytovat k tomu také adekvátní nástroje. Pomocí *XPath* lze z HTML stromu vybrat potřebné uzly, s kterými je možné dále pracovat.

Důležitým prvkem frameworku *Selenium* je pak samotný *WebDriver*, pro testování je možné použít drivery prohlížečů třetích stran, které nejsou vyvinuty přímo vývojáři toho frameworku. Jedná se např. o *Mozilla GeckoDriver*, *Google Chrome Driver*, *Microsoft Edge Driver* atd. Veškeré tyto zmíněné WebDrivery však při spuštění testu otevrou okno daného webového prohlížeče, což je sice pro testování pochopitelně velice důležitá věc, pro vývoj aplikace, o které pojednává tato diplomová práce, by však tato skutečnost byla rušivým elementem. Naštěstí však existují i WebDrivery, které běží na pozadí a žádné okno neotvírají, jedná se o tzv. *headless webové prohlížeče* (bez grafického uživatelského rozhraní), jedním z nich je např. *GhostDriver (PhantomJS)* [33].

PhantomJS je rovněž i využíván serverem Sreality.cz, tento prohlížeč je spuštěn na serveru, kde překládá stránky s JavaScriptem do HTML, poněvadž vyhledávače a roboti, kteří neumějí spustit JavaScript, by viděli pouze prázdnou bílou webovou stránku. Při přidání proměnné *_escaped_fragment_* do query stringu je pak na serveru předán URL odkaz požadavku na PhantomJS (URL je předáno bez této proměnné), tam je proveden překlad na klasický HTML dokument (stáhnou se veškeré potřebné soubory a spustí se JavaScript), který je vrácen zpět tomu, kdo zaslal požadavek, tedy např. zmíněnému robotovi. Vykreslení webové stránky tímto způsobem zabere poměrně dost času, v průměru je to cca 1 – 5 vteřin [34].

S použitím frameworku Selenium a vhodného WebDriverů by tedy bylo možné potřebná data úspěšně získat. Je potřeba si však uvědomit, že se jedná relativně o dosti výpočetně náročný způsob, i když je možné u WebDriverů nastavit různé *capabilities* (schopnosti), jakými jsou např. zákaz načítání obrázků, podpora SSL protokolů atd., pořád je nutné spustit JavaScript a webovou stránku vyrenderovat, abychom získali potřebný HTML kód.

Nabízí se ještě jedno mnohem elegantnější a výpočetně méně náročné řešení, které bude rovněž použito v této práci. Jak již bylo v úvodu této kapitoly nastíněno, realitní server Sreality.cz je Single Page Aplikací, framework AngularJS se tak vlastně stará o „prezentaci dat“, pro vyvíjenou aplikaci jsou však zcela postačují pouze „surová data“, tedy data, která server poskytuje na vyžádání tomuto frameworku ve formátu JSON. Podobně jako lze provádět parsování HTML dokumentů, obdobně lze provádět tuto činnost i u souborů JSON. Realitní server Sreality.cz sice nemá veřejně zdokumentované API (alespoň jsem žádnou dokumentaci nenašel, existuje však dokumentace k importu inzerátů), nicméně požadavky s konkrétními parametry, které jsou zasílány na konkrétní URL adresy, lze zachytit např. pomocí *nástrojů pro vývojáře* ve webovém prohlížeči *Mozilla Firefox*. Stejně tak lze zjistit, co bylo na daný požadavek vráceno, viz *obrázek 4.3*. Podrobnějším rozбором tohoto řešení se bude zabývat následující podkapitola.



Obrázek 4.3: Zachycení požadavku a odpovědi pomocí nástrojů vývojáře

(Zdroj: Vlastní zpracování)

4.3.1 Získání potřebných dat z realitního serveru Sreality.cz

Ze všeho nejdříve je potřeba udělat to stejné, co udělá běžný uživatel při příchodu na tento server, vyplnit nabízený formulář, na jehož základě bude provedena filtrace nemovitostí dle zadaných parametrů.

Je zřejmé, že bude nejspíše zbytečné brát v úvahu veškeré nemovitosti, které jsou v ČR nabízeny, proto bude do filtru vstupovat parametr, který omezí výběr pouze na Jihomoravský kraj. I tato oblast je však vzhledem k velikosti působnosti firmy dosti velká, proto bude výběr dále omezen pouze na okres Břeclav. Po poradě s firmou byl do filtru zahrnut také fakt, že nemá zájem o dražby.

Po vyplnění a odeslání formuláře lze pomocí nástrojů pro vývojáře ve webovém prohlížeči Mozilla Firefox zachytit URL adresu požadavku, která vypadá následovně: *https://www.sreality.cz/api/cs/v2/estates?category_main_cb=2&category_type_cb=1&locality_district_id=74&locality_region_id=14&no_auction=1&page=1*.

Jednoduchou analýzou pak lze zjistit, co dané parametry a jejich hodnoty znamenají. Možných parametrů a jejich hodnot je samozřejmě podstatně více, dále budou pouze popsány ty, které jsou důležité pro tuto práci:

- **category_main_cb** – hodnota 1 udává, že se bude jednat o byty, hodnota 2 znamená, že se bude jednat o domy, hodnota 3 pozemky atd.,
- **category_type_cb** – hodnota 1 říká, že se bude jednat o prodej, hodnota 2 znamená pronájem,
- **locality_district_id** – tento parametr se týká zvoleného okresu, hodnota 74 znamená okres Břeclav, hodnota 75 okres Hodonín atd., přičemž je samozřejmě možné současně zvolit více hodnot, např. 75|74|77,
- **locality_region_id** – tento parametr značí kraj, přičemž hodnota 14 vyjadřuje Jihomoravský kraj,
- **no_auction** – hodnota 1 udává, že do výběru nebudou zahrnuty aukce, hodnota 0 znamená, že obdržíme přehled nemovitostí včetně aukcí,
- **page** – výčet všech nemovitostí se většinou nevejde na jednu stránku, počet stran se odvíjí od celkového počtu záznamů a od množství záznamu na jedné stránce. Hodnota tohoto parametru představuje konkrétní požadovanou stranu.

Pokud nebude v požadavku uveden parametr *per_page*, který udává počet záznamů nemovitostí v souboru JSON, bude brána v úvahu defaultní hodnota 20 záznamů. Řazení je defaultně nastaveno od nejnovějších inzerátů, proto není potřeba do požadavku zahrnout také parametr *sort*. Odpověď na výše uvedenou URL adresu požadavku je znázorněna na **obrázku 4.4**. Pochopitelně se nejedná o kompletní odpověď, cílem je pouze poukázat na ty části, které jsou důležité pro tuto práci.

```

▶ meta_description: "223 realit v nabídce prodej domů Břeclav. Vyberte si novou nemovitost na sreality.cz s hledáním na mapě a velkými náhledy fotografií nabízených domů."
result_size: 223
▼ _embedded:
  ▼ estates:
    ▼ 0:
      ▶ labelsReleased:
        has_panorama: 0
      ▶ labels:
        is_auction: false
      ▶ labelsAll:
      ▶ seo:
        category: 2
        has_floor_plan: 0
      ▶ _embedded:
        paid_logo: 0
        locality: "Podivín, okres Břeclav"
        has_video: false
      ▼ _links:
        ▶ dynamicDown:
        ▶ dynamicUp:
        ▶ iterator:
        ▼ self:
            href: "/cs/v2/estates/2009915740"
        ▶ images:
        ▶ image_middle2:
        new: false
        auctionPrice: 0
        type: 1
        hash_id: 2009915740
        attractive_offer: 0
        price: 1
      ▶ price_czk:
      rus: false
      name: "Prodej rodinného domu 114 m², pozemek 643 m²"
      region_tip: 0
      ▶ gps:
    ▼ 1:

```

Obrázek 4.4: Odpověď na požadavek ve formátu JSON – Seznam inzerátů

(Zdroj: Upraveno dle [31])

Prvním důležitým atributem je *result_size*, jehož hodnota vyjadřuje celkový počet záznamů, které vyhovují zvoleným kritériím. V závislosti na počtu záznamů, které jsou vráceny v jedné odpovědi (odpovídá to počtu zobrazených inzerátů na webové stránce) pak známe počet dotazů, které bude potřeba uskutečnit (počet stran, které je potřeba projít). Základní informace o jednotlivých nemovitostech, které jsou jinak zobrazeny v přehledu nemovitostí na webové stránce, jsou v souboru JSON reprezentovány jako pole *estates* vnořeného objektu *_embedded*. Každý záznam tohoto pole pak obsahuje vnořený objekt *_links*, půjdeme-li v hierarchii dále, tak narazíme na další objekt *self*, kde nás zajímá hodnota atributu *href*, která představuje relativní adresu požadavku pro získání podrobných informací o dané nemovitosti.

Parsováním URL adres konkrétních nemovitostí získáme seznam, s kterým lze dále pracovat. Podrobnější popis aktualizací a přidávání nových záznamů do databáze na základě tohoto seznamu bude rozebrán v této práci později. Nyní se budeme věnovat získávání dat z JSON souboru, který obdržíme na základě URL adresy z toho seznamu, viz *obrázek 4.5*. Opět se nejedná o kompletní odpověď, příklad slouží pouze pro ilustraci toho, jak budou potřebná data získávána.

Z vnořeného objektu *map* lze získat GPS souřadnice dané nemovitosti, ty dále poslouží k výpočtu vzdálenosti objektu od sídla firmy. Dále je důležité pole objektů *items*, které obsahuje zbytek důležitých dat, které budou potřebné pro hodnocení nemovitosti. Jak lze vidět, u příslušných objektů v tomto poli jsou důležité hodnoty atributů *name* a *value*. Je však potřeba počítat s faktem, že ne vždy jsou uvedena veškerá potřebná data, jinými slovy, některé atributy, jako např. umístění, energetická náročnost nebo vybavení, mohou u různých nemovitostí chybět.


```

▶ meta_description: "Rodinný dům 210 m² k prodeji Lanžhotská, Kostice; 2 090 000 Kč (včetně DPH, včetně poplatků, + provize, včetně právního servisu, cena bez daně z nabytí, cena k jednání), garáž, patrový, Řadový, cihlová stavba, v dobrém stavu."
▼ map:
▶ geometry: [1]
  lon: 16.9780256167044
  zoom: 15
▶ bounding_box:
  lat: 48.7412405846313
  type: "geometry"
logged_in: false
▶ _embedded:
▶ name:
▶ locality:
▶ text:
▶ price_czk:
  is_topped: true
▶ _links:
  panorama: 1
▶ seo:
  rus: false
  is_topped_today: true
▼ items:
▼ 0:
  negotiation: true
  name: "Celková cena"
  ▶ notes:
    value: "2 090 000"
    currency: "Kč"
    type: "price_czk"
    unit: "za nemovitost"
▶ 1:
▶ 2:
▼ 3:
  type: "edited"
  name: "Aktualizace"
  value: "Dnes"
  topped: true
▶ 4:
▼ 5:
  type: "string"
  name: "Stav objektu"
  value: "Dobrý"
▶ 6:

```

Obrázek 4.5: Odpověď na požadavek ve formátu JSON – Informace o konkrétní nemovitosti

(Zdroj: Upraveno dle [31])

4.3.2 Získání dat o vzdálenosti pomocí Google Maps API

K získání dat o vzdálenosti bude využito Google Maps API [35], které je již, na rozdíl od API Sreality.cz, zdokumentované. Pro výpočet vzdálenosti budou zapotřebí dva údaje, jednak GPS souřadnice sídla firmy, které poslouží jako výchozí bod, dále pak GPS souřadnice konkrétního objektu, jejichž postup získání byl popsán v předchozí kapitole.

Veškeré požadavky budou zasílány s patřičnými parametry na následující URL:
<https://maps.googleapis.com/maps/api/distancematrix/outputFormat?parameters>.

Popis a význam některých vybraných parametrů je pak následující [35]:

- **outputFormat** – jedná se o formát, v jakém bude odpověď zaslána. Může nabývat hodnot *json* nebo *xml*, což odpovídá daným formátům,
- **units** – tento parametr udává, v jakých jednotkách se bude vzdálenost počítat. Hodnota *metric* říká, že se bude počítat v kilometrech a metrech, hodnota *imperial* značí, že vzdálenost bude vrácena v mílích a stopách,
- **origins** – jedná se o výchozí bod, od něhož se bude počítat vzdálenost a doba trvání cesty. Hodnotou mohou být jak konkrétní GPS souřadnice, tak také adresa nebo ID místa,
- **destinations** – jedná se o cílový bod, platí zde stejná pravidla jako pro předchozí parametr *origins*,
- **mode** – jedná se o specifikaci druhu cestování, defaultní hodnota je nastavena na *driving*, pro výpočet vzdálenosti je tedy použita síť silnic.

Příklad URL adresy konkrétního požadavku pak může vypadat následovně:
<https://maps.googleapis.com/maps/api/distancematrix/json?units=metric&origins=48.7508841,16.9048569&destinations=48.8328924625326,16.9340527851665>.

