

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra počítačů

## Aplikace pro podporu hodnocení závěrečných prací

**Viktoriia Havrylenko**

Vedoucí: Ing. Pavel Náplava, Ph.D.  
Obor: Softwarové inženýrství a technologie  
Srpen 2022



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Havrylenko** Jméno: **Viktoriia** Osobní číslo: **492202**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávací katedra/ústav: **Katedra počítačů**  
Studijní program: **Softwarové inženýrství a technologie**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Aplikace pro podporu hodnocení závěrečných prací**

Název bakalářské práce anglicky:

**Application for support of a final thesis evaluation**

Pokyny pro vypracování:

Provedte analýzu způsobů hodnocení závěrečných prací jak ze strany vedoucích/oponentů, tak i studentů. Následně navrhnete a implementujete aplikaci, která bude pomáhat s hodnocením závěrečných prací. Postupujte následovně:

- 1) Seznamte se s existujícími doporučeními/formuláři pro posudky závěrečných prací.
- 2) Analyzujte existující doporučení pro hodnocení závěrečných prací v rámci ČVUT a dalších univerzit.
- 3) Provedte průzkum mezi vedoucími a oponenty, případně studenty, jakým způsobem provádějí hodnocení závěrečných prací, a zda jim stávající způsob vyhovuje.
- 4) Na základě průzkumu sestavte sadu kritérií a vah, které mohou posloužit jako vodítko pro tvorbu posudků závěrečných prací a studentům pomohou pochopit, jakým způsobem je jejich práce hodnocená.
- 5) Na základě kritérií a vah vytvořte aplikaci, kterou bude možné pro účely podpory hodnocení závěrečných prací využít.
- 6) Použitelnost aplikace ověřte pomocí uživatelského testování na skupině uživatelů, zapojených do průzkumu.

Seznam doporučené literatury:

1. VOJTÍŠEK, Petr. Výzkumné metody. 2012. online: [skoly.praha.eu/files/=84121/Skripta%20V%C3%BDzkumn%C3%A9\\_metody.pdf](http://skoly.praha.eu/files/=84121/Skripta%20V%C3%BDzkumn%C3%A9_metody.pdf). Vyšší odborná škola sociálně právní, Praha.
2. Descriptions of Grades and Assessment Criteria for Bachelor's Theses in Engineering, online: [www.uhr.no/\\_f/p1/i2803bcbe-1f46-4822-8a70-f0e52f5a6a8f/eng\\_grade\\_descriptions\\_and\\_assessor\\_information\\_bachelorthesis\\_uhr\\_logo.pdf](http://www.uhr.no/_f/p1/i2803bcbe-1f46-4822-8a70-f0e52f5a6a8f/eng_grade_descriptions_and_assessor_information_bachelorthesis_uhr_logo.pdf)
3. MASTER'S THESIS EVALUATION. Leuven, online: [iiv.kuleuven.be/english/students/master-thesis/evaluation](http://iiv.kuleuven.be/english/students/master-thesis/evaluation)

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Pavel Náplava, Ph.D. Centrum znalostního managementu FEL**

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

Datum zadání bakalářské práce: **02.02.2022**

Termín odevzdání bakalářské práce: **15.08.2022**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. Pavel Náplava, Ph.D.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

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

### III. PŘEVZETÍ ZADÁNÍ

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

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studentky

## Poděkování

Ráda bych poděkovala vedoucímu své bakalářské práce panu doktorovi Pavlu Náplavovi za cenné rady, užitečné konzultace a všechny čas. Děkuji všem respondentům, studentům, vyučujícím a externistům, kteří věnovali svůj čas pohovorům a neocenitelně přispěli k rozvoji projektu.

## Prohlášení

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

V Praze 15. května 2022

## Abstrakt

Tato práce se zabývá analýzou způsobů a přístupů k provedení hodnocení závěrečných prací studentů bakalářského a magisterského studia různých univerzit. Na vytvořené analýze je postaven návrh a implementace první verze aplikace pro pomoc s provedením hodnocení závěrečných prací bakalářského studia ČVUT FEL SIT. První část práce je zaměřena na výběr metod pro řešení dané problematiky. Další části jsou rozděleny do fází na základě odvedené práce. Tyto fáze zahrnují rešerši existujících metrik a způsobů hodnocení prací v rámci ČVUT i jiných univerzit; provedení pohovorů se studenty, vedoucími, oponenty a externisty pro zjištění problémů a návrhů na zlepšení existujícího přístupu k provedení a hodnocení závěrečných projektů. Na základě toho byl vytvořen první návrh dvou prototypů modelu hodnocení, které se následně upřesnily po provedení druhého kola pohovorů s respondenty. Navržené přístupy k hodnocení a odvedená analytická práce byly dále použity jako základ pro navazující aplikační část, kde byl vytvořen a implementován návrh jednoduché aplikace. Výsledky byly podrobeny uživatelskému testování a interpretovány v závěru.

**Klíčová slova:** model hodnocení, metriky hodnocení, závěrečná práce, kvalitativní výzkum, aplikovaný výzkum, webová aplikace

**Vedoucí:** Ing. Pavel Náplava, Ph.D.  
ČVUT FEL,  
Technická 2,  
166 27 Praha 6, Dejvice

## Abstract

This thesis deal with the analysis of the methods and approaches to the evaluation of final theses of Bachelor and Master studies of different universities. The design and implementation of the first version of an application to assist in the evaluation process of the final theses of the bachelor students of CTU FEL SIT is based on this analysis. The first part of the thesis is focused on the selection of the methods for solving the given problem. The next parts are divided into relevant phases based on the work done. These phases include a search of existing metrics and ways of evaluating theses both within CTU and other universities; conducting interviews with students, supervisors, opponents and external experts to identify problems and suggestions for improving the existing approach to conducting and evaluating final projects. Based on this, a first version of two prototypes of the evaluation model was developed, which were subsequently refined after a second round of interviews with respondents. The proposed approaches to assessment and the analytical work done were further used as a basis for the application part, where a simple application design was created and implemented. The results were subjected to user testing and interpreted in the final section.

**Keywords:** evaluation model, evaluation metrics, final thesis, qualitative research, applied research, web application