Získaná odpověď je opět ve formátu JSON, viz **obrázek 4.6**. K potřebné informaci o vzdálenosti se lze dostat přes prvek pole *rows*, který obsahuje další pole *elements*. Hodnota atributu *text*, objektu *distance*, pak již je požadovaná vzdálenost.

```

▼ destination_addresses:
  0: "Prušánská 20, 691 01 Moravský Žižkov, Česko"
▼ origin_addresses:
  0: "Lanžhotská 10-16, 690 02 Břeclav, Česko"
▼ rows:
  ▼ 0:
    ▼ elements:
      ▼ 0:
        ▼ distance:
          text: "13,3 km"
          value: 13292
        ▼ duration:
          text: "17 min"
          value: 1021
          status: "OK"
        status: "OK"

```

Obrázek 4.6: Odpověď na požadavek ve formátu JSON – Vzdálenost objektu od sídla firmy

(Zdroj: Upraveno dle [29])

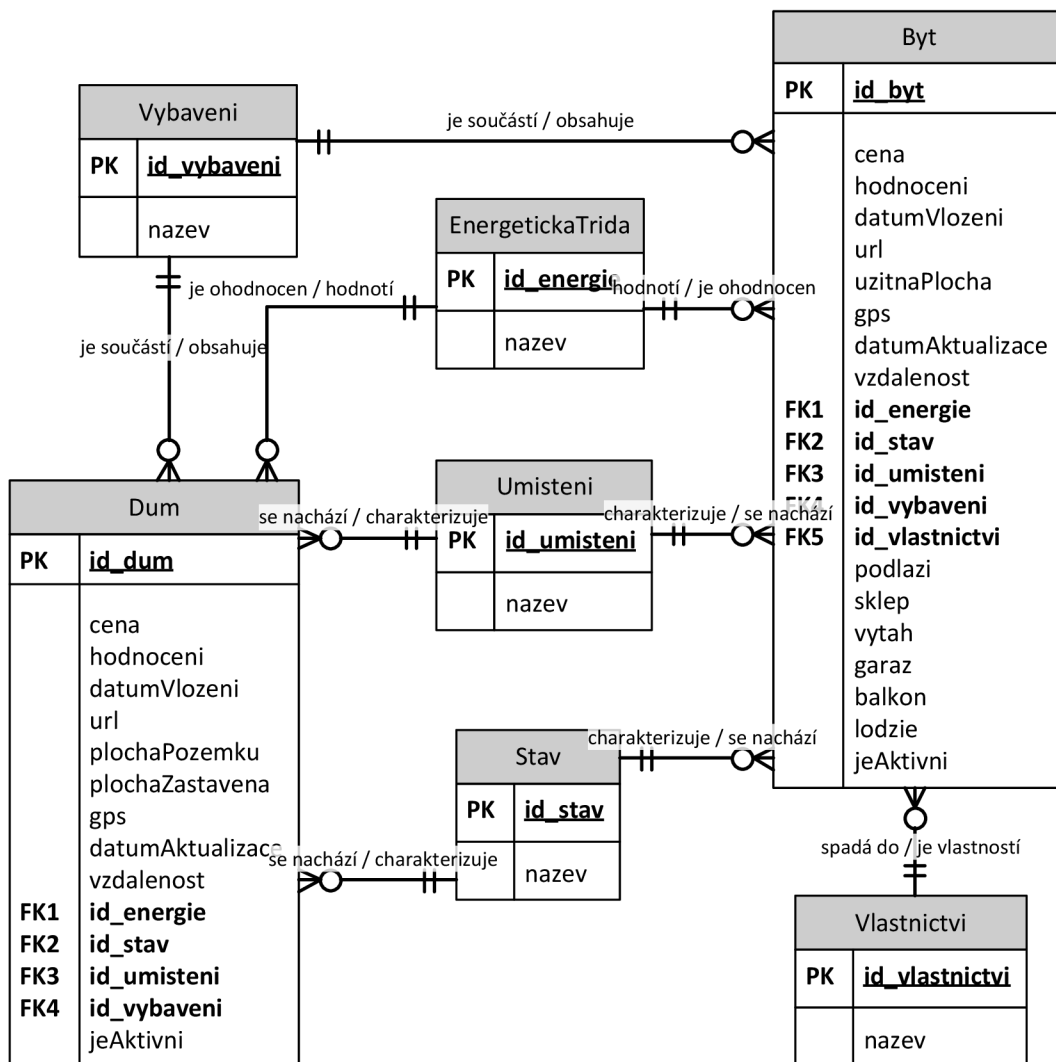
4.4 Ukládání dat

V předchozí kapitole byl prezentován postup, jak získat potřebná data, je nasnadě otázka, co s těmito „surovými“ daty udělat dál. V této fázi je potřeba se zamyslet nad tím, zda budou data potřebná k pozdějšímu využití. Jinými slovy, zda budou data pouze předána fuzzy modelu k vyhodnocení a výsledek prezentován uživateli (samozřejmě s jistým uchováním výsledků a průběžným přidáváním nových záznamů) nebo získaná data budou ukládána do perzistentního úložiště k ještě např. pozdějším analýzám.

Vzhledem k tomu, že je zamýšleno aplikaci později rozšířit o další analýzy, budou získaná data ukládána do databáze. Dostatečné množství dat posléze umožní získat cenné informace, které poslouží ke zlepšení fuzzy modelu a tedy k lepším výsledkům. Aby však bylo něco takového uskutečnitelné, bude potřeba data nejprve po určitou dobu sbírat, teprve pak bude možné dodatečnou funkcionalitu využívat. S takto nasbíraných dat je např. zamýšleno odhadnout lokality, které budou více rentabilní (na základě cen nemovitostí) apod. Touto dodatečnou funkcionalitou se však tato práce zabývat nebude, při návrhu aplikace je však potřeba s ní počítat.

4.4.1 Entito-relační diagram

Na základě analýzy potřebných atributů, která byla provedena v kapitole o analýze současného stavu, byl vytvořen Entito-relační diagram, který je zachycen na **obrázku 4.7**. Tento diagram poslouží jako podklad při vytváření schématu databáze pro vyvíjenou aplikaci. Nejdůležitějšími entitami jsou *Dům* a *Byt*, ostatní entity představují číselníky. Jak je patrné, tyto dvě entity obsahují také atributy, které nebyly součástí zmiňované analýzy a nejsou tak pro hodnocení bezprostředně potřebné, při vytváření aplikace však hrají důležitou roli, proto budou v rámci popisu entit podrobně popsány.



Obrázek 4.7: Entito-relační diagram

(Zdroj: Vlastní zpracování)

Entita Dům

Tato entita představuje nemovitost typu dům, atributy *id_energie*, *id_stav*, *id_umisteni* a *id_vybaveni* jsou číselníky, které představují vlastnosti nemovitosti, které jsou bezprostředně uvedeny v inzerátu a patří mezi kritéria pro hodnocení. Dalším bezprostředně zjistitelným kritériem je ještě atribut *cena*, který vyjadřuje pořizovací cenu nemovitosti (cena se může po aktualizaci změnit, nemovitost může být zlevněna) a nakonec atributy *plochaPozemku* a *zastavenaPlocha*. Mezi hodnotící kritéria dále patří atribut *vzdalenost*, který představuje vzdálenost nemovitosti od sídla firmy. Tento atribut je však potřeba vypočítat, GPS souřadnice sídla firmy jsou známy, GPS souřadnice objektu jsou uvedeny v inzerátu a je potřeba je uchovávat pro pozdější zamýšlené analýzy, proto se v této entitě vyskytuje také atribut *gps*.

Atributy *datumVlozeni* a *datumAktualizace* mají jednak informativní charakter, ale poslouží rovněž k pozdějším analýzám. Atribut *datumVlozeni* představuje datum, kdy byly údaje o nemovitosti vloženy do databáze. Atribut *datumAktualizace* pak vyjadřuje datum, kdy byly údaje inzerátu jakkoliv pozměněny. Tento údaj je bezprostředně uveden v inzerátu, je potřeba však zmínit skutečnost, že toto datum bývá často aktualizováno, ale k žádným úpravám údajů přitom nedochází. K této činnosti většinou dochází pouze za účelem posunu daného inzerátu na přední příčky.

Jak již bylo nastíněno, záznamy o nemovitostech budou archivovány k pozdějším analýzám, z tohoto důvodu obsahuje tato entita atribut *jeAktivni*, který umožní rozlišit, zda je nemovitost ještě v inzerci nabízena, či nikoliv.

Posledním nepopsaným atributem je atribut *hodnoceni*, který reprezentuje výstup fuzzy modelu, který daný objekt ohodnotí. Zahnutí tohoto atributu do entity *Dům* je však diskutabilní, poněvadž tuto informaci není nutné uchovávat, ale je možné ji v případě potřeby vždy znovu vypočítat. Budeme-li však vycházet z předpokladu, že fuzzy model již nebude po jeho odladění měněn¹, pak je zbytečné tuto činnost neustále při každém spuštění aplikace vykonávat. I když výpočet není nikterak rapidně výkonnostně náročný, na druhou stranu potřebná velikost úložiště také není nikterak velká, proto byla dána přednost druhé zmíněné variantě.

¹ Pro zakomponování výsledků pokročilejších analýz sice bude muset být fuzzy model změněn, tato skutečnost však sebou přinese také jiné mírné změny v celé aplikaci

Entita Byt

Tato entita obsahuje některé atributy, které se vyskytují rovněž v entitě *Dům* a byly tak již popsány, proto zde bude věnována pozornost pouze odlišným a nepopsaným atributům.

U entity *Byt* je plocha jinak vnímána, bere se zde v potaz pouze užitná plocha nemovitosti, která je reprezentována atributem *uzitnaPlocha*. Dále zde přibyl číselník, který objasňuje, do jakého vlastnictví daná nemovitost spadá, jedná se o atribut *id_vlastnictvi*. U této entity se lze dále setkat s atributem *podlazi*, který vyjadřuje, v jakém patře se byt nachází. Posledními nepopsanými atributy jsou *sklep*, *vytah*, *garaz*, *balkon* a *lodzie*. Tyto atributy patří do pomyslné skupiny booleovských výrazů, které nám poskytují odpověď na otázku, zda *je*, či *není* daná vlastnost, kterou atribut představuje, u této entity přítomna.

Entity reprezentující číselníky

Entita *Vybaveni* představuje číselník, který nám říká, zda je součástí nabízené nemovitosti také vybavení. Tento číselník bylo nutné použít z toho důvodu, že tato skutečnost může nabývat celkem tří stavů (včetně vybavení, částečně vybaveno nebo bez vybavení).

Dalším číselníkem je entita *EnergetickaTrida*, který zachycuje energetickou náročnost budovy. Dle klasifikačních tříd může atribut *nazev* nabývat hodnot A až G.

Pro specifikaci toho, kde se vybraná nemovitost nachází, pak slouží entita *Umisteni*. Dle analýzy se nemovitosti mohou vyskytovat v centru obce, v klidné části obce, na okraji obce, v polosamotě, v rušné části obce nebo v úplné samotě.

Při hodnocení objektu hraje významnou roli také jeho stav, tento parametr je reprezentován entitou *Stav*. Z výsledků analýzy vyplynulo, že objekt se může nacházet ve stavu velmi dobrém, dobrém, před rekonstrukcí, špatném, po rekonstrukci, ve výstavbě nebo se může jednat o projekt, či novostavbu.

Poslední nepopsanou entitou je *Vlastnictvi*, přičemž vlastnictví nemovitosti může nabývat hodnot osobní, družstevní nebo státní/obecní.

4.4.2 Řešení perzistence dat v aplikaci

Pro perzistenci dat bude použit framework *Hibernate*, který umožní objektově-relační mapování. Tento Framework se vyznačuje rozsáhlou podporou SQL dialektů, otázkou tedy zůstává, jaký databázový systém použít.

Zásadní podmínkou jsou nízké pořizovací náklady, ideální by bylo, kdyby byl databázový systém dostupný zdarma. Zaměříme-li se tedy pouze na zdarma dostupná řešení, nabízí se pak např. databázové systémy *MySQL*, *PostgreSQL* nebo *SQLite*. Vzhledem k charakteru aplikace a také přihlédnutím k tomu, jaké nároky na tento systém budou kladeny a jaké operace bude potřeba vlastně provádět, byl pro perzistenci dat vybrán databázový systém *SQLite* [36]. Je však nutné poznamenat, že dialekt k tomuto databázovému systému není podporován přímo vývojáři frameworku *Hibernate*. Nicméně, potřebný dialekt si může vývojář napsat sám, případně použít nějaký již vytvořený a zveřejněný komunitou, jeden z takových např. je [37].

Schéma databáze bude vytvořeno na základě zde prezentovaného ER diagramu. V jazyce Java bude potřeba vytvořit k uváděným entitám odpovídající POJO objekty. Je zde potřeba ještě upozornit na fakt, že objektový model mohl pochopitelně vypadat i jinak, poněvadž mohla být využita dědičnost. V modelu mohla být zastoupena třída *Nemovitost*, od které by teprve dědily třídy *Dům* a *Byt*. Pro někoho by toto řešení možná bylo i „čistší“, nicméně v této práci jsem se rozhodl držet těch entit, které jsou prezentovány v ER diagramu, protože přinášejí snadnější ORM.

4.4.3 Aktualizace dat

Aktualizace dat bude postavena na získaném seznamu URL adres jednotlivých inzerátů. Kontrola aktuálnosti se bude skládat z dvou typů kontrol.

První druh kontroly bude mít za cíl odhalit, zda se nevyskytují nové inzeráty, které v databázi ještě nejsou uloženy. Zároveň bude provedena kontrola „živosti“ již uložených záznamů. Algoritmus pro tuto kontrolu je jednoduchý, je postaven na porovnání dvou seznamů URL adres. Jeden seznam bude získán v rámci požadavku na realitní server *Sreality.cz*, druhý seznam bude získán na základě dotazu na interní databázi vyvíjené aplikace. Na základě tohoto dotazu budou vybrány jen URL adresy těch záznamů, které jsou aktivní (odpovídá atributu *jeAktivni* ER diagramu). Oba seznamy se porovnají, ty položky, které jsou obsaženy v prvním seznamu, ale v druhém

nikoliv, jsou nové inzeráty, jejichž data je potřeba do databáze přidat. Opačně, ty položky, které se vyskytují v druhém seznamu, ale v prvním nikoliv, je potřeba udělat neaktivními (inzerát byl smazán).