**Title translation:** Application for support of a final thesis evaluation

# Obsah

<b>Zadání práce</b>	<b>iii</b>	
<b>Úvod</b>	<b>1</b>	
Motivace .....	1	
Cíle práce .....	1	
Struktura práce .....	1	
<b>Část I</b>		
<b>Výzkumná část práce</b>		
<b>1 Klíčové pojmy</b>	<b>5</b>	
<b>2 Výzkum jako zvolený přístup řešení dané problematiky</b>	<b>7</b>	
2.1 Definice výzkumu .....	7	
2.2 Kategorizace výzkumu .....	8	
2.2.1 Základní výzkum .....	8	
2.2.2 Aplikovaný výzkum .....	9	
2.2.3 Porovnání typů výzkumů .....	9	
2.3 Výzkumné metody .....	10	
2.3.1 Kvantitativní výzkum .....	10	
2.3.2 Kvalitativní výzkum .....	11	
2.3.3 Porovnání výzkumných metod .....	12	
2.4 Výzkum použitý v bakalářské práci .....	13	
<b>3 Projekt výzkumu</b>	<b>15</b>	
3.1 Použité fáze výzkumu .....	15	
3.2 Formulace výzkumného problému .....	15	
3.3 Cíle výzkumu .....	17	
3.4 Přínosy výzkumu .....	18	
3.5 Časový harmonogram výzkumu .....	18	
3.6 Popis vybraných metod sběru a analýzy dat .....	20	
3.7 Popis skupin respondentů .....	21	
<b>4 Rešerše existujících metrik a modelů hodnocení</b>	<b>23</b>	
4.1 Obecné informace o zkoumaných modelech .....	23	
4.2 Vybrané modely hodnocení bakalářských závěrečných prací v ČR .....	24	
4.3 Vybrané modely hodnocení bakalářských závěrečných prací v technicky zaměřených oborech mimo ČR .....	25	
4.4 Vybraný model hodnocení bakalářských závěrečných prací v dalších oborech mimo ČR .....	27	
4.5 Vybrané modely hodnocení diplomových závěrečných prací ...	29	
4.6 Vyhodnocení rešerše .....	30	
<b>5 Vliv typu práce na její hodnocení</b>	<b>33</b>	
5.1 Důležitost zaměření práce při hodnocení .....	33	
5.2 Dělení závěrečných prací podle zaměření .....	34	
<b>6 Vnímání stávajícího přístupu k hodnocení absolventy</b>	<b>37</b>	
6.1 Shrnutí připomínek absolventů ..	37	
<b>7 Návrh prototypů modelu hodnocení</b>	<b>39</b>	
7.1 Kostra navržených prototypů ...	40	
7.2 Hlavní rozdíly prototypů .....	41	
<b>8 Vyhodnocení navržených prototypů hodnotiteli</b>	<b>43</b>	
8.1 Připomínky k navrženým prototypům .....	43	
<b>9 Finalizace přístupů k provedení hodnocení</b>	<b>45</b>	
9.1 Upřesnění obsahu modelu hodnocení .....	45	
9.2 Možné přístupy k ohodnocení dílčích částí modelu .....	46	
9.2.1 Bloky: zadání a splnění zadání .....	46	
9.2.2 Základní bloky .....	46	
9.2.3 Dílčí kritéria .....	46	
9.3 Možné kombinace stanovených přístupů .....	47	
9.4 Ověření vytvořených přístupů ..	47	
<b>Část II</b>		
<b>Aplikační část práce</b>		
<b>10 Návrh aplikace</b>	<b>51</b>	
10.1 Obecný popis požadavků na budoucí aplikaci .....	51	
10.2 Business cíle a business požadavky .....	52	
10.3 Analýza systémových požadavků .....	53	
10.4 Architektura .....	54	
10.5 Použité technologie .....	55	
10.5.1 Srovnání frameworků Vue, Angular, React .....	55	
10.5.2 Springboot framework .....	57	

10.5.3 PostgreSQL .....	57	<b>Závěr</b>	<b>87</b>
10.6 Datový model .....	57	<b>Literatura</b>	<b>89</b>
10.7 Sitemap .....	59		
10.8 Shrnutí návrhu aplikace .....	60	<b>Přílohy</b>	
<b>11 Implementace aplikace</b>	<b>63</b>	<b>A Diagramy z návrhové částí</b>	<b>95</b>
11.1 Backendová část .....	63	<b>B Elektronické přílohy práce</b>	<b>99</b>
11.1.1 Zakládání projektu .....	63		
11.1.2 Architektura Spring Boot ..	63		
11.1.3 Struktura projektu .....	64		
11.1.4 Vrstva přístupu k datům ...	64		
11.1.5 Servisní logika .....	66		
11.1.6 REST API, Kontroléry ....	67		
11.1.7 Autorizace .....	68		
11.2 Frontendová část .....	69		
11.2.1 Založení projektu .....	69		
11.2.2 Struktura projektu .....	69		
11.2.3 Zásadní principy práce s Reactem použité v aplikaci .....	70		
11.2.4 Použité knihovny .....	71		
11.2.5 Autorizace .....	72		
11.2.6 Seznam vytvořených obrazovek a návaznost na komponenty .....	74		
11.3 Komunikace backendu a frontendu .....	75		
<b>12 Uživatelské testování</b>	<b>77</b>		
12.1 Ověření aplikace a vytvořených přístupů .....	77		
12.2 Příprava k testování .....	78		
12.2.1 Návrh testovacího scénáře ..	78		
12.2.2 Popis vytvořených dotazníků pro zpětnou vazbu .....	79		
12.3 Průběh testování .....	79		
12.4 Výstupy testování .....	80		
12.4.1 Zpětná vazba od hodnotitelů	80		
12.4.2 Zpětná vazba od studentů ..	80		
<b>13 Vyhodnocení projektu výzkumu</b>	<b>83</b>		
13.1 Aktuální stav .....	83		
13.2 Interpretace výsledků .....	83		
13.3 Možnosti dalšího rozvoje projektu .....	85		



## Obrázky

2.1	Kategorizace výzkumu . . . . .	8
2.2	Fáze kvantitativního výzkumu . .	10
2.3	Fáze kvalitativního výzkumu . . .	12
3.1	Průběh projektu . . . . .	16
10.1	Architektura klient–server[23] .	56
10.2	Evidence uživatelů . . . . .	58
10.3	Evidence přístupů k hodnocení	59
10.4	Evidence dokončeného hodnocení	59
10.5	Diagram sitemap . . . . .	61
11.1	Architektura Spring Boot . . . . .	64
A.1	Business cíle a požadavky . . . . .	95
A.2	Systémové požadavky . . . . .	96
A.3	Datový model . . . . .	97



## Úvod

### Motivace

Na konci studia každého studenta na jakékoli vysoké škole nastává okamžik, kdy se poprvé setkává s tak důležitým krokem k dosažení vysokoškolského vzdělání, jako je psaní závěrečné práce. Vzhledem k tomu, že studenti bakalářského studia nemají téměř žádné zkušenosti s těmito typy prací, někteří z nich se ztratí hned na začátku. Studenti se proto během své nelehké práce neustále obracejí na oficiální dostupné zdroje svých univerzit nebo se ptají svých kamarádů a předchůdců, aby se dozvěděli více o tom, jak budou nakonec hodnoceni a jaké požadavky by měly být základem jejich práce.

Na začátku své práce se studenti snaží najít kritéria hodnocení, která předpokládají, že jsou v posudcích. Po zhlédnutí několika z nich studenti bohužel začnou být ještě nervóznější. Kritéria v nich uvedená a příslušné popisy jsou velmi abstraktní a nalezené příklady hodnocení hotových prací pokaždé mají jinou strukturu v rámci jednotlivých částí posudků.

Úkol, kterému čelí hodnotitelé, není o nic jednodušší. Vyučující mohou hodnotit desítky prací, přitom si musejí pamatovat detaily každé z nich, aby mohli ke všemu přihlídnout v posudcích. Hodnocení jednoho studenta může trvat hodiny. Studenti pak stále mají dotazy na finální verzi svého hodnocení. Nejčastější otázka ale zní: „Proč jsem byl hodnocen tak, a ne jinak?“ Motivací této práce je proces hodnocení zpřehlednit.

### Cíle práce

Tato práce se zabývá naplněním následujících cílů:

1. zjištění aktuálního stavu provedení a hodnocení závěrečných prací;
2. zvýšení povědomí studentů o požadavcích a průběhu práce;
3. zrychlení a zefektivnění procesu vyhodnocení prací.