Druhý druh kontroly bude spočívat v revizi uložených informací o dané nemovitosti. Inzeráty jsou na realitním serveru Sreality.cz pravidelně aktualizovány, nutno však podotknout, že většinou se tak děje pouze z důvodu posunu nabídky na přední příčky, v samotném obsahu inzerátu se tedy většinou toho moc nezmění (až na datum aktualizace). Někdy však dojde k tomu, že je nemovitost např. zlevněna nebo může být nějaký jiný údaj změněn nebo přidán. Aby byla podchycena i tato situace, musí se projít veškeré inzeráty, které jsou v seznamu URL adres. Algoritmus bude tedy podobný, jak tomu bylo v předchozím případě, opět dostaneme dva seznamy, tentokrát nás však zajímají veškeré URL adresy v prvním seznamu (získán z Sreality.cz), které se rovněž vyskytují v seznamu druhém. Následuje revize, zda obsah inzerátu koresponduje s uloženými daty.

Je zřejmé, že druhý typ kontroly je výpočetně i časově náročnější. Kdy a jaká kontrola bude provedena, záleží na uživateli, obecně lze doporučit provádět první typ kontroly každý den. Druhý typ kontroly není již tolik důležitý, může však někdy změnit výsledné hodnocení dané nemovitosti. Obecně lze doporučit tuto kontrolu provádět jednou týdně.

4.5 Návrh a vytvoření fuzzy modelů

Pro hodnocení nemovitostí bude použita fuzzy logika. Poněvadž samotná nemovitost je příliš obecný pojem, bude zapotřebí dvou fuzzy modelů, jeden bude určený pro domy, druhý pro byty. Jedině tak lze dosáhnout maximálního přizpůsobení modelu konkrétním požadavkům.

Pro návrh a odladění modelu bude použit program *MATLAB*, konkrétně sada nástrojů (knihovna funkcí) obsažená ve *Fuzzy Logic Toolbox*. Jedná se o profesionální nástroj, který rovněž disponuje grafickým uživatelským rozhraním. Aplikace je ovšem vyvíjena v jazyce Java, tento nesoulad však lze vyřešit, poněvadž *MATLAB* obsahuje i rozhraní pro různé programovací jazyky. Pomocí *MATLAB Builder JA* lze z funkcí *MATLABu* vytvářet Java komponenty.

4.5.1 Fuzzy model pro hodnocení domů

Při tvorbě fuzzy modelu pro nemovitost typu dům je potřeba počítat celkem s osmi vstupními proměnnými, kterým byl dále přidělen určitý počet fuzzy množin (uvedeno v závorce): zastavěná plocha (5), plocha pozemku (5), umístění (6), vybavení (3), stav (5), energetická náročnost budovy (7), cena (5), vzdálenost (4).

Kdyby se celý fuzzy model skládal pouze z jednoho bloku, do kterého by směřovaly veškeré vstupy, obsahoval by tento model příliš velké množství pravidel. Počet pravidel získáme jako kombinaci všech možných vstupů, pro ilustraci lze potřebný počet pravidel demonstrovat na následujícím výpočtu:

$$POČET PRAVIDEL = 5 \times 5 \times 6 \times 3 \times 5 \times 7 \times 5 \times 4 = \underline{315000}$$

Takto velký počet pravidel velice komplikuje odladění celého modelu, i když MATLAB umožňuje sledovat závislost jednotlivých proměnných, která se odvíjí právě od použitých pravidel, tento přístup by vedl k příliš složitému řešení. Aby bylo možné model co nejlépe odladit, nabízí se možnost celý problém dekomponovat.

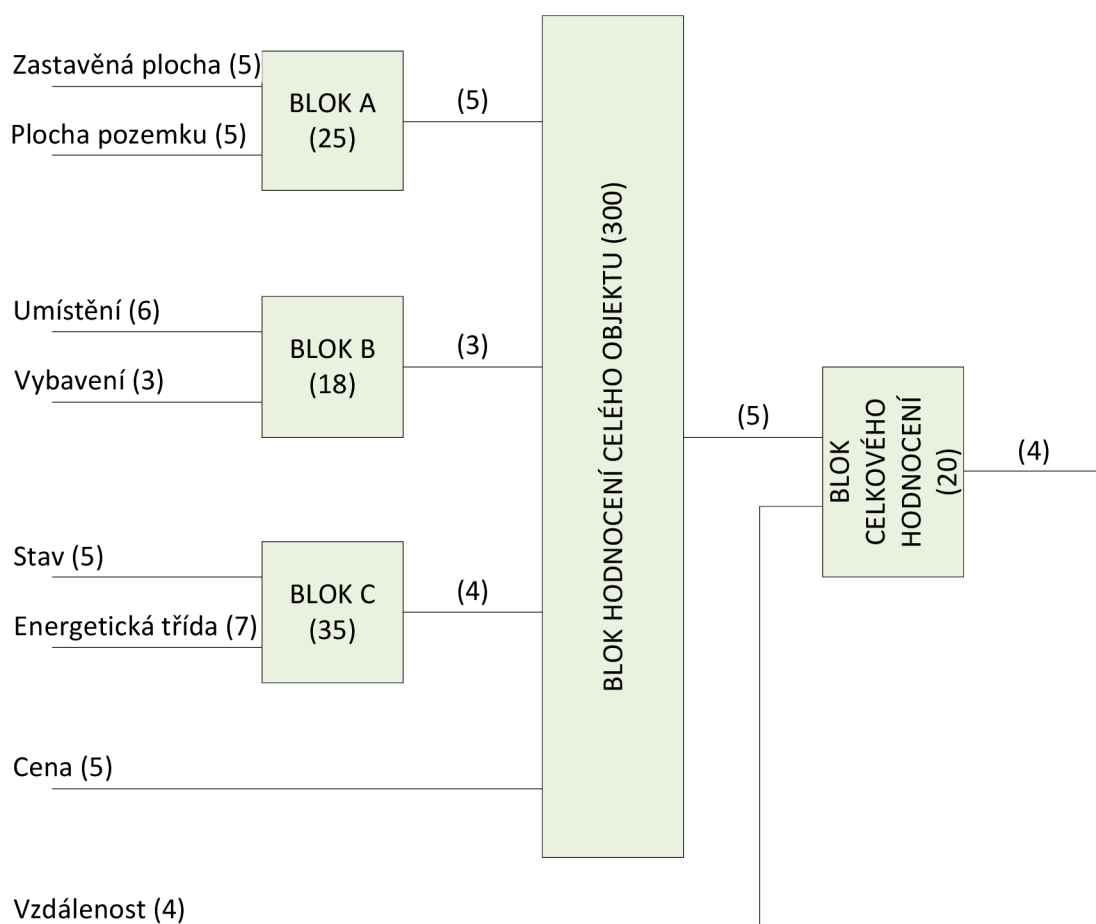
Dekompozicí problému získáme několik samostatných bloků, které budou řešit dílčí podproblémy. Do bloku tak budou vstupovat např. jen 2 vstupy, namísto 8. Tento blok bude mít jeden výstup, který bude zase vstupem pro další blok. Konkrétní dekomponovaný model výše popisovaného problému je znázorněn na **obrázku 4.8**.

Líčené schéma obsahuje všech 8 vstupů, místo jednoho vyhodnocovacího bloku však zde je použito bloků 5. Čísla v závorkách na vstupech a na výstupech značí, kolik bylo použito fuzzy množin. Údaj v závorce uvnitř bloku pak říká, kolik pravidel je potřeba pro hodnocení vytvořit. Celkový počet pravidel, potřebných pro vyhodnocení tohoto dekomponovaného modelu, je 398. Porovnáme-li toto číslo s předchozím případem, tak je zřejmé, že tento model se bude podstatně lépe ladit.

Rozdělení na bloky podle prezentovaného schématu však není nahodile vytvořené, nýbrž logicky promyšlené. **Blok A** integruje data o zastavěné ploše a o ploše pozemku do jednoho výstupu, tyto vstupy spolu logicky souvisí. **Blok C** zase integruje data o stavu domu a jeho energetické náročnosti, tyto informace spolu opět souvisí, poněvadž to, zda je dům zateplen, se odráží v jeho energetické náročnosti. Zateplení domu pak pochopitelně reflektuje stav domu.

U *bloku B* již souvislost vstupů tolik na první pohled patrná není, vypuštěním tohoto bloku by však bylo nutné jeho vstupy přiřadit bloku, který vyhodnocuje celý objekt, tím by však u něj podstatně narostl počet pravidel. Podíváme-li se na celou situaci podrobněji, zjistíme, že veškeré tyto parametry ovlivňují cenu nemovitosti. Pak lze na tento problém pohlížet tak, že když je nemovitost situována na špatném místě (např. rušná část obce), ale prodává se včetně vybavení, tak bude mít vyšší pořizovací cenu, než nemovitost bez vybavení. Tuto skutečnost je potřeba vzít v potaz při hodnocení, proto vstupy rušná část obce a vybavení dají lepší výstup, než rušná část obce bez vybavení. Blok B tedy funguje na tomto popisovaném principu.

Výstupy výše popisovaných bloků jsou, spolu s cenou, vstupy bloku, který má na starost hodnocení celého objektu. Výstup tohoto bloku pak vstupuje do bloku celkového hodnocení, který má dále vstup vzdálenost.

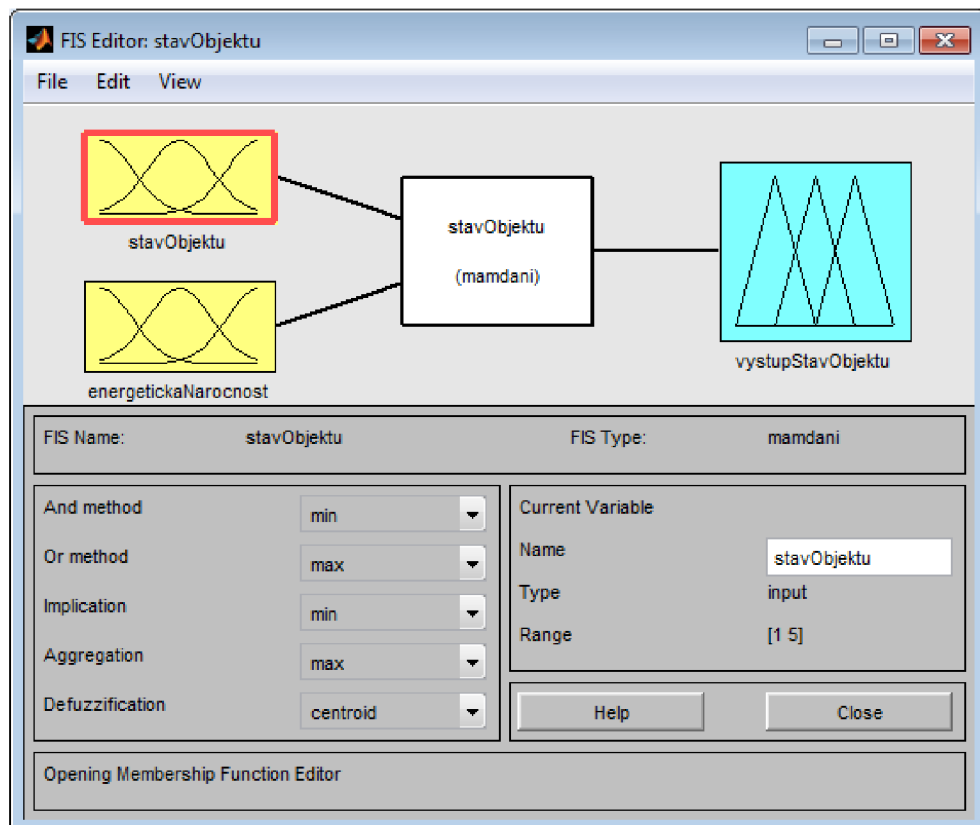


Obrázek 4.8: Schéma fuzzy modelu pro hodnocení domů

(Zdroj: Vlastní zpracování)

4.5.1.1 Vytvoření modelu ve Fuzzy Logic Toolboxu v programu MATLAB

V prostředí MATLAB se Fuzzy Logic Toolbox vyvolá příkazem *fuzzy*, v této práci bude použit fuzzy model typu *Mamdani*. Na **obrázku 4.9** je zachycen snímek obrazovky, na kterém je demonstrováno vytvoření bloku C, který hodnotí stav objektu. Postup pro ostatní bloky modelu je obdobný, vychází se z prezentovaného schématu, proto bude postup vždy demonstrován pouze na jednom příkladu vybraného bloku.

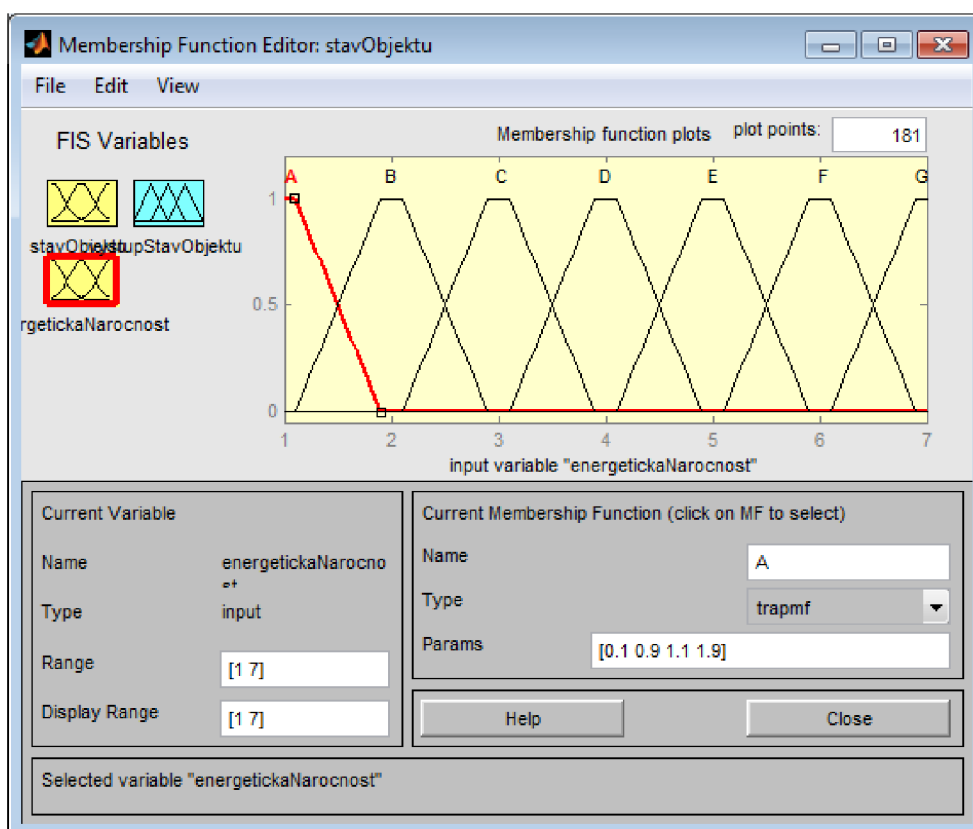


Obrázek 4.9: FIS Editor – Vytvoření bloku C

(Zdroj: Vlastní zpracování)

Pro jednotlivé proměnné je pak potřeba navolit funkce členství. Na **obrázku 4.10** jsou zachyceny funkce členství pro proměnnou energetická náročnost. Vrátime-li se zpět do kapitoly, kde byl představen ER diagram, zjistíme, že tato proměnná je číselník, který nabývá hodnot A – G, což odpovídá dané energetické třídě budovy. Číselníky nejsou z hlediska modelování příliš zajímavé, nenabízí moc možností, jak daný problém

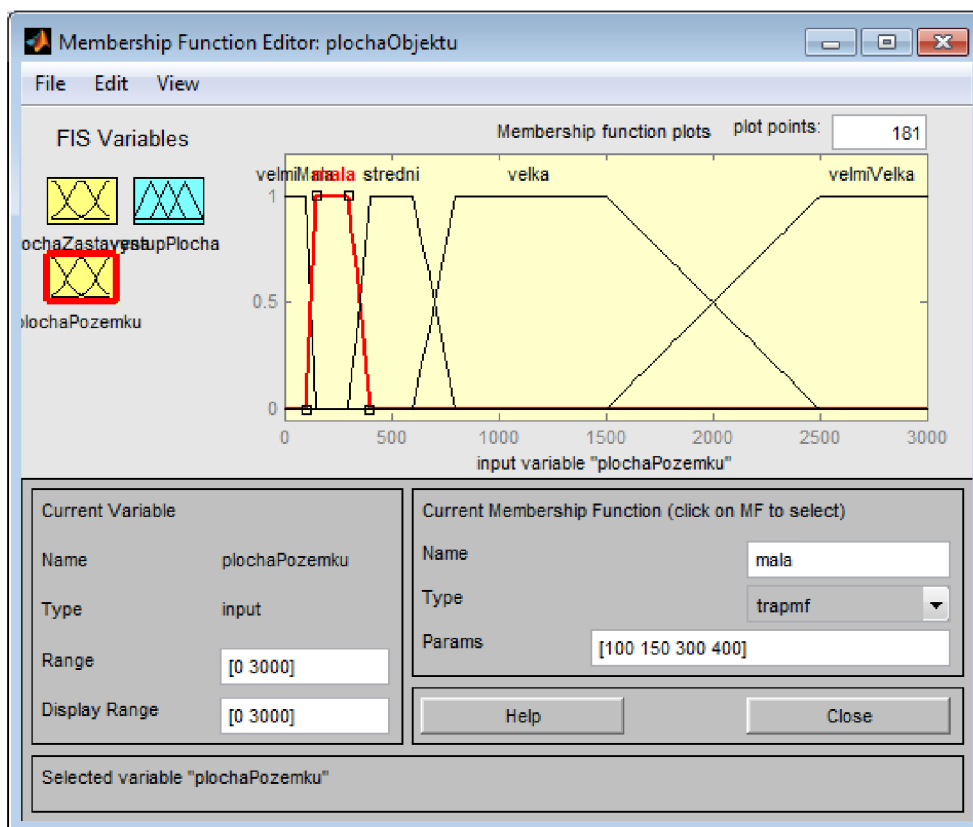
uchopit. V tomto konkrétním případě byly použity tvary funkce členství typu Π , samozřejmě by bylo možné použít i funkce členství jiného typu, např. Λ .



Obrázek 4.10: FIS Editor – Funkce členství proměnné energetická náročnost

(Zdroj: Vlastní zpracování)

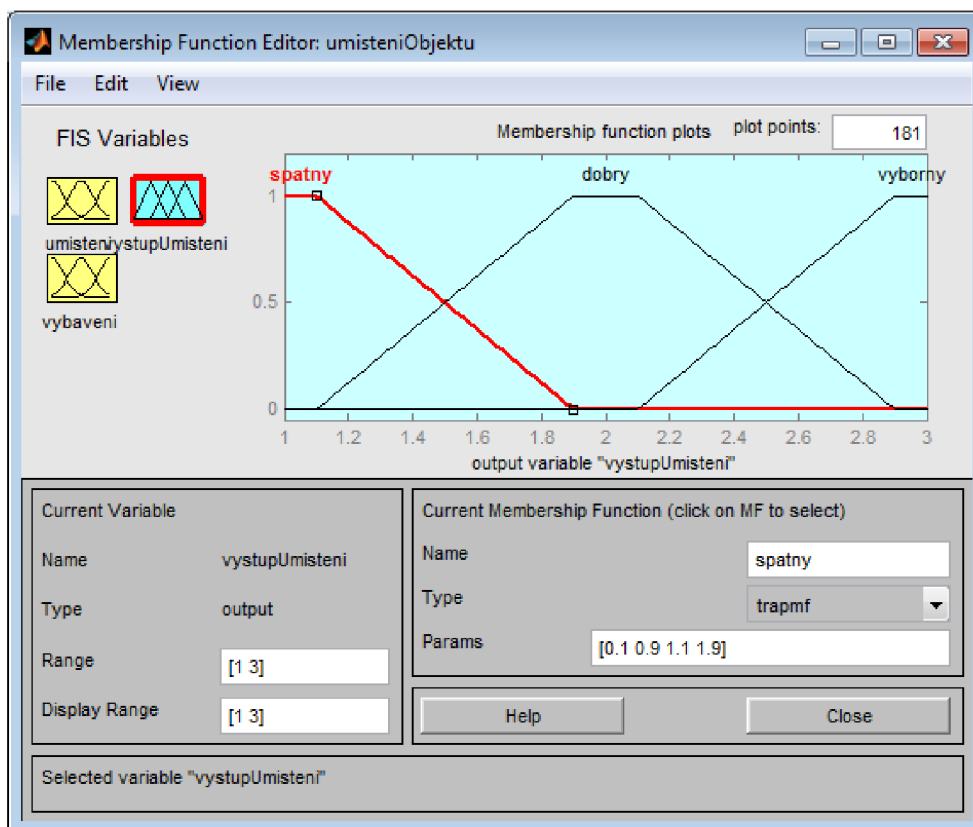
Jak dokládá **obrázek 4.11**, plocha pozemku je pro modelování mnohem lepším příkladem. Opět zde jsou použity tvary funkcí členství typu Π . Fuzzy množiny *velmiMala*, *mala* a *stredni* jsou poměrně dosti „nahuštěny“, tento fakt však odráží osobní pohled firmy, který je podložen analýzou. Interval hodnot, který tyto množiny pokrývají, je zároveň i intervalem, jehož hodnoty se u sledovaného atributu nemovitosti nejčastěji vyskytují a liší se v řádech desítek metrů čtverečních, proto bylo nutné tento interval dále rozdělit mezi již zmíněné fuzzy množiny. Naproti tomu u nemovitostí, jejichž plochu pozemku pokrývá fuzzy množina *velka*, se již hodnota tohoto atributu může lišit v řádech sta metrů čtverečních.



Obrázek 4.11: FIS Editor – Funkce členství proměnné plocha pozemka

(Zdroj: Vlastní zpracování)

Na *obrázku 4.12* je zachycen výstup bloku C. Tento výstup je pak vstupem bloku, který vyhodnocuje celý objekt, návrh vstupu bude vypadat obdobně, jak je tomu u zde prezentovaného výstupu. Jinými slovy, budou použité stejné fuzzy množiny. U všech bloků, kromě posledního, si vystačíme s číselnou hodnotou výstupu, poněvadž je vždy následně vstupem pro další blok. U posledního bloku, který má na starost celkové hodnocení, je pak potřeba číselnou hodnotu převést na hodnotu jazykovou, tedy na kvalitativní slovní vyjádření.



Obrázek 4.12: FIS Editor – Výstup bloku C

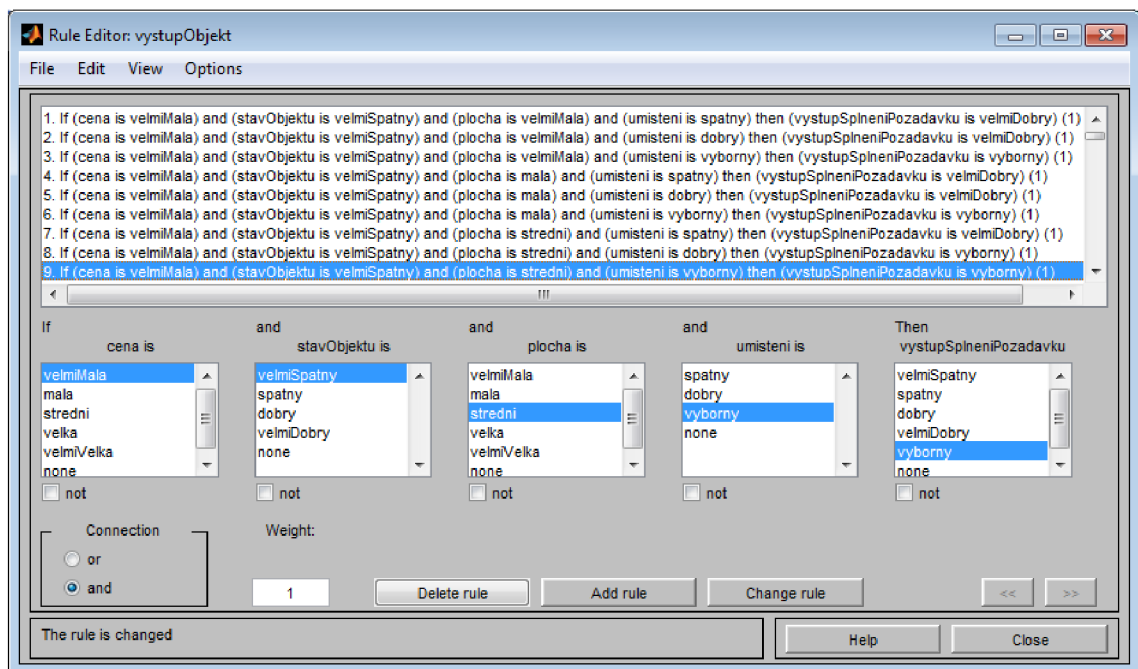
(Zdroj: Vlastní zpracování)

Vytváření pravidel

Dalším nezbytným krokem je vytvoření báze pravidel, jinak by modelovaný fuzzy systém nevěděl, jak se má chovat. Ze schématu na **obrázku 4.8** již známe počty potřebných pravidel pro jednotlivé subsystémy, pro jejich definování lze použít v prostředí MATLAB Fuzzy Logic Toolbox editor pravidel, viz **obrázek 4.13**. GUI rozhraní, které tento editor nabízí je sice pohodlné, nicméně je potřeba definovat výsledek pro všechny potřebné kombinace. Druhou možností je psát pravidla ručně přímo do FIS souboru, viz ukázka prvních deseti pravidel bloku celkového hodnocení objektu na **obrázku 4.14**.

Při malém počtu pravidel nejspíše nezáleží na tom, jaký způsob si vybereme, při formulaci většího počtu pravidel, jak je tomu i v této práci, však člověk již hledá cestu, jak si tuto činnost, pokud možno, ulehčit. Jedna taková cesta se ukrývá v kombinaci obou předchozích způsobů zápisu. Nejprve je potřeba si nechat vygenerovat všechny

možné kombinace vstupů spolu s defaultně nastaveným výstupem. Formát vygenerovaných pravidel musí odpovídat syntaktickému zápisu pravidel v souboru FIS, tak jak je ukázáno na **obrázku 4.14**. Když pak tento soubor s vygenerovanými pravidly otevřeme v editoru pravidel, budeme zde již mít veškeré kombinace. U každého pravidla již jen stačí změnit defaultní výstup na námi požadovaný.