---

## Struktura práce

Práce je rozdělena do čtyř základních částí:

1. Výzkumná část práce (viz I) je zaměřena na provedení náležité analytické práce tzn. výběr vhodné techniky pro naplnění cílů, provedení rešerše mezi dalšími univerzitami, hledání problémů aktuálního stavu hodnocení a návrh na zlepšení modelu hodnocení.
2. Aplikační část práce (viz II) se zabývá aplikací výsledků analytické části v praxi, a to prostřednictvím návrhu, implementace a testování první verze aplikace.
3. Třetí část obsahuje vyhodnocení úspěšnosti (viz 13) odvedené práce na základě analýzy naplnění cílů a předpokladů stanovených v průběhu práce.
4. Celá práce končí závěrem (viz 13.3).



## **Část I**

### **Výzkumná část práce**



# Kapitola 1

## Klíčové pojmy

V této kapitole jsou pomocí krátkého seznamu uvedeny hlavní termíny a pojmy, které v této práci často používám. Některé pojmy se mohou zdát podobné, ale používají se v různých kontextech a mají odlišný význam.

- **Posudek** – formulář, který používají vedoucí a oponenti pro hodnocení závěrečných prací studentů.
- **Model hodnocení** – struktura formuláře používaného pro hodnocení prací, přístup k hodnocení jeho dílčích částí a přístup ke spočítání výsledné známky (například dělení formuláře na části, které jsou ohodnoceny body od 0 do 100, výsledná známka je spočítaná jako průměr bodů za každou část).
- **Přístup k hodnocení** – popisuje, jak se hodnotí bloky neboli části modelu, kritéria v rámci bloku (pokud existují), výsledná známka.
- **Stávající model hodnocení** – model hodnocení, který se aktuálně používá pro ohodnocení prací pomocí posudků FEL. Tzn. rozdělení formuláře do šesti částí, kde první dvě jsou hodnoceny slovy, ostatní známku A až F, výsledná známka je stanovena hodnotitelem na základě jeho detailního odůvodnění (více v sekci 4.2.1).
- **Prototyp modelu hodnocení** – návrh úprav nebo rozšíření stávajícího modelu hodnocení, který vzešel z odvedené analytické práce. Tento návrh může být v dalším průběhu práce změněn nebo upřesněn, jestli to bude vyžadováno.
- **Prototyp aplikace** – první jednoduchá verze implementované aplikace, která vychází z vytvořeného prototypu modelu hodnocení. Nejedná se o finální řešení, ale návrh, který je využit pro vyhodnocení správnosti a smysluplnosti navrženého řešení.
- **Zaměření práce** – jedná se o oblast informačních technologií, v jejímž rámci byla práce realizována. Vychází ze zadání.
- **Úvaha** – definice myšlenkových pochodů a předpokladů, které vznikaly v průběhu plnění zadání práce. Úvahy analyzují a strukturují výstupy

1. Klíčové pojmy

---

obdržené v průběhu výzkumné části práce a jsou využity jako zadání pro aplikační část.



## Kapitola 2

### Výzkum jako zvolený přístup řešení dané problematiky

Po diskuzi s vedoucím práce jsme došli k závěru, že naplnění cílů této práce bude uchopeno formou výzkumu. Vzhledem k tomu tato kapitola popisuje klasifikaci výzkumu, možné metodiky a techniky provádění výzkumu, průběh a naplnění jednotlivých fází výzkumu. V poslední sekci kapitoly uvádím zvolený přístup pro řešení problematiky v této práci.

#### 2.1 Definice výzkumu

Organizace pro hospodářskou spolupráci a rozvoj<sup>1</sup> uvádí následující definici pojmu výzkum:

*"Výzkum a experimentální vývoj jsou tvořeny kreativní/tvůrčí a systematickou prací, vykonávanou za účelem zvýšení úrovně vědomostí včetně znalostí lidstva, kultury a společnosti a k navržení nových způsobů aplikace dostupných znalostí."* [1]

Jinými slovy lze říct, že výzkum je proces zkoumání definovaného problému s cílem objevování, přepracování nebo zahazování stanovených předpokladů za použití metod a technik, které naplnění tohoto cíle umožňují. Stejně tak se jedná o sběr dat, informací, jejich následnou analýzu prostřednictvím výzkumných metod. Záměr a úvahy stanovené na začátku výzkumu jsou ve výsledku potvrzené nebo vyvrácené na základě odvedené analytické práce.

V rámci rozpracování bakalářských a diplomových prací se obvykle rozumí příprava, sběr, zpracování a interpretace vlastních výzkumných dat, na které navazuje teoretické rozpracování problému, analýza, případná aplikace výsledků v praxi a vyhodnocení odvozených závěrů [2].

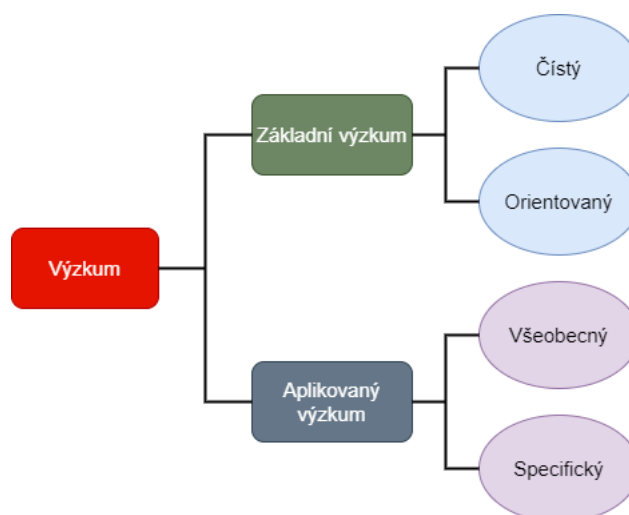
---

<sup>1</sup>Odkaz na stránky OECD: <https://www.oecd.org>

## 2.2 Kategorizace výzkumu

Výzkum lze rozdělit do několika kategorií podle účelu provádění výzkumné činnosti, podle metod a technik využitých pro naplnění těchto účelů, podle oblasti, v níž je výzkum prováděn, podle možné aplikace dosažených výsledků, podle míst, kde se provádí, atd.

Rozlišujeme základní a aplikovaný výzkum. Přehled dělení výzkumu na kategorie a podkategorie je uveden na obrázku č.2.1. Jedná se o základní dělení používané i ve skriptech s doporučeními k vypracování závěrečných prací studentů [3]. Některé zdroje přidávají i další kategorie k základnímu dělení. Například Rada pro výzkum, vývoj a inovace v České republice<sup>2</sup> na stejné úrovni se základním dělením uvádí experimentální výzkum [4]. Dalším příkladem může být dělení do akčního výzkumu, který se zaměřuje na bádání v oblasti sociálních věd a sledování organizačních problémů [5]. Dále se detailněji zaměřím pouze na základní a aplikovaný výzkum, jež na rozdíl od dalších typů mohou být využité pro řešení dané problematiky.



Obrázek 2.1: Kategorizace výzkumu

### 2.2.1 Základní výzkum

Jádrem základního výzkumu, stejně jako i aplikovaného, je experimentální nebo teoretická práce. Cílem základního výzkumu je objevit nové poznatky o jevech přírody, fenoménech prostředí nebo vysvětlit důvody jejich vzniku. Často je pro definici této kategorie výzkumu použito slovo „bádání“. Důvodem je to, že obvykle vědec provádějící základní výzkum je motivován výhradně svojí zvědavostí. Očekávaným výsledkem jeho práce je potvrzení nebo vyvrácení teorie definované v širším kontextu, která ve finále nemusí být použita v praxi. Interpretované výsledky základního výzkumu se předávají převážně

<sup>2</sup>Odkaz na stránky: <https://www.vyzkum.cz>

mezi akademickými pracovníky, a to prostřednictvím konferencí, vědeckých publikací nebo článků [5].

Základní výzkum lze rozdělit do dvou kategorií. První je čistý výzkum. Tento druh výzkumu se provádí bez snahy přispět v hospodářském nebo sociálním rozvoji. Druhý typ výzkumu – orientovaný výzkum – má za cíl poskytnout širokou základnu znalostí o problému, na nichž mohou být v budoucnu postavena další objevování [4].

### ■ 2.2.2 Aplikovaný výzkum

Aplikovaný výzkum se od základního liší tím, že získané poznatky jsou často ve výsledku použité v praxi. Rozsah práce je ze začátku jasně stanoven a je zaměřen na specifické oblasti použití. Náplní aplikovaného výzkumu, na rozdíl od základního, je nalezení metod a technik pro řešení přesně dané problematiky. Kvalita výsledků je měřitelná a obvykle je vyhodnocena organizacemi, které jsou ochotné za dodané výsledky zaplatit [5].

Aplikovaný výzkum se dělí na všeobecný a specifický. První je zaměřen na zkoumání problému pro definování jeho přesnějších specifikací. V rámci specifického výzkumu již existuje přesná specifikace cílů a výzkum se směřuje ke zkoumání metod pro jejich dosažení [4].

### ■ 2.2.3 Porovnání typů výzkumů

Přehledný seznam hlavních rozdílů mezi aplikovaným a základním výzkumem je uveden v tabulce č. 2.1. <sup>3</sup>

	Základní výzkum	Aplikovaný výzkum
Typ práce	Experimentální nebo teoretická	
Záměr	„Bádání“ v problému	Zaměření na určitý cíl
Motivace	Hlavně zvědavost	Řešení skutečného problému
Cíl	Rozšíření vědecké báze poznatků o jevu, fenoménu nebo vytvoření nové; zvýšení poznání lidstva o prostředí	Nalezení metod řešení problému pro následnou aplikaci v praxi
Dopad výzkumu	Potvrzená nebo vyvrácená teorie, nová myšlenka nebo vysvětlení	Nové poznatky o problému se specifickým zaměřením
Aplikace, využití	Nezabývá se praktickým využitím a předáním poznatku zainteresovaným stranám	Možná aplikace

**Tabulka 2.1:** Hlavní rozdíly základního a aplikovaného výzkumu

Pro studijní účely se doporučuje využívat aplikovaný výzkum, jelikož student má jasně definovaný problém, po důsledné analýze je schopen určit možnosti jeho řešení a ve výsledku vidí přínosy praktického využití [3].

## 2.3 Výzkumné metody

Pro oba typy výzkumů, které byly popsány v předchozí sekci, existují speciální výzkumné metody sběru a analýzy dat.

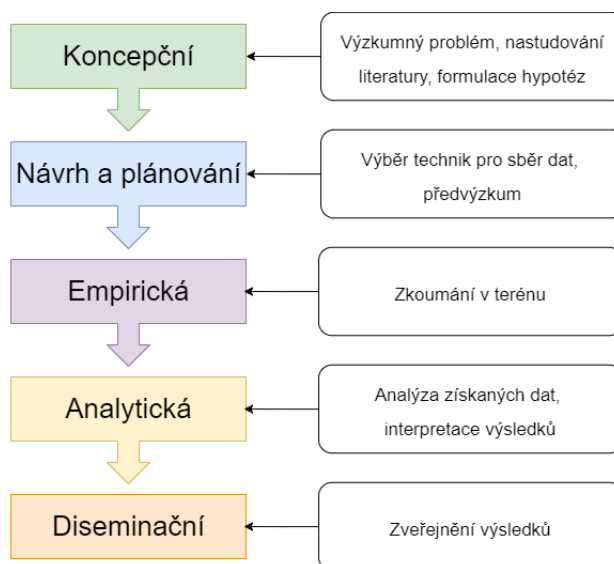
Klasifikace výzkumných metod do skupin se provádí z různých hledisek, například podle typu vědeckého postupu, způsobu vysvětlení zkoumaného problému, stupně obecnosti atd. Standardně se výzkumné metody klasifikují do dvou základních skupin: kvantitativní a kvalitativní.

### 2.3.1 Kvantitativní výzkum

Kvantitativní výzkum je založen na práci s číselnými nebo numerickými daty, které jsou podloženy analýzou a interpretovány pomocí statistických zákonů. V rámci kvalitativního výzkumu se stanoví hypotézy. Tyto hypotézy popisují vztahy mezi zkoumanými jevy. Pomocí deduktivní logiky jsou tyto vztahy ve výsledku buď zamítnuté, nebo přijaté pro další zkoumání [5].

#### Postup práce při kvantitativním výzkumu

Fáze, které do sebe zahrnuje kvantitativní výzkum, znázorňuje obrázek č. 2.2.



Obrázek 2.2: Fáze kvantitativního výzkumu

<sup>3</sup>Vytvořeno autorem

V rámci těchto fází výzkumník postupně prochází následujícími kroky: stanovení výzkumného problému, studování existujících zdrojů pro danou problematiku (koncepční fáze), vytváření výzkumného plánu (návrh a plánování), získávání dat o zkoumaném problému v terénu (empirická), analýza získaných údajů pomocí statistických metod (analytická) a zveřejnění výsledků (diseminační) [6].

### Metody sběru a analýzy dat kvantitativního výzkumu

Tabulka č. 2.2 uvádí stručný popis hlavních statistických metod využívaných v rámci empirické a analytické fáze kvantitativního výzkumu.

Metoda	Cíl	Výhody
<b>Korelační analýza</b>	potvrzení nebo vyvrácení existence vztahů mezi dvěma nebo více náhodnými veličinami	korelační koeficienty se poměrně snadno počítají, jejich aplikace nevyžaduje zvláštní matematické znalosti
<b>Lineární regresní analýza</b>	zkoumání existujícího vztahu mezi dvěma nebo více proměnnými, jeho vyhodnocení a analýza	pevné matematické základy
<b>Anketa</b>	zjišťování názorů u vybraných skupin lidí na určité téma, shromažďování informací	dostupnost, univerzálnost, relativně rychlá příprava
<b>Standardizovaný strukturovaný rozhovor</b>	zjišťování názorů u vybraných skupin lidí na určité téma prováděné prezenčně	možnost odstoupení o struktury dotazníku a přizpůsobit se respondentovi
<b>Přehledová studia</b>	shrnutí aktuálního stavu výzkumu, review odvedené práce	soustředění rozptýlené informace na jednom místě
<b>Dotazníkové šetření</b>	strukturovaný rozhovor provedený písemně	pokrytí relativně velkého vzorku respondentů
<b>Experimenty</b>	sledování jevu za nasimulovaných podmínek	možnost systematicky měnit podmínky průchodu

Tabulka 2.2: Metody využívané při kvantitativním výzkumu[3]

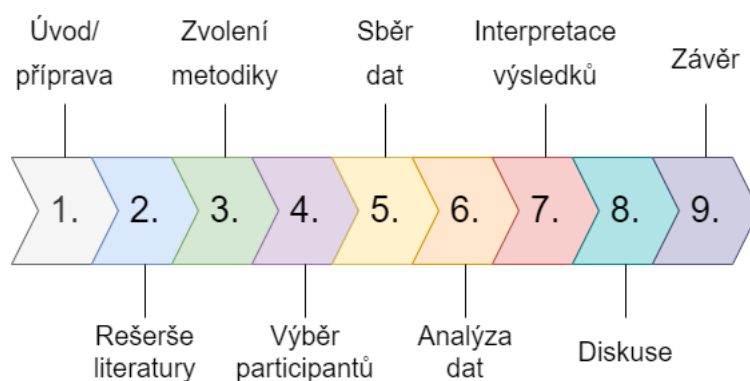
### 2.3.2 Kvalitativní výzkum

Kvalitativní výzkum se zaměřuje na vyprávění, popis, interpretaci a kontext. Cílem kvalitativního výzkumu je popsat, interpretovat a pochopit jev, který

nás zajímá, a to v kontextu, v němž se vyskytuje. Kvalitativní údaje se obvykle sbírají pozorováním nebo provedením rozhovorů a zaznamenávají se ve formě vyprávění. Výsledky z pozorování jsou zkoumány s cílem vytvořit společné příznaky a vzorce [5].