Obrázek 4.13: Fuzzy Logic Toolbox – Editor pravidel

(Zdroj: Vlastní zpracování)

```
[Rules]
1 1 1 1, 4 (1) : 1
1 1 1 2, 4 (1) : 1
1 1 1 3, 5 (1) : 1
1 1 2 1, 4 (1) : 1
1 1 2 2, 4 (1) : 1
1 1 2 3, 5 (1) : 1
1 1 3 1, 4 (1) : 1
1 1 3 2, 5 (1) : 1
1 1 3 3, 5 (1) : 1
1 1 4 1, 4 (1) : 1
```

Obrázek 4.14: FIS soubor – Způsob zápisu pravidel

(Zdroj: Vlastní zpracování)

Aby bylo možné generátor pravidel vytvořit, je potřeba porozumět zápisu pravidel v souboru FIS. Každé pravidlo se zapisuje na nový řádek, nejprve se zapíše indexy odpovídající funkcím členství vstupních proměnných. Následuje čárka, za kterou se zapíše index funkce členství, která odpovídá výstupní proměnné. Pokud potřebujeme negaci, index se zapíše se záporným znaménkem. Číslo v závorce pak udává váhu pravidla, přičemž jsou povoleny hodnoty z intervalu $<0; 1>$. Poslední číselný údaj, který následuje za dvojtečkou, pak říká, jestli je použit operátor AND (hodnota 1) nebo operátor OR (hodnota 2) [38].

Jednoduché ad hoc řešení generátoru pravidel pro dva vstupy je znázorněno na **obrázku 4.15**.

```
public void kombinace(int x, int y) {
    for(int i = 1; i <= x; i++) {
        for(int j = 1; j <= y; j++) {
            System.out.println(i + " " + j + " " + ", 1 (1) : 1");
        }
    }
}
```

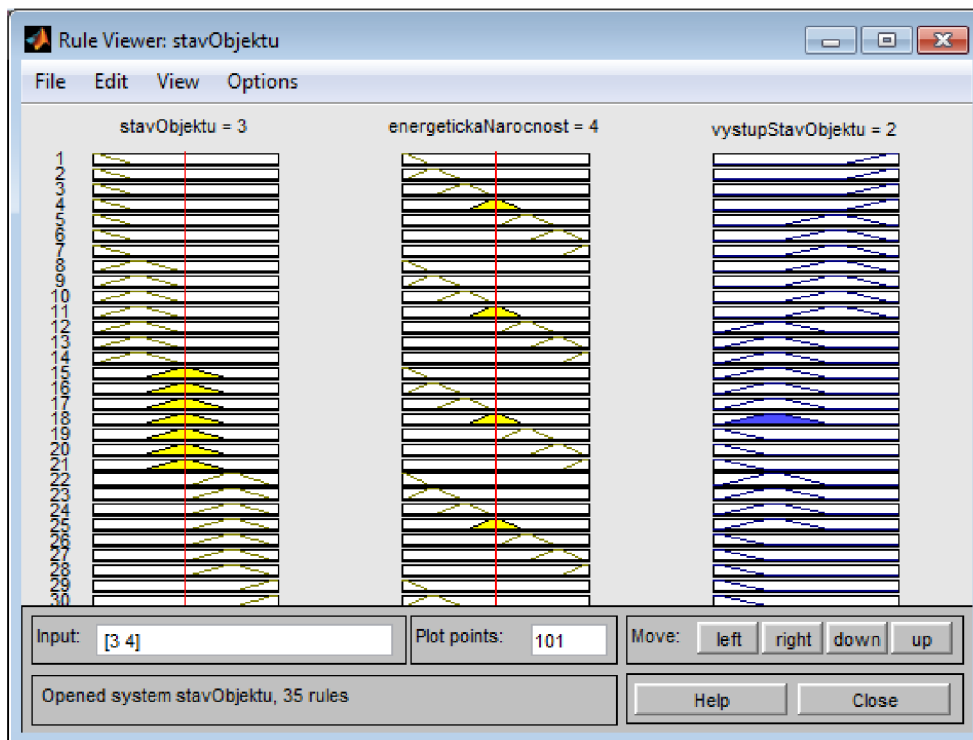
Obrázek 4.15: Ad hoc řešení generátoru pravidel pro dva vstupy

(Zdroj: Vlastní zpracování)

Užitečnými nástroji pro testování a vyladění celého fuzzy systému jsou pak **Rule Viewer** a **Surface Viewer**.

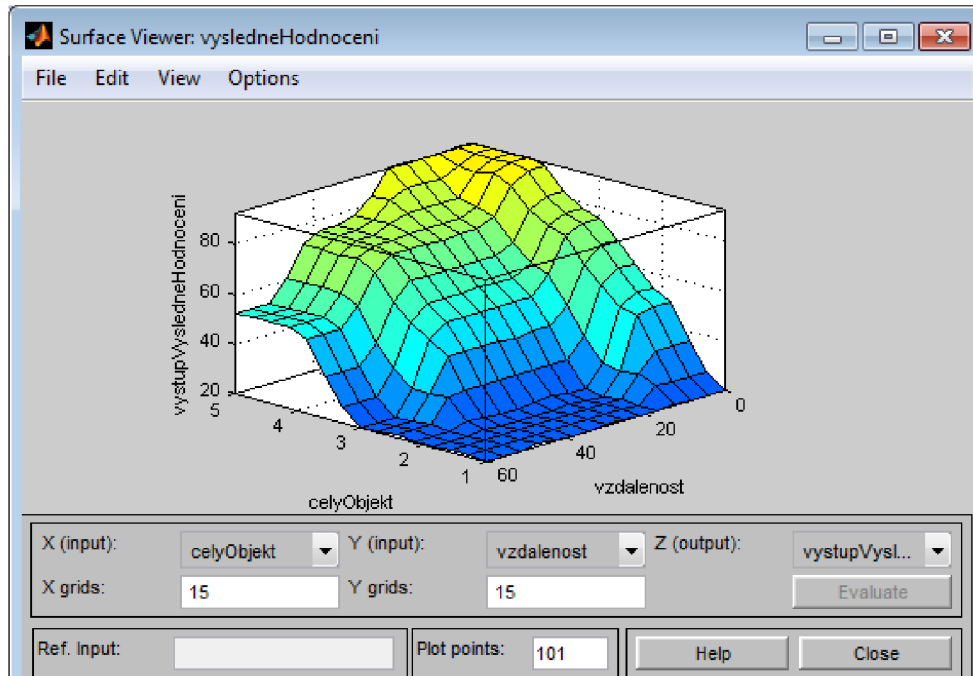
Na **obrázku 4.16** je zachyceno použití **Rule Viewer** pro testování bloku C. Jednotlivé řádky jsou grafické ekvivalenty k nadefinovaným pravidlům. Testování lze provádět dvojím způsobem, na prezentovaném snímku byly do pole **Input** zadány hodnoty vstupů **stavObjektu** a **energetickaNarocnost** v podobě [3 4]. Na základě pravidel odpovídá tomuto vstupu výstup 2. Druhý způsob spočívá v posouvání červené vertikální čáry, tímto způsobem lze sledovat, jak se celý model chová.

Použití **Surface Viewer** je pak demonstrováno na **obrázku 4.17**, jedná se o blok **celkového hodnocení**, kde je zkoumána závislost vstupů **celyObjekt** a **vzdalenost**, spolu s výstupem. Trojrozměrný pohled umožňuje lépe zhodnotit a ověřit, zda návrh pravidel odpovídá požadavkům.



Obrázek 4.16: Ruler Viewer – Testování modelu

(Zdroj: Vlastní zpracování)



Obrázek 4.17: Surface Viewer

(Zdroj: Vlastní zpracování)

4.5.2 Fuzzy model pro hodnocení bytů

Při tvorbě fuzzy modelu pro nemovitost typu byt je potřeba počítat s více vstupními proměnnými, než tomu bylo u nemovitosti typu dům. Výčet vstupních proměnných, spolu s přiděleným počtem fuzzy množin (uvedeno v závorce), je následující: Sklep (2), Výtah (2), Garáž (2), Balkón (2), Lodžie (2), Umístění (6), Vybavení (3), Podlaží (3), Vlastnictví (2), Stav (5), Energetická třída (7), Cena (5), Užitná plocha (5), Vzdálenost (4).

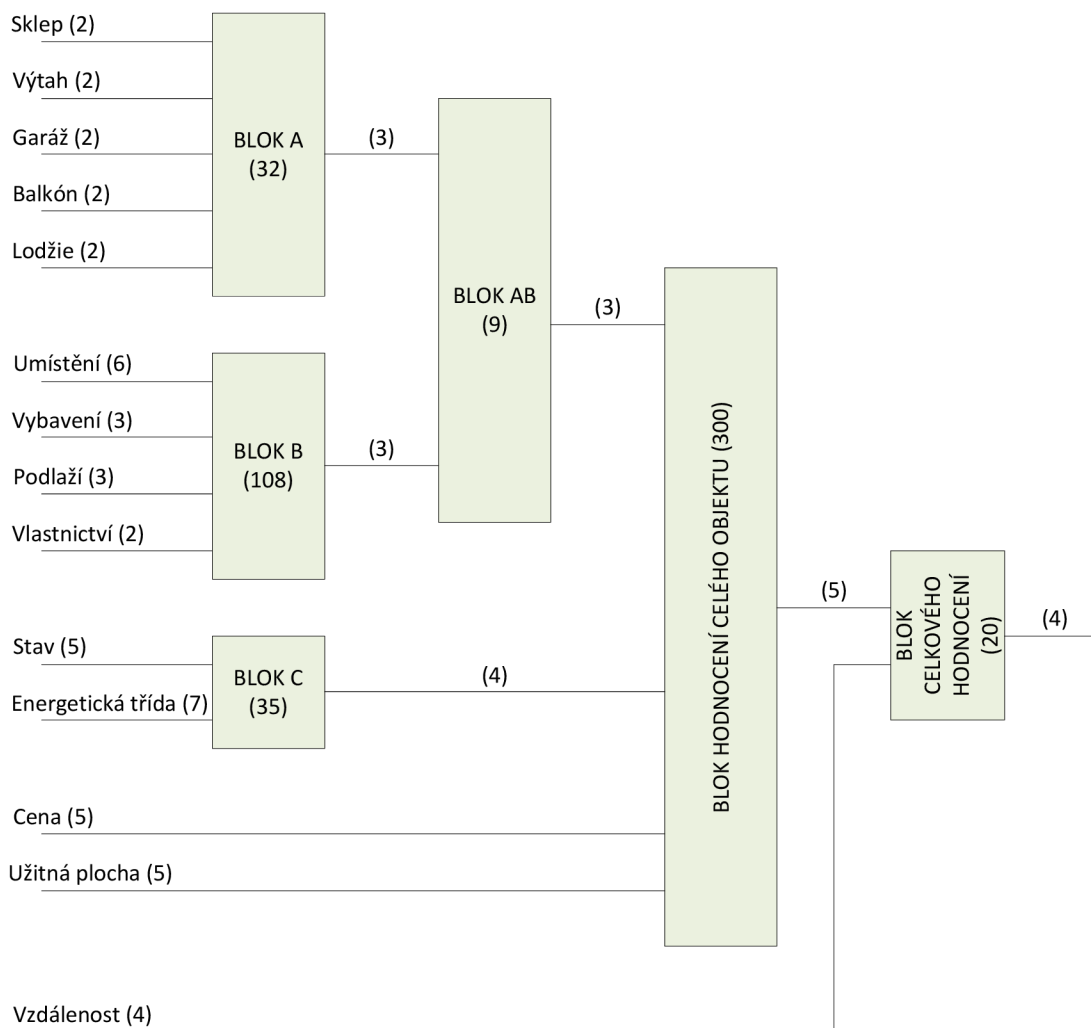
Jedná se tedy celkem o 14 vstupních proměnných, kdyby se celý fuzzy model skládal pouze z jednoho bloku, do kterého by směřovaly veškeré vstupy, obsahoval by tento model velké množství pravidel, k jehož počtu lze dojít stejným způsobem, jak tomu bylo v případě domů:

$$POČET PRAVIDEL = 2 \times 2 \times 2 \times 2 \times 2 \times 6 \times 3 \times 3 \times 2 \times 5 \times 7 \times 5 \times 5 \times 4 = \underline{12096000}$$

Výsledný počet pravidel je podstatně větší, než tomu bylo u domů, formulace pravidel a celkové ladění by tak byl velice náročný úkon, nehledě také na potřebný výpočetní výkon. Z tohoto důvodu bude opět provedena dekompozice celého problému, podobně, jako tomu bylo u fuzzy modelu domů.

Výsledné schéma je znázorněno na **obrázku 4.18**. Pro dekompozici bylo použito celkem 6 vyhodnocovacích bloků. Tímto postupem byl původní potřebný počet pravidel eliminován na 504.

Postup tvorby fuzzy modelu v prostředí programu MATLAB je identický, jak tomu bylo v případě tvorby fuzzy modelu pro domy, proto zde již nebude znovu popisován.



Obrázek 4.18: Schéma fuzzy modelu pro hodnocení bytů

(Zdroj: Vlastní zpracování)

4.5.3 Vytvoření fuzzy modelů pomocí knihovny jFuzzyLogic

MATLAB nabízí opravdu propracované prostředí pro tvorbu fuzzy modelů, jak to u takových profesionálních nástrojů ale většinou bývá, nejedná se o bezplatný volně dostupný software. Školní licence umožňuje používat tento program pouze k účelům výuky a výzkumu, komerční použití tak není možné. Firma, která bude výslednou aplikaci používat, žádnou licenci tohoto softwaru nevlastní, proto je potřeba hledat jiné, nejlépe zdarma dostupné, alternativy.

Aplikace je vyvíjena v jazyku Java, proto by bylo vhodné použít nějakou dostupnou open source knihovnu, která by vytvoření a používání fuzzy modelu podporovala. Jednou takovou je i knihovna *jFuzzyLogic*.

Vývoj fuzzy systémů zde probíhá pomocí standardizovaného jazyka FCL. Příklad části kódu, který popisuje vytvoření bloku pro hodnocení celého objektu, je znázorněn na *obrázku 4.19*. MATLAB však využívá svůj vlastní jazyk, pro porovnání syntaxe jednotlivých jazyků je na *obrázku 4.20* k dispozici ukázka části kódu v souboru FIS, který se týká stejného subbloku.