### ■ Postup práce při kvalitativním výzkumu

Postup provádění kvalitativního výzkumu je podobný kvantitativnímu výzkumu (viz obrázek č. 2.3). Začíná se úvodem, literární rešerší, popisem metodik a respondentů pro návaznou fázi sběru a analýzu dat, výsledky jsou interpretovány a zveřejněny. Hlavním rozdílem je použití jiných metod získávání, analýzy a následného vyhodnocení dat [7].



Obrázek 2.3: Fáze kvalitativního výzkumu

### ■ Metody pro sběr i analýzu dat kvalitativního výzkumu

Hlavní techniky využívané v rámci kvalitativního výzkumu, jejich cíle a výhody popisuje tabulka 2.3.

#### ■ 2.3.3 Porovnání výzkumných metod

Základní rozdíly těchto dvou výzkumných metod hlavně spočívají v tom, že zatímco kvantitativní výzkum klade větší důraz na zpracování měřitelných faktů za využití poměrně velkého počtu vzorků pomocí statistických metod, kvalitativní výzkum zkoumá problémy a sociální realitu kolem nich na poměrně malém počtu respondentů, zaznamenává a interpretuje kontext, který dává půdu pro tvoření hypotéz a úvah. Za zmínku stojí to, že hypotézy kvantitativního výzkumu jsou tvořeny ihned na začátku výzkumu. Pak následuje jejich vysvětlení. Při kvalitativním přístupu fakty přibývající v průběhu projektu stanoví další úvahy. Výstupy kvalitativního přístupu jsou zcela subjektivní (studují problém v určitém kontextu) a oproti kvantitativnímu nezajišťují standardní statistický vzorec k replikaci postupu.

Metoda	Cíl	Výhody
<b>Hlubkové rozhovory</b>	shromáždění velkého množství informací o chování, postojích a vnímání dotazovaných	účastníci mají možnost prozkoumat další body problému a v případě potřeby změnit směr pohovoru
<b>Cílové skupiny</b>	zkoumání lidského chování a názoru pomocí provedení skupinových diskuzí na zvolené téma	participanti nejsou omezení předem stanovenými otázkami
<b>Zúčastněné pozorování</b>	dlouhodobé pozorování chování a interakcí participantů v přirozených situacích	větší prostor pro zkoumání jevu
<b>Případové studie</b>	komplexní zkoumání jevu z hlediska vztahu a souvislosti pro stanovení faktorů, které tento jev ovlivňují	komplexnost, rozsáhlost
<b>Sekundární analýza dat</b>	snaha o objevení nových předpokladů v již zpracovaných dokumentech	hlubší analýza dat

Tabulka 2.3: Metody využívané při kvantitativním výzkumu [3]

## 2.4 Výzkum použitý v bakalářské práci

Řešení problému této práce je bráno jako **výzkumný problém**. Podle teoretických základů uvedených v kapitole je vidět, že se jedná o **aplikovaný výzkum**. Závěry a úvahy definované v průběhu tohoto projektu (viz I. Výzkumná část práce) jsou implementované a ověřené v kontextu návazné aplikační části (viz II).

Téma této práce je specifické, a to především z důvodu omezenosti na systém konkrétní vzdělávací instituce. V daném případě se jedná o elektrotechnickou fakultu ČVUT. To znamená, že přístup k výzkumu vyžaduje individuální řešení pro realizaci požadavků budoucích uživatelů a následného posouzení správnosti jejich aplikace. Vzhledem k tomu mohou být některé fáze použitých metod změněné nebo mohou obsahovat techniky napříč dvěma metodami. Tento přístup se v odborné řeči obvykle nazývá **triangulace**. Triangulací se rozumí kombinace různých datových zdrojů, metodik, teorií a technik v rámci zkoumání jednoho jevu nebo problémů [8]. Za zmínku však stojí to, že všechny změny práce byly pečlivě projednány s vedoucím a jím potvrzeny.

Kostrou zvoleného přístupu jsou **kvalitativní metody**, protože pracují s nestatistickými daty, provádějí analýzy „do hloubky“ a zkoumají problém za účasti cílových skupin a respondentů, kteří jsou prioritně vázani s proble-





## Kapitola 3

### Projekt výzkumu

V předchozí kapitole byly analyzovány možné přístupy provedení výzkumu a zvolen ten, který se nejlépe hodí pro tuto práci (viz kapitola 2.4). Tato kapitola se zabývá definicí projektu výzkumu této bakalářské práce na základě zvolené metody. Na začátku se uvádí stručné seznámení s průběhem výzkumu, které je pak detailně popsáno v sekci 3.5. Následně se definuje výzkumný problém, cíle, přínosy a techniky jejich naplnění.

#### 3.1 Použité fáze výzkumu

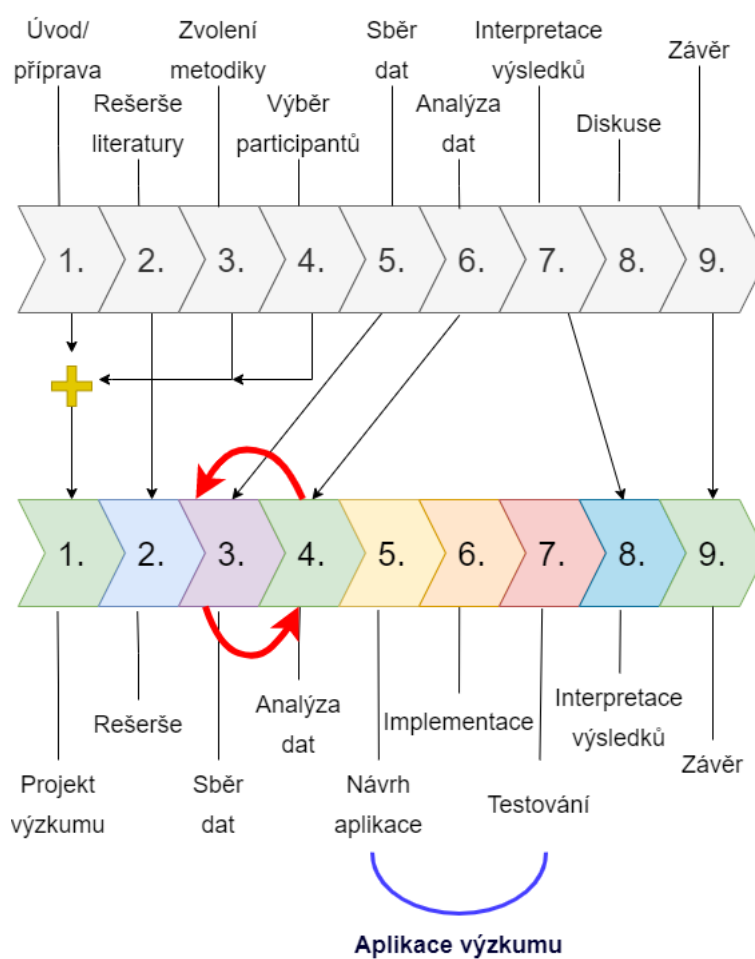
Struktura práce je založena na kvalitativním aplikačním výzkumu. Po diskuzi s vedoucím práce byly fáze kvalitativní metody upraveny tak, aby průběh výzkumu co nejlépe vyhovoval pro naplnění cílů této práce. Všechny změny jsou uvedené na obrázku č. 3.1 níže. Horní část diagramu ukazuje standardní průběh kvalitativního výzkumu shodný s obrázkem č. 2.3 z předchozí kapitoly. Dolní diagram znázorňuje změny, které byly do průběhu výzkumu zapracované.

Naplnění této kapitoly odpovídá první fázi výzkumu a kombinuje tři fáze standardního průběhu. Následuje fáze rešerše zdrojů. Z obrázku je zřejmé, že další fáze, tj. sběr a analýza dat, se mohou opakovat v závislosti na tom, kolik toho bude řešení problému vyžadovat. Poté následují tři fáze aplikace analyzovaných dat v praxi ve formě návrhu, implementace a testování aplikace. Výsledky jsou dále interpretovány v osmé fázi. Další fáze uzavírá a vyhodnotí úspěšnost výzkumu. Poslední fází je závěr projektu. Obsah jednotlivých fází je podrobněji popsán v sekci č. 3.5.

#### 3.2 Formulace výzkumného problému

Práce se po celou dobu výzkumu bude zabývat následujícím problémem:

*"Zmapování existujících přístupů a s tím spojených problémů při vyhodnocení závěrečných prací bakalářského studia ČVUT FEL v programu SIT. Implementace vhodného řešení pro použití v praxi."*



Obrázek 3.1: Průběh projektu

Pro lepší pochopení záměrů výzkumu a znění problému definujeme rozšiřující otázky, na které bude zodpovězeno v průběhu práce:

1. Jak přistupují k hodnocení další univerzity?
2. S jakými problémy se studenti a učitelé potýkají v průběhu naplnění a vyhodnocení závěrečných prací?
3. Jak studenti a učitelé vnímají aktuálně používaný přístup hodnocení na univerzitě?
4. Lze vytvořit nebo rozšířit stávající model použitý v posudcích tak, aby vyhovoval různorodým požadavkům uživatelů a posloužil jako pohodlný nástroj pro definování zadání a finálního vyhodnocení?
5. Pokud řešení existuje, jak navrhnout jeho implementaci pro použití v praxi?

## 3.3 Cíle výzkumu

Jak je patrné ze zadání a definice problému, hlavním záměrem této práce je nalezení vhodného řešení pro hodnocení závěrečných prací mezi existujícími přístupy; zkombinovat je či rozšířit takovým způsobem, aby to uživatelům poskytlo pohodlný nástroj při jejich práci. V návrhu je třeba zohlednit aktuální problémy a požadavky hodnotitelů a studentů spojené s průběhem a vyhodnocením závěrečných projektů. Vzhledem k tomu byly stanoveny hlavní cíle výzkumu vyplývající ze zadání a dílčí výzkumné cíle související s realizací výzkumu (viz níže).

### Hlavní a dílčí cíle výzkumu

#### ■ Cíl1: Návrh rozšíření modelu

Navrhnout rozšíření stávajícího modelu a kritéria pro hodnocení závěrečných prací studentů s přihlédnutím k osvědčeným přístupům jiných univerzit a požadavkům vybraných uživatelů.

##### ■ Dcíl1: Analýza existujících modelů hodnocení

Provést analýzu existujících řešení při hodnocení závěrečných (bakalářských i magisterských) prací studentů různých oborů vysokých škol. Najít rozdíly, podobnosti či vzorce v přístupech a určit taková řešení, která by bylo možné využít pro následný vývoj aplikace.

##### ■ Dcíl2: Definice problémů při hodnocení

Určit problémy, se kterými se studenti a hodnotitelé potýkají při plnění zadání závěrečných prací a jejich hodnocení.

##### ■ Dcíl3: Nalezení souvislostí v modelu

Najít vzorce/vztahy mezi kritérii hodnocení a jejich dopadem na celkový výsledek hodnocení.

##### ■ Dcíl4: Finalizace modelu

Vytvořit finální verzi modelu hodnocení, která bude implementována v aplikaci.

#### ■ Cíl2: Implementace první verze aplikace

Navrhnout a následně implementovat prototyp modelu hodnocení v první jednoduché verzi aplikace.

##### ■ Dcíl5: Výběr technologií

Krátce porovnat existující technologie a nástroje návrhu a vývoje webových aplikací, zvolit nejpraktičtější řešení pro danou problematiku.

##### ■ Dcíl6: Návrh aplikace

Vytvořit potřebné specifikace aplikace, na kterých bude založena implementace první verze aplikace.

##### ■ Dcíl7: Implementace

Vytvořit první funkční verzi nástroje pro hodnocení závěrečných prací.



#	Kroky	Termín	Obsah
1.	Projekt výzkumu	září 2021	Úvod, příprava projektů výzkumu, definice výzkumného problému, stanovení cílů, sestavení harmonogramu, výběr metod a technik pro řešení problému, výběr respondentů (náplň této kapitoly).
2.	Rešerše	říjen 2021	Analýza existujících metod hodnocení závěrečných prací různých univerzit včetně aktuálně používaného hodnocení na katedře počítačů FEL. Vyhodnocení modelů, vyčlenění technik, které budou spojené do jednotného modelu a ukázané vyučujícím v dalších fázích. Definování prvních úvah a požadavků, které je třeba zohlednit při návrhu modelu (viz kapitola 4). Naplněný Dcíl1: Analýza existujících modelů hodnocení.
3.	Zpětná vazba k hodnocení od absolventů	začátek listopadu 2021	První kolo sběru dat. Provedení pohovorů s absolventy ČVUT FEL. Definování problémů, se kterými se studenti pravidelně potýkají v průběhu náplně projektů. Rozšíření seznamu úvah (viz kapitola 6). Naplněný Dcíl2: Definice problémů při hodnocení.
4.	Návrh rozšíření modelu hodnocení	konec listopadu 2021	První kolo analýzy dat. Na základě rešerše existujících přístupů k hodnocení a pohovorů se studenty bude vytvořen první návrh rozpadů kritérií. Bude proveden pokus o vytvoření prvního rozšíření a upřesnění stávajícího modelu (viz kapitola 7). Naplněný Dcíl3: Nalezení souvislostí v modelu.
5.	Zpětná vazba k hodnocení od hodnotitelů	začátek prosince 2021	Druhé kolo sběru dat. Provedení pohovorů s hodnotiteli, kde jim bude ukázán návrh z předchozí fáze a budou posbírané názory a návrhy na zlepšení modelu (viz kapitola 8). Naplněný Dcíl2: Definice problémů při hodnocení.