Pokud již máme k dispozici hotový fuzzy model, který byl vytvořený v MATLABu, stačí již jen přepsat potřebné FIS soubory do jazyka FCL. Podíváme-li se např. na fuzzifikaci proměnné cena, tak si lze povšimnout různého zápisu funkcí členství. Zatímco ve FIS souboru je přímo definovaný tvar *trapmf* (trapezový tvar) a pak již jen souřadnice *osy x*, jazyk FCL žádnou definici tvaru v sobě nezahrnuje. Funkce členství je zde definována pomocí *bodů*, používají se tedy souřadnice jak *osy x*, tak také *osy y*. Podobných odlišností zápisu je pochopitelně více, v zásadě však přepis již existujícího FIS souboru do jazyka FCL nečiní výrazné obtíže.

```
1 FUNCTION_BLOCK celyObjekt
2
3 // Define input variables
4 VAR_INPUT
5     cena : REAL;
6     stavObjektu : REAL;
7     plocha : REAL;
8     umistení : REAL;
9 END_VAR
10
11 // Define output variable
12 VAR_OUTPUT
13     vystupSplneniPozadavku : REAL;
14 END_VAR
15
16 // Fuzzify input variable 'cena'
17 FUZZIFY cena
18     TERM velmiMala := (0, 0) (0, 1) (0.8, 1) (1.1, 0);
19     TERM mala := (0.8, 0) (1.1, 1) (1.5, 1) (1.8, 0);
20     TERM stredni := (1.5, 0) (1.8, 1) (2.5, 1) (2.9, 0);
21     TERM velka := (2.5, 0) (2.9, 1) (3.5, 1) (4, 0);
22     TERM velmiVelka := (3.5, 0) (4, 1) (5.0, 1) (5.0, 0);
23 END_FUZZIFY
```

Obrázek 4.19: Ukázka části kódu bloku pro vyhodnocení celého objektu v jazyku FCL

(Zdroj: Vlastní zpracování)

```

1 [System]
2 Name='celyObjekt'
3 Type='mamdani'
4 Version=2.0
5 NumInputs=4
6 NumOutputs=1
7 NumRules=300
8 AndMethod='min'
9 OrMethod='max'
10 ImpMethod='min'
11 AggMethod='max'
12 DefuzzMethod='centroid'
13
14 [Input1]
15 Name='cena'
16 Range=[0 5]
17 NumMFs=5
18 MF1='velmiMala':'trapmf',[-1.125 -0.125 0.8 1.1]
19 MF2='mala':'trapmf',[0.8 1.1 1.5 1.8]
20 MF3='stredni':'trapmf',[1.5 1.8 2.5 2.9]
21 MF4='velka':'trapmf',[2.5 2.9 3.5 4]
22 MF5='velmiVelka':'trapmf',[3.5 4 5.125 6.125]

```

Obrázek 4.20. Ukázka části kódu bloku pro vyhodnocení celého objektu v souboru FIS

(Zdroj: Vlastní zpracování)

V případě pravidel je situace již poněkud trochu obtížnější. Tím ovšem není myšlená samotná forma zápisu, ale jejich počet. Přepisování velkého počtu pravidel je pracná záležitost, proto by bylo namístě, když jsou již pravidla jednou vytvořena v souboru FIS, tento úkon automatizovat.

V zásadě se opět nabízí více způsobů, jak tuto automatickou konverzi provést. První možností je vycházet přímo ze souboru FIS, kde jsou pravidla zapsána ve formě indexů. Druhý způsob spočívá v použití příkazu *showrule(fis)*, který vypíše veškerá pravidla fuzzy modelu ve formátu, který se již více podobá zápisu pravidel v jazyku FCL. Porovnání obou zápisů je zachyceno na **obrázku 4.21**.

I když skript, který by převáděl pravidla z indexové formy zápisu, by nebyl nikterak složitý, druhá varianta řešení bude ještě o něco jednodušší. Podíváme-li se pozorně na formát pravidla, které obdržíme příkazem *showrule(fis)*, tak pro převod stačí, aby každé pravidlo začínalo řetězcem „RULE“. Dále pak je potřeba zaměnit tečku, která následuje za číslem udávající pořadí pravidla, za dvojtečku. Klíčová slova,

jako např. *if* nebo *and*, je potřeba převést na velká písmena. Poslední nutnou operací pak je odstranění závorek. Vytvořit potřebný skript tak je celkem jednoduchou záležitostí.

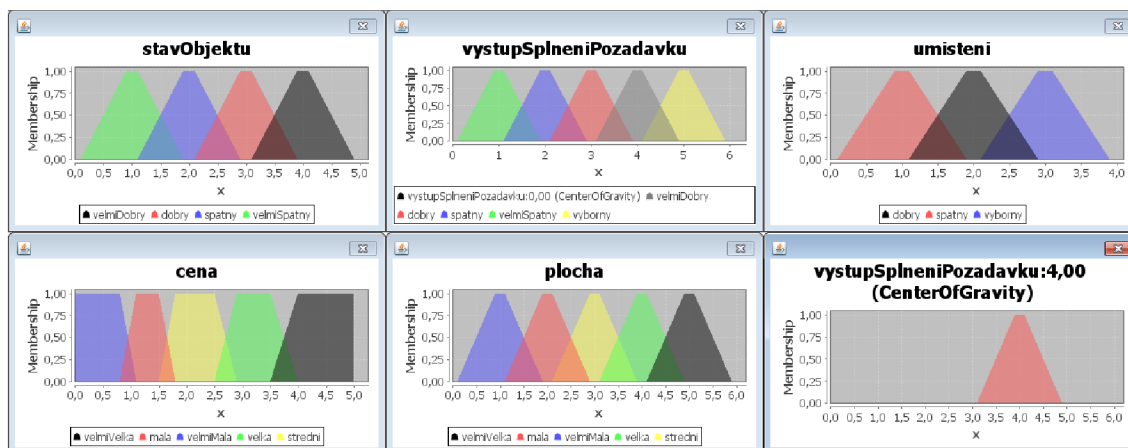
```
1. If (cena is velmiMala) and (stavObjektu is velmiSpatny) and
(placha is velmiMala) and (umistení is spatny) then
(vystupSplneniPozadavku is velmiDobry) (1)
```

```
RULE 1 : IF cena IS velmiMala AND stavObjektu IS velmiSpatny AND
plocha IS velmiMala AND umistení IS spatny THEN
vystupSplneniPozadavku IS velmiDobry;
```

Obrázek 4.21: Porovnání zápisu pravidel

(Zdroj: Vlastní zpracování)

Knihovna *jFuzzyLogic* sice nenabízí podobné prostředky pro ladění fuzzy modelu, jak je tomu v programu MATLAB, nicméně užitím *jFuzzyLogic's Eclipse pluginu* lze provádět i zde grafické testování. Vizualizace výsledku a funkcí příslušnosti u jednotlivých proměnných, je pro blok celkového hodnocení objektu, znázorněna na *obrázku 4.22*.



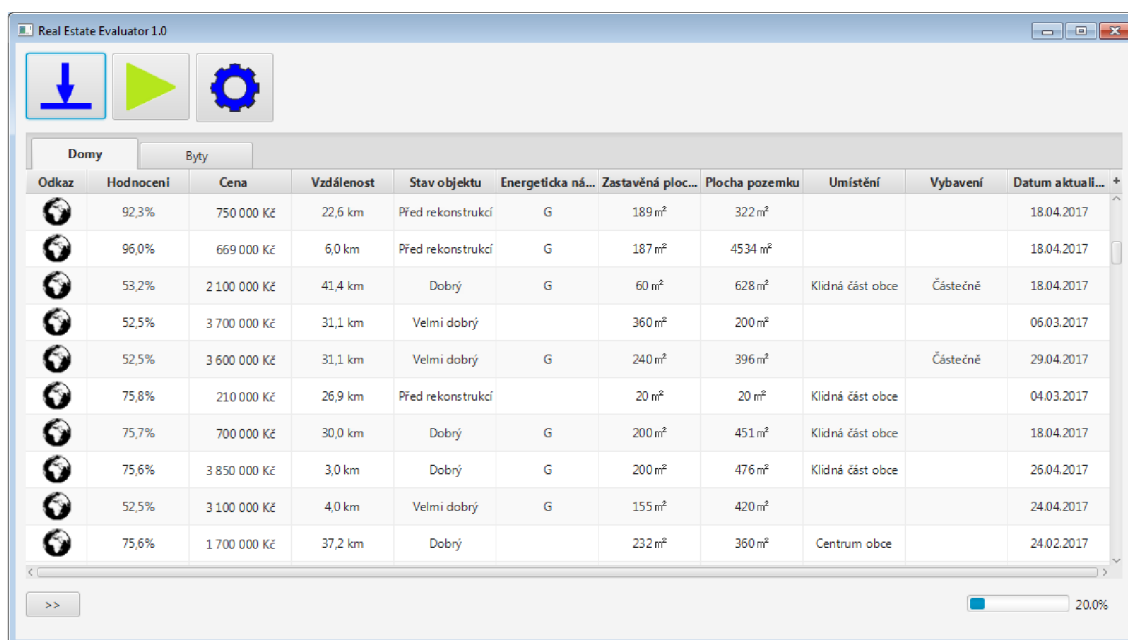
Obrázek 4.22: Vizualizace výsledku a funkcí příslušnosti

(Zdroj: Vlastní zpracování)

4.6 Výsledná hotová aplikace

Výsledná hotová aplikace je zachycena na **obrázku 4.23**. Grafické uživatelské rozhraní bylo vytvořeno v **JavaFX**. Mezi hlavní požadavky firmy také patřilo jednoduché a srozumitelné ovládání celé aplikace, na což byl brán při návrhu zřetel.

Prvním tlačítkem zleva se provede stáhnutí nových inzerátů, zároveň se provede kontrola stávajících uložených záznamů, zda jsou ještě aktivní. První tlačítko tedy slouží k aktualizaci databáze. Na výběr jsou dva typy aktualizací, jedna provede pouze přidání nově dostupných inzerátů a odstranění neaktivních záznamů (nejedná se však o fyzické odstranění, záznamy pouze nebudou figurovat ve výpisu). Druhá (hloubkovější) kontrola pak kontroluje aktuálnost dat již dříve uložených záznamů, např. zda nemovitost nebyla zlevněna. Aktualizace probíhá na základě seznamu URL adres, jelikož je po získání tohoto seznamu znám počet nových inzerátů, které je potřeba stáhnout, je možné o celkovém průběhu činnosti informovat uživatele. K tomuto účelu slouží indikátor průběhu v pravém dolním rohu. Bude-li např. nalezeno 20 nových záznamů, pak 5% průběhu bude odpovídat uložení 1 záznamu, 20% pak 4 záznamům atd.



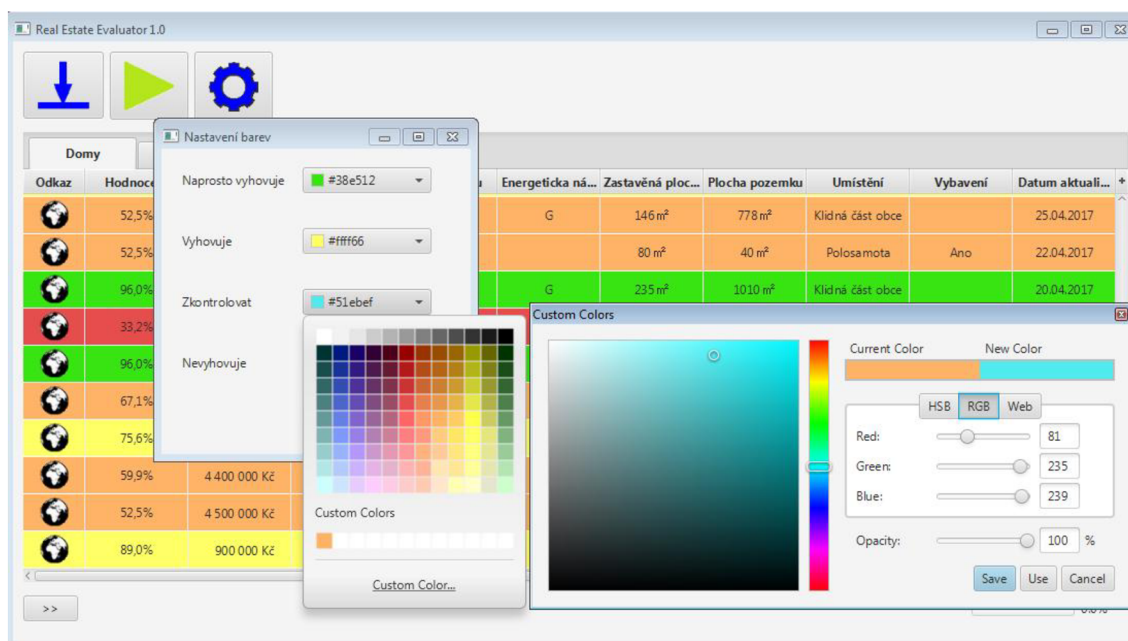
Odkaz	Hodnocení	Cena	Vzdálenost	Stav objektu	Energetická ná...	Zastavěná ploch...	Plocha pozemku	Umístění	Vybavení	Datum aktualiz...
	92,3%	750 000 Kč	22,6 km	Před rekonstrukcí	G	189 m ²	322 m ²			18.04.2017
	95,0%	669 000 Kč	6,0 km	Před rekonstrukcí	G	187 m ²	4534 m ²			18.04.2017
	52,2%	2 100 000 Kč	41,4 km	Dobry	G	60 m ²	628 m ²	Klidná část obce	Částečné	18.04.2017
	52,5%	3 700 000 Kč	31,1 km	Velmi dobrý		360 m ²	200 m ²			05.03.2017
	52,5%	3 600 000 Kč	31,1 km	Velmi dobrý	G	240 m ²	396 m ²		Částečné	29.04.2017
	75,8%	210 000 Kč	26,9 km	Před rekonstrukcí		20 m ²	20 m ²	Klidná část obce		04.03.2017
	75,7%	700 000 Kč	30,0 km	Dobry	G	200 m ²	451 m ²	Klidná část obce		18.04.2017
	75,6%	3 850 000 Kč	3,0 km	Dobry	G	200 m ²	476 m ²	Klidná část obce		26.04.2017
	52,5%	3 100 000 Kč	4,0 km	Velmi dobrý	G	155 m ²	420 m ²			24.04.2017
	75,6%	1 700 000 Kč	37,2 km	Dobry		232 m ²	360 m ²	Centrum obce		24.02.2017

Obrázek 4.23: Hotová aplikace – Aktivní karta s domy

(Zdroj: Vlastní zpracování)

Druhé tlačítko zleva pak slouží k výpisu záznamů a aplikaci prováděných změn. Pokud jsou např. spuštěny aktualizace, patřičné změny se projeví až po kliknutí na toto tlačítko. Je to z toho důvodu, že během aktualizací si uživatel může prohlížet záznamy, jakákoliv změna sestavy při této činnosti by měla nechtěný efekt.

Poslední nepopsané tlačítko pak slouží k uživatelskému přizpůsobení výpisu. Kromě defaultního vzhledu, který byl zobrazen na **obrázku 4.23**, je možné také jednotlivé záznamy od sebe barevně odlišit, viz **obrázek 4.24**. V aplikaci je použita pro hodnocení nemovitosti číselná proměnná, která je výstupem fuzzy modelu. Její převod na slovní hodnocení není použit z toho důvodu, aby při řazení a následném porovnávání záznamů byly patrné větší rozdíly. Uživatel má však možnost jednotlivé záznamy barevně odlišit právě podle odpovídajícího slovního hodnocení. Barevné vnímání je čistě subjektivní záležitostí, proto je lepší, když si uživatel navolí barevné schéma sám. K tomuto účelu je možné využít jak základní, tak také rozšířenou paletu barev. Záznamy lze samozřejmě řadit, proto může někomu barevné odlišení připadat zbytečné, v takovém případě je možné využít jen defaultní schéma.



Obrázek 4.24: Hotová aplikace – Barevné odlišení záznamů

(Zdroj: Vlastní zpracování)

Řazení záznamů je samozřejmě možné také podle více klíčů (sloupců tabulky). Na **obrázku 4.25** je pro změnu tentokrát zachycen výpis bytů, vícekritériální řazení se provede tak, že se přidrží klávesa SHIFT a vybírají se požadované sloupce v pořadí, v jakém chceme mít prioritu řazení (jsou-li položky totožné, řadí se podle dalšího klíče v pořadí atd.). Na uvedeném obrázku je nejprve vybráno řazení od nejnižší ceny, následně podle nejlepšího hodnocení a nakonec od nejkratší vzdálenosti. Pořadí výběru ilustrativně demonstruje počet teček, které se nacházejí nad nebo pod znakem, který vyjadřuje, zda se řadí sestupně, či vzestupně.

Jednotlivé sloupce je možné libovolně přesouvat přetažením myši (drag and drop), dále pak měnit jejich šířku, případně vybrat, které sloupce mají být zobrazeny a které naopak skryty.

Součástí záznamu je rovněž URL adresa inzerátu, dvojklikem na záznam se tak otevře webový prohlížeč s konkrétním inzerátem. Pro informování uživatele o probíhaných akcích slouží kromě indikátoru průběhu také jednoduchý log, který lze zobrazit, případně skrýt, pomocí tlačítka v levém spodním rohu aplikace. Uživatel tak může získat užitečné informace o tom, kolik přibylo nových záznamů nebo kolik se jich stalo neaktivními.

Odkaz	Hodnocení	Cena	Vzdálenost	Stav objektu	Energetická ná...	Užitná plocha	Umístění	Vybavení	Datum aktual...	Sklep	Výtah	Garáž	Balkon	Lodže
	93,9%	610 000 Kč	5,2 km	Dobný		44 m ²	Okraj obce		08.03.2017	Ano	Ano			
	75,6%	635 000 Kč	3,0 km	Velmi dobrý		43 m ²			21.04.2017	Ano				
	95,0%	850 000 Kč	5,2 km	Dobný		44 m ²			18.01.2017	Ano	Ano			
	75,6%	850 000 Kč	4,0 km	Velmi dobrý		36 m ²			13.04.2017	Ano				
	89,0%	900 000 Kč	15,0 km	Dobný	G	68 m ²	Klidná část obce		05.04.2017					
	96,0%	1 135 000 Kč	3,4 km	Před rekonstrukcí	C	76 m ²			18.04.2017	Ano				
	96,0%	1 260 000 Kč	5,8 km	Dobný	G	73 m ²	Okraj obce	Ne	26.04.2017	Ne	Ano			Ne
	52,5%	1 365 000 Kč	29,3 km	Velmi dobrý		32 m ²		Částečné	26.04.2017		Ano			Ne

Prohlašuje aktualizace databáze...
Počet nových záznamů: 26
Počet neaktivních záznamů: 18

Obrázek 4.25: Hotová aplikace – Řazení podle více sloupců

(Zdroj: Vlastní zpracování)

4.7 Návrhy do budoucna

Výsledná hotová aplikace již byla firmě prezentována, proto je možné do budoucích návrhů také zakomponovat již konzultované návrhy a připomínky firmy.

První návrh spočívá v automatizaci probíhaných aktualizací, jinými slovy, aby aktualizace probíhaly „v tichosti“ v pozadí automaticky, bez nutnosti zásahu uživatele. Princip by byl podobný, jak je tomu např. u antivirových programů, při aktualizaci virové databáze.

Aplikace by se tedy spustila ihned po naběhnutí operačního systému, uživatel by pouze nastavil intervaly, v jakých mají být aktualizace prováděny. S tímto krokem rovněž souvisí zavedení (tray) ikony do oznamovací oblasti a také patřičných upozornění, které budou uživatele o dostupnosti nových inzerátů informovat.

Mezi další návrhy patří rozšíření podpory aplikace o více realitních serverů. I když je Sreality.cz nejnavštěvovanějším realitním serverem a inzeráty na ostatních realitních serverech se budou většinou pouze opakovat, pořád existuje jistá šance na nalezení unikátního inzerátu. Tento návrh si však sebou také vyžádá potřebu podrobně propracovat systém na detekci duplicitních záznamů, poněvadž skutečnost, že bude nemovitost inzerována zároveň na více realitních serverech ještě neznamená, že musí mít inzeráty rovněž stejné údaje (někde mohou např. některé údaje chybět).

Při testování aplikace se firma vyjádřila v tom smyslu, že v dohledné době se chce, kromě domů a bytů, zaměřit také na pozemky. Vyslovila tedy přání, zda by nebylo možné aplikaci, o schopnost hodnotit také pozemky, rozšířit. Tato dodatečná funkcionality je pochopitelně možná a potup implementace bude obdobný, jak tomu bylo v případě domů a bytů.

Aplikace ukládá i jiná data, než která jsou nyní k hodnocení bezprostředně nutná. Při návrhu aplikace bylo počítáno se skutečností, že v budoucnu budou tato data použita k pokročilejším analýzám, které budou sloužit k přesnějšímu hodnocení nemovitostí. Jedna z věcí, která by měla jít z dostatečného množství záznamu vyčistit, je určení lokalit, které jsou pro investici do nemovitosti rentabilnější. Tato pomyslná mapa pak bude sloužit jako další parametr fuzzy modelu.

4.8 Přínosy práce

Aplikace byla vyvíjena pro stavební firmu, která v současné době rozšiřuje svůj obor podnikání. Pro nový podnikatelský záměr potřebuje vyhledat a koupit vhodnou nemovitost. Na tomto záměru je nalezení potřebné nemovitosti právě tou nejtěžší záležitostí, poněvadž požadovaný objekt musí splňovat jisté atributy, jinak by podnikatelský záměr nebyl rentabilní.

Z výše uvedeného je zřejmé, že nalezení vhodného objektu může být otázkou týdnů, dokonce i měsíců. Každý den přibývají nové a nové inzerce s nabídkou prodeje různých nemovitostí. Na některé nabídky je potřeba reagovat co nejrychleji, poněvadž se může stát, že nemovitost již bude prodána. Ideální tedy je sledovat nabídky téměř každý den, což si také vyžaduje potřebný čas, který je potřeba této činnosti věnovat.

Realitní servery sice obsahují možnost filtrace záznamů, čímž se prohledávaná množina inzerce zmenší. Tyto filtry však většinou pracují s „tvrdými“ čísly, jinými slovy, pokud zadáme požadovanou maximální vzdálenost objektu od zvolené lokality např. 20 km, objekt, který bude vzdálen 21 km, již ve výběru nebude figurovat, přičemž by se mohlo stát, že ostatní parametry budou naprosto vyhovovat. Vzhledem k této „diskriminaci“ je většinou potřeba nastavit kritéria filtru tak, aby byla množina výsledků o něco obsáhlejší, což však sebou zase přináší větší časové nároky na její vyhodnocení.

Vyvinutá aplikace se snaží jednak eliminovat nedostatky klasických filtrů, ale především šetří čas tím, že nabídne uživateli již ohodnocený seznam nabídek. Aplikace tedy slouží jako podpůrný nástroj, který provádí automatizované vyhledávání nově dostupných nabídek, jejich následné ohodnocení na základě vytvořeného fuzzy modelu a předložení výsledků uživateli. Uživatel tedy již nemusí procházet desítky inzerátů, ale pouze ty, které nejvíce vyhovují jeho požadavkům. Aplikace tedy šetří firmě čas, který lze využít jinak a lépe.

ZÁVĚR

Cílem diplomové práce bylo vytvořit pro vybranou stavební firmu aplikaci, která bude sloužit jako podpůrný nástroj při hodnocení investičního záměru.

Poněvadž v oboru stavebnictví neustále roste konkurence, rozhodla se firma, z důvodu stability a nezávislosti na poptávkách zákazníků, vstoupit na realitní trh a podnikat s nemovitostmi. Hlavní idea tak spočívá v koupi nemovitosti, její následné rekonstrukci a prodeji. Problém se tedy ukrývá v nalezení vhodného objektu, který by bylo možné po jeho revitalizaci se ziskem prodat a právě s touto činností vyvinutá aplikace firmě pomáhá.

Při návrhu aplikace bylo nejdříve nutné vyřešit otázku získávání dat. Jako zdroj dat byl vybrán realitní server *Sreality.cz*, který je Single Page Aplikací. Pomocí analýzy zachycených požadavků a odpovědí se podařilo zjistit potřebné API. Jako odpověď na konkrétní požadavek je obdržena odpověď ve formátu JSON, odkud jsou získávána potřebná data o vybrané nemovitosti. K získání dat o vzdálenosti však bylo použito *Google Maps API*.

Pro perzistenci dat pak byl použit databázový systém *SQLite* a framework *Hibernate*, který umožňuje objektově-relační mapování. Samotná aktualizace dat je pak postavena na získaném seznamu URL adres jednotlivých inzerátů, přičemž kontrola aktuálnosti se skládá ze dvou typů kontrol. První typ kontroly pouze zkontroluje a stáhne nové inzeráty, případně, pokud se již některé inzeráty na serveru nevyskytují, udělá je neaktivními. Druhý druh kontroly je „hloubkovější“ a spočívá v revizi již uložených informací o dané nemovitosti, poněvadž informace v inzerátech se mohou průběžně aktualizovat.

Dalším nutným krokem pak bylo vytvoření fuzzy modelů pro hodnocení domů a bytů, na základě získaných dat. Pro návrh a odladění těchto modelů byl použit program *MATLAB*. S ohledem na fakt, že firma tento software nevlastní a aplikace byla navíc vyvíjena v jazyku *Java*, byly výsledné fuzzy modely přepsány do standardizovaného jazyka *FCL*. Tento jazyk je používán pro vývoj fuzzy systémů a využívá jej i open source knihovna *jFuzzyLogic*, která byla použita ve vyvíjené aplikaci.

Posledním krokem pak byl návrh grafického uživatelského rozhraní, které bylo vytvořeno v *JavaFX*, přičemž byly brány v potaz požadavky firmy na jednoduché a srozumitelné ovládání aplikace.

Závěrem lze konstatovat, že veškeré vytyčené cíle se podařilo splnit. Aplikace je firmou hodnocena velice pozitivně, o čemž mj. také vypovídá požadavek na její další rozšíření.