6.	Finalizace rozšíření modelu	konec prosince až začátek ledna 2022	Druhé kolo analýzy dat. Navržení rozsáhlejšího modelu hodnocení s přihlédnutím k požadavkům a přístupům vyučujících (viz kapitola 9). Naplněný Dcíl3: Nalezení souvislostí v modelu a Dcíl4: Finalizace modelu.
7.	Ohodnocení návrhu hodnotiteli	únor až březen 2022	Třetí kolo sběru dat. Provedení dalších pohovorů s hodnotiteli, v rámci kterých budou navržené kombinace přístupů aplikované na již vyhodnocené práce studentů s cílem definovat nejpraktičtější řešení, které bude dále implementováno (viz sekce 9.4). Naplněný Dcíl4: Finalizace modelu.
8.	Návrh aplikace	duben 2022	Volba technologií, soupis specifikací (viz kapitola 10). Naplněný Dcíl5: Výběr technologií a Dcíl6: Návrh aplikace.
9.	Implementace	květen až červen 2022	Postupná implementace první verze aplikace na základě vznikajících technických a funkčních požadavků na systém (viz kapitola 11). Naplněný Dcíl7: Implementace.
10.	Testování	konec června 2022	Příprava testovacích scénářů, dotazníků. Předložení vytvořené aplikace a implementovaného řešení ke zpětné vazbě od klíčových uživatelů prostřednictvím uživatelského testování a odpovědí na dotazníky (viz kapitola 12). Naplněný Dcíl8: Testování.
11.	Vyhodnocení projektu výzkumu	červenec 2022	Vyhodnocení výstupů fáze testování, vyhodnocení stupně naplnění cílů a návržení budoucího rozvoje projektu (viz kapitola 13). Naplněný Dcíl9: Odvozování závěrů.

**Tabulka 3.1:** Harmonogram a rozdělení práce během výzkumu

### 3.6 Popis vybraných metod sběru a analýzy dat

Dále je uveden seznam metod, které byly zvolené pro použití během výzkumu a jejich praktické uplatnění v projektu:

- Kombinace standardizovaného strukturovaného rozhovoru kvantitativní metody a hloubkových rozhovorů kvalitativní metody.

Standardizované rozhovory zajistí odpovědi respondentů na předpřipravené otázky. Ty ale mohou být následně rozšířené a přizpůsobené jednotlivým

rozhovorům pro získání detailnějších názorů na výzkumný problém. Tento přístup je použit v několika fázích projektu: během pohovorů s absolventy a hodnotiteli. Využití této metody pomohlo naplnit Dcíl2: Definice problémů při hodnocení.

- Případové studie

Metoda je použita pro definování úvah týkajících se rozpadu modelu hodnocení a hledání souvislostí napříč modely hodnocení. Výsledky pomohly co nejlíp přizpůsobit navržený model jednotlivým přístupům hodnotitelů. Úvahy byly následně analyzovány a výsledky byly interpretovány pro potvrzení i vyvrácení stanovených předpokladů. Využití této metody pomohlo naplnit Dcíl3: Nalezení souvislostí v modelu a Dcíl4: Finalizace modelu.

- Sekundární analýza dat

Tato metoda se především používá pro analýzu existujících modelů hodnocení s cílem vyhodnocení přístupu a jejich využití v budoucím návrhu. Využití této metody pomohlo naplnit Dcíl1: Analýza existujících modelů hodnocení.

- Dotazníkové šetření

Dotazníkem jsou nahrazené rozhovory ve finální fázi výzkumu při testování aplikace na vybraných uživateli. Samotný dotazník a výsledky šetření jsou přiložené jako elektronická příloha práce B (viz soubory *dotaznik\_hodnotitele* a *dotaznik\_studenti*). Využití této metody pomohlo naplnit Dcíl8: Testování a Dcíl9: Odvozování závěrů.

## 3.7 Popis skupin respondentů

Vzhledem k tomu, že výzkum a jeho výsledky jsou vytvářeny převážně pro Fakultu elektrotechnickou, a budou sloužit k hodnocení prací programu SIT, je výběr respondentů omezen především na zmíněnou fakultu. Pro získání názoru nezávislých účastníků bylo také rozhodnuto o rozhovoru s jedním z pedagogů Fakulty informačních technologií. Mezi respondenty lze rozlišit dvě hlavní skupiny, na které byly rozhovory zaměřeny.

- Studenti a absolventi

Tato skupina zahrnuje studenty i absolventy fakulty. Výběr byl omezen dostupností studentů a jejich ochotou pomáhat při výzkumu. Počet respondentů byl 11 osob. První polovina se zúčastnila prvního kola pohovorů, zatímco ostatní se přidali do testování aplikace a vyplnění dotazníků. Rozhovory byly vedeny za účelem získání informací o tom, jak studenti vnímají dosavadní proces vedení a vyhodnocení práce, s jakými problémy se potýkají při tvorbě projektů a jestli souhlasí s navrženým řešením.

- Zaměstnanci univerzity

Do této kategorie patří **vedoucí prací, oponenti i další vyučující**. Další vyučující přímo nesouviseli s vedením závěrečných prací, ale měli zajímavé nápady a osobní názory na tuto problematiku (např. **členové komisí, garanti programů**). Počet respondentů, kteří se zúčastnili pohovorů a testování, byl 10 osob.

V této kapitole byly definovány tři hlavní a devět dílčích cílů projektu vyplývající z výzkumného problému. Jejich naplněním jsem se zabývala v rámci jedenácti popsaných kroků projektu (viz 3.1) za použití vybraných výzkumných metod (viz 3.6). Na konci kapitoly byly uvedeny dvě skupiny respondentů (studenti, absolventi a zaměstnanci univerzity), kteří se zúčastnili pohovorů pro sběr a analýzu dat.



## Kapitola 4

### Rešerše existujících metrik a modelů hodnocení

Tato kapitola se zabývá naplněním Dcíle1: Analýza existujících modelů hodnocení, který byl stanoven v rámci definice projektu výzkumu. Náplní kapitoly je seznámení s již existujícími modely hodnocení závěrečných prací studentů. První část kapitoly obsahuje analýzu posudků aktuálně používaných pro hodnocení studentů ČVUT v Praze (FEL, FIT) i VUT v Brně. V další části se hlavně zaměřím na rešerši modelu hodnocení používaných v univerzitách mimo Českou republiku.

Všechny dokumenty (posudky oponenta, vedoucího, další modely) se nachází v elektronické příloze práce (viz B soubor *zkoumane\_modely*).

#### 4.1 Obecné informace o zkoumaných modelech

Kapitola uvádí příklady modelů hodnocení (viz definice v kapitole 1), které se od sebe něčím liší. Ve většině případů univerzity nemají pořádně sepsané modely hodnocení závěrečných prací anebo nevystavují hotová hodnocení na svoje webové stránky pro veřejný přístup. Kvůli tomu je seznam univerzit k analýze zvolen na základě dostupnosti jejich informací. Stojí za zmínku říct, že tato překážka nemá vliv na kvalitu výsledné analýzy. Hlavním cílem této části nebyla analýza konkrétního výčtu světově známých univerzit, ale zkoumání co nejvíce možných a dostupných variant modelů hodnocení a do nich spadajících kritérií. Seznámení s variantami z různých oborů pomůže se podívat na vyřešení problému z jiného pohledu a rozšířit budoucí návrh o kritéria, na která předtím nebyl kladen žádný důraz. Kapitola kromě modelů hodnocení bakalářských prací navíc zkoumá modely magisterských prací, a to v různých oborech.