SEZNAM POUŽITÉ LITERATURY

- [1] SCHILDT, Herbert. *Java 8: výukový kurs*. Brno: Computer Press, 2016. ISBN 978-80-251-4665-1.
- [2] ORACLE CORPORATION. 1 JavaFX Overview (Release 8). *Oracle* [online]. 2014 [cit. 2016-11-11]. Dostupné z: <http://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-overview.htm#JFXST784>
- [3] HEDLEY, Jonathan. *jsoup Java HTML Parser, with best of DOM, CSS, and jquery* [online]. ©2009-2016 [cit. 2016-11-11]. Dostupné z: <https://jsoup.org/>
- [4] CINGOLANI, Pablo. *JFuzzyLogic* [online]. [cit. 2016-11-11]. Dostupné z: <http://jfuzzylogic.sourceforge.net/html/index.html>
- [5] CINGOLANI, Pablo a Jesús ALCALÁ-FDEZ. *JFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming* [online]. 2013 [cit. 2016-11-11]. Dostupné z: http://jfuzzylogic.sourceforge.net/html/pdf/Cingolani_Alcala-Fdez_jFuzzyLogic_2013_IJCIS.pdf
- [6] GONCALVES, Antonio. *Beginning Java EE 7*. Berkeley, CA: Apress, 2013. ISBN 978-143-0246-275.
- [7] RED HAT. What is Object/Relational Mapping?: Hibernate ORM. *Hibernate. Everything data.* [online]. [cit. 2017-01-10]. Dostupné z: <http://hibernate.org/orm/what-is-an-orm/>
- [8] JOSEPH B. OTTINGER, Dave MINTER a Jeff LINWOOD. *Beginning Hibernate*. Third edition. Berkeley, CA: Apress, 2014. ISBN 978-143-0265-184.

- [9] SMITH, Ben. *Beginning JSON*. CA: Apress, 2015. ISBN 978-1-4842-0203-6.
- [10] *JSON* [online]. [cit. 2017-04-29]. Dostupné z: <http://json.org/>
- [11] MAŘÍK, Vladimír, Olga ŠTĚPÁNKOVÁ, Jiří LAŽANSKÝ a kol. *Umělá inteligence*. Praha: Academia, 1993. ISBN 80-200-0496-3.
- [12] JURA, Pavel. *Základy fuzzy logiky pro řízení a modelování*. Brno: Vysoké učení technické v Brně, Nakladatelství VUTIUM, 2003. ISBN 80-214-2261-0.
- [13] MAIXNER, Ladislav. *Mechatronika: učebnice*. Brno: Computer Press, 2006. Učebnice (Computer Press). ISBN 80-251-1299-3.
- [14] DOSTÁL, Petr. *Pokročilé metody analýz a modelování v podnikatelství a veřejné správě*. Brno: Akademické nakladatelství CERM, 2008. ISBN 978-80-7204-605-8.
- [15] Prodáváte byt? Minimální úpravy zvednou jeho cenu o stovky tisíc. *E15.cz: Byznys, politika, ekonomika, finance, události* [online]. 2016 [cit. 2017-01-14]. Dostupné z: <http://zen.e15.cz/telegraf/prodavate-byt-minimalni-upravy-zvednou-jeho-cenu-o-stovky-tisic-1326024>
- [16] ZUZÁK, Vladimír. Prodáváme byt: Jak zvýšit cenu těsně před prodejem? *Peníze.cz* [online]. 2015 [cit. 2017-01-14]. Dostupné z: <http://www.penize.cz/nakup-a-prodej-nemovitosti/294248-prodavame-byt-jak-zvysit-cenu-tesne-pred-prodejem>
- [17] POJAR, Petr. Jak opravdu zvýšit hodnotu nemovitosti. *České Stavby* [online]. 2015 [cit. 2017-01-14]. Dostupné z: <http://www.ceskestavby.cz/clanky/jak-opravdu-zvysit-hodnotu-nemovitosti-24279.html>

- [18] VOSMÍK, Petr. Ne každá investice během rekonstrukce se při prodeji domu vrátí. *Novinky.cz* [online]. 2007 [cit. 2017-01-15]. Dostupné z: <https://www.novinky.cz/bydleni/jak-na-to/111334-ne-kazda-investice-behem-rekonstrukce-se-pri-prodeji-domu-vrati.html>
- [19] ZÁMEČNÍKOVÁ, Jana. Realitní byznys šlape jako hodinky. Jak dlouho ještě? *Hypindex: odborný server společnosti Fincentrum* [online]. 2016 [cit. 2017-01-15]. Dostupné z: <http://www.hypindex.cz/clanky/realitni-byznys-slape-jako-hodinky-jak-dlouho-jeste/>
- [20] HRUŠOVÁ, Monika. Fincentrum Hypindex listopad 2016: Strach ze zákona o úvěru poslal sazbu na 1,77 %. *Hypindex: odborný server společnosti Fincentrum* [online]. 2016 [cit. 2017-01-15]. Dostupné z: <http://www.hypindex.cz/clanky/fincentrum-hypindex-listopad-2016-strach-ze-zakona-o-uveru-poslal-sazbu-na-1-77/>
- [21] REALCITY. Ceny bytů rostou. Novostavby jdou na dračku. *Vyhledávání nemovitostí: REALCITY.cz* [online]. 2016 [cit. 2017-01-16]. Dostupné z: <http://www.reality.cz/magazin/10353/ceny-bytu-rostou-novostavby-jdou-dracku/>
- [22] ČESKÝ STATISTICKÝ ÚŘAD. Ceny bytů. *Český statistický úřad* [online]. [cit. 2017-01-23]. Dostupné z: https://www.czso.cz/csu/czso/ceny_bytu Ceny bytů. *Český statistický úřad* [online]. [cit. 2017-01-16]. Dostupné z: https://www.czso.cz/csu/czso/ceny_bytu
- [23] KOTULA, Daniel. 2017: Rok ohnivého kohouta v nemovitostech. *Peníze.cz* [online]. 2017 [cit. 2017-01-16]. Dostupné z: <http://www.penize.cz/nemovitosti/319400-2017-rok-ohniveho-kohouta-v-nemovitostech>

- [24] MINISTERSTVO FINANCÍ ČR. 42. Kolokvium: průzkum prognóz makroekonomického vývoje ČR (2016–2019). *Ministerstvo financí ČR* [online]. 2016 [cit. 2017-01-16]. Dostupné z: <http://www.mfcr.cz/cs/verejny-sektor/makroekonomika/makroekonomicka-predikce/2016/42-kolokvium-pruzkum-prognoz-makroekono-26749>
- [25] MINISTERSTVO FINANCÍ ČR. *Průzkum prognóz makroekonomického vývoje ČR* [online]. 2016 [cit. 2017-01-16]. Dostupné z: http://www.mfcr.cz/assets/cs/media/Odborne-vyzkumy_Prognoza_2016-11_42-Kolokvium-pruzkum-prognoz-makroekonomickeho-vyvoje-CR-2016-2019.pdf
- [26] ZÁMEČNÍKOVÁ, Jana. Hypoteční žně v roce 2017 skončí. *Hypindex: odborný server společnosti* [online]. 2017 [cit. 2017-01-16]. Dostupné z: <http://www.hypindex.cz/clanky/hypotecni-zne-roce-2017-skonci/>
- [27] ČTK a BBB. ČNB bude moci nově stanovit podmínky úvěrů na bydlení, schválila vláda. *Hospodářské noviny* [online]. 2017 [cit. 2017-01-16]. Dostupné z: <http://byznys.ihned.cz/c1-65588780-cnb-ziska-pravomoc-stanovovat-podminky-uveru-na-bydleni-schvalila-vlada>
- [28] HYPOTEČNÍ BANKA, A.S. Nový zákon o spotřebitelském úvěru mění od 1. prosince podmínky uzavírání i splácení hypotečních úvěrů. *Hypoteční banka* [online]. 2016 [cit. 2017-01-16]. Dostupné z: <https://www.hypotecnibanka.cz/o-bance/pro-media/tiskove-zpravy/ostatni/novy-zakon-o-spotrebitelskem-uveru-meni-od-1-prosince-podminky/>
- [29] REALITY PRORADOST S.R.O. Prodej rodinného domu 7+2 v Němčičkách nad Jevišovkou, RD Němčičky nad Jevišovkou. *Realitní kancelář Břeclav, nemovitosti: REALITY PRORADOST s.r.o.* [online]. ©2017 [cit. 2017-01-16]. Dostupné z: <http://www.reality-proradost.cz/prodej-rodinneho-domu-7-2-v-nemcickach-nad-jevisevkou-rd-nemcicky-nad-jevisevkou>

- [30] SCHMIDT REALITY S.R.O. Prodej prostorného RD 5+1 v Týnci. *Reality, byty, domy v oblasti Brno, Břeclav, Hodnonín: Schmidt Reality s.r.o.* [online]. ©2017 [cit. 2017-01-16]. Dostupné z: <http://www.schmidt-reality.cz/reality/prodej-prostorneho-rd-5-1-v-tyn-ci-1776>
- [31] SEZNAM.CZ, A.S. *Sreality.cz: reality a nemovitosti z celé ČR* [online]. ©1996-2017 [cit. 2017-05-05]. Dostupné z: <https://www.sreality.cz/>
- [32] *Selenium: Web Browser Automation* [online]. [cit. 2017-05-05]. Dostupné z: <http://www.seleniumhq.org/>
- [33] HIDAYAT, Ariya. *PhantomJS* [online]. ©2010-2017 [cit. 2017-05-05]. Dostupné z: <http://phantomjs.org/>
- [34] HLAVINKA, Jaroslav. Jak se indexuje JavaScriptová Single Page Aplikace Zboží.cz: AngularJS a SEO. *Jak dělat SEO* [online]. 2017 [cit. 2017-05-05]. Dostupné z: <https://jakdelatseo.cz/zbozi-angularjs-seo/>
- [35] GOOGLE INC. *Google Maps APIs* [online]. [cit. 2017-05-06]. Dostupné z: <https://developers.google.com/maps/>
- [36] *SQLite Home Page* [online]. [cit. 2017-05-06]. Dostupné z: <https://www.sqlite.org/>
- [37] MNMTANISH. SQLite dialect for Hibernate. *The world's leading software development platform: GitHub* [online]. 2012 [cit. 2017-05-07]. Dostupné z: <https://gist.github.com/mnmtanish/4005758>
- [38] THE MATHWORKS, INC. *MATLAB Documentation* [online]. ©1994-2017 [cit. 2017-05-07]. Dostupné z: <https://www.mathworks.com/help/>

SEZNAM OBRÁZKŮ

Obrázek 2.1: Ukázka pluginu pro vývojové prostředí Eclipse	14
Obrázek 2.2: Role frameworku Hibernate v java aplikaci	16
Obrázek 2.3: Grafické znázornění operací mezi množinami	18
Obrázek 2.4: Demonstrace hranice mezi pravdivostí výroků na příkladu vyhodnocení teploty vody	19
Obrázek 2.5: Průběh a definice L -funkce	20
Obrázek 2.6: Průběh a definice Γ -funkce	20
Obrázek 2.7: Průběh a definice Λ -funkce	21
Obrázek 2.8: Průběh a definice Π -funkce	21
Obrázek 2.9: Funkce členství μ_A a μ_B	22
Obrázek 2.10: Grafické znázornění operací doplňku (nahore vlevo), průniku (nahore vpravo) a sjednocení (dole)	22
Obrázek 2.11: Postup při tvorbě systému s fuzzy logikou	23
Obrázek 4.1: Schéma návrhu aplikace	37
Obrázek 4.2: Porovnání některých atributů z dvou různých webů	38
Obrázek 4.3: Zachycení požadavku a odpovědi pomocí nástrojů vývojáře	41
Obrázek 4.4: Odpověď na požadavek ve formátu JSON – Seznam inzerátů	43
Obrázek 4.5: Odpověď na požadavek ve formátu JSON – Informace o konkrétní nemovitosti	45
Obrázek 4.6: Odpověď na požadavek ve formátu JSON – Vzdálenost objektu od sídla firmy	47
Obrázek 4.7: Entito-relační diagram	48
Obrázek 4.8: Schéma fuzzy modelu pro hodnocení domů	54
Obrázek 4.9: FIS Editor – Vytvoření bloku C	55
Obrázek 4.10: FIS Editor – Funkce členství proměnné energetická náročnost	56
Obrázek 4.11: FIS Editor – Funkce členství proměnné plocha pozemka	57
Obrázek 4.12: FIS Editor – Výstup bloku C	58
Obrázek 4.13: Fuzzy Logic Toolbox – Editor pravidel	59
Obrázek 4.14: FIS soubor – Způsob zápisu pravidel	59

Obrázek 4.15: Ad hoc řešení generátoru pravidel pro dva vstupy	60
Obrázek 4.16: Ruler Viewer – Testování modelu	61
Obrázek 4.17: Surface Viewer	61
Obrázek 4.18: Schéma fuzzy modelu pro hodnocení bytů	63
Obrázek 4.19: Ukázka části kódu bloku pro vyhodnocení celého objektu v jazyku FCL	64
Obrázek 4.20. Ukázka části kódu bloku pro vyhodnocení celého objektu v souboru FIS	65
Obrázek 4.21: Porovnání zápisu pravidel	66
Obrázek 4.22: Vizualizace výsledku a funkcí příslušnosti	66
Obrázek 4.23: Hotová aplikace – Aktivní karta s domy	67
Obrázek 4.24: Hotová aplikace – Barevné odlišení záznamů	68
Obrázek 4.25: Hotová aplikace – Řazení podle více sloupců	69

SEZNAM TABULEK

Tabulka 3.1: Souhrn zhodnocujících a nezhodnocujících investic	27
---	-----------

SEZNAM GRAFŮ

Graf 3.1: Vývoj průměrných úrokových sazeb	30
Graf 3.2: Vývoj cen bytů v ČR (bez hlavního města Prahy)	31

SEZNAM POUŽITÝCH ZKRATEK

API	Application Programming Interface
AWT	Abstract Window Toolkit
CSS	Cascading Style Sheets
DOM	Document Object Model
GUI	Graphical User Interface
HTML	HyperText Markup Language
JDK	Java Development Kit
JPA	Java Persistence API
JRE	Java Runtime Environment
JSON	JavaScript Object Notation
ORM	Object-Relational Mapping
POJO	Plain Old Java Object
SQL	Structured Query Language
URL	Uniform Resource Locator
XML	Extensible Markup Language