Většina modelů uvedených v práci obsahuje i hodnocení prezentací výstupů studenta, což pro tuto práci není relevantní. Vzhledem k tomu daná analýza nezahrnuje popis těchto kritérií a nezvažuje je při návrhu finálního modelu hodnocení.

## 4.2 Vybrané modely hodnocení bakalářských závěrečných prací v ČR

Účelem této části práce bylo zkoumání aktuálně používaných modelů hodnocení jak FEL, tak i dalších fakult a univerzit stejného zaměření. V průběhu analýzy se ukázalo, že všechny vybrané univerzity a fakulty používají víceméně podobný přístup. Vzhledem k tomu první příklad bude rozebrán podrobněji. U zbylých příkladů budou naznačeny pouze hlavní odlišnosti.

### 4.2.1 ČVUT v Praze, Fakulta elektrotechnická

Formuláře posudků se nachází v elektronické příloze B (viz soubory *oponent\_FEL* a *vedouci\_FEL*)

#### Přístup k hodnocení

Dokument je rozdělen do šesti bloků, z nichž čtyři jsou ohodnoceny známkou A až F v závislosti na rozhodnutí vedoucího (oponenta). Zbytek bloků se hodnotí slovně: např. u bloku Zadání se hodnotí jeho obtížnost pomocí popisného slova: od mimořádně náročného do nedostatečně náročného. Výsledná známka není aritmetickým průměrem všech známek za bloky, ale je stanovena vedoucím (oponentem) na základě jeho argumentace.

#### Sekce hodnocení:

- zadání (neboli jeho náročnost);
- splnění zadání;
- aktivita a samostatnost při zpracování práce (pro vedoucího) / zvolený postup řešení (pro oponenta);
- odborná úroveň;
- formální a jazyková úroveň, rozsah práce;
- výběr zdrojů, korektnost citace.

Každá z těchto sekcí má krátký popis uděleného hodnocení, k němuž mohou přihlídnout hodnotitelé při tvorbě posudků. Toto schéma se považuje za **stávající model** (viz kapitola 1), na který je odkazováno v průběhu výzkumu.

### 4.2.2 ČVUT v Praze, Fakulta informačních technologií

Výtahy posudků se nachází v elektronické příloze B (viz soubory *oponent\_FIT* a *vedouci\_FIT*)

**Přístup k hodnocení** je víceméně podobný předchozímu přístupu, hlavní rozdíly jsou však:

- absence hodnocení náročnosti zadání;
- hodnocení písemných i nepísemných výsledků projektu;
- do samostatné kategorie je zařazen bod vyhodnocení využitelnosti výsledků, jedná se o posouzení možností nasazení výsledků práce v praxi, může se uvádět, zda výsledky práce rozšiřují již publikované známé výsledky nebo přinášejí zcela nové poznatky (v předchozí metodě tento bod patřil do bloku Odborná úroveň);
- oponent nehodnotí zvolený postup řešení.

#### ■ 4.2.3 VUT v Brně, Fakulta informačních technologií

Výtahy posudků se nachází v elektronické příloze B (viz soubory *oponent\_VUT* a *vedouci\_VUT*)

**Přístup k hodnocení** obsahuje následující rozdíly:

- hodnotí se rozsah práce (pro bakalářskou práci je to 30 až 60 normostran);
- formální úprava technické zprávy a práce s literaturou (citacemi) jsou vyčleněny do samostatných bodů;
- hodnotí se realizační výstup;
- stejně jako v předchozím přístupu k hodnocení (viz sekce 4.2.2) se zvlášť hodnotí využitelnost výsledků;
- významným rozdílem je to, že vedoucí přikládá pouze výslednou známku, neprovádí hodnocení každého bloku, jak bylo identifikováno v předchozích modelech, pouze komentuje zadání, aktivitu a samostatnost (zvlášť v průběhu a na konci práce), práci s literaturou a publikační činnost.

### ■ 4.3 Vybrané modely hodnocení bakalářských závěrečných prací v technicky zaměřených oborech mimo ČR

Podklady pro další rešerši jsem získala dvěma způsoby. Dva modely hodnocení mi zaslal vedoucí hned na začátku práce. Byly to modely hodnocení diplomových prací Technické univerzity v Delftu v Nizozemí (viz soubor *technicka\_univerzita\_v\_delftu*) a Fakulty EEMCS Delft v Nizozemí (viz soubor *fakulta\_eemcs\_delft*). Další příklady modelů hodnocení jsem vyhledala na internetu v prohlížeči Google pomocí následujících klíčových slov: model hodnocení závěrečných prací, bachelor/diploma thesis scheme. Dále jsem hledala webové stránky konkrétních univerzit a pak jsem se dívala na stránky popisující provedení závěrečných prací, jako tomu bylo s Norskou asociací vysokých škol<sup>1</sup> (viz sekce 4.3.1).

<sup>1</sup>Odkaz na stránky: <https://www.uhr.no/en/>

### 4.3.1 Norská asociace vysokých škol

Schéma hodnocení se nachází v elektronické příloze B (viz soubor *norska\_asociace\_vysokych\_skol*).

Prvním modelem pro porovnání byl vybrán klasifikační formulář pro hodnocení bakalářských prací vytvořený Norskou asociací vysokých škol. Důvodem pro vybrání právě norské univerzity byl seznam nejlepších zemí v dosažení vzdělání z roku 2021 [9]. Výstup je založen na studii, která zkoumala více než 17 000 občanů ze čtyř regionů, aby zhodnotila výsledky ze 78 zemí za použití 76 různých metrik. Norsko se v tomto výkazu dostalo mezi 15 vítězů. Tento seznam svědčí o správném a užitečném postupu použitým na dané univerzitě, a proto je přínosným příkladem a inspirací pro zavedení změn v modelu hodnocení na SITu.

#### Přístup k hodnocení

Za odvedenou práci student může dostat body v rozsahu od 0 do 100, což je ve výsledném hodnocení převedeno na známku A (90 až 100) až F (0 až 49).

#### Kritéria hodnocení:

- Všeobecný dojem – kvalita odvedené práce, samostatnost studenta, míra iniciativy, úroveň použitých odborných znalostí, strávený čas na projektu.
- Náhled do inženýrství – dostatečnost popisu použitých základů v oblasti inženýrství, možnost zařazení práce do komplexních systémů, dostatečnost rešerší a komunikace s odborníky.
- Teoretické porozumění – kvalita teoretického přehledu, kvalita znalostí studenta v daném zaměření, dostatečnost znalostí o relevantním výzkumu a vývoji, metodách a přístupech.
- Výsledky – kvalita výsledku, podklady pro analýzu, vlastní příspěvek/-úspěchy.

Zajímavým přístupem je přidělení vah k jednotlivým kritériím v rozsahu od 10/15 do 15/20 v závislosti na kritériích. O jejich poměru v rámci celkového hodnocení se rozhoduje po dialogu se studentem a případně externistou před zahájením bakalářské práce. Další odlišností je to, že náhled do inženýrství a teoretické porozumění nemají rozpad do dílčích bodů, protože jsou závislé na individuálním zaměření práce a definují se stejně jako váhy na základě rozhovoru s vedoucím a studentem.

### 4.3.2 Německá univerzita v Káhiře

Schéma hodnocení se nachází v elektronické příloze práce B (viz soubor *nemecka\_univerzita\_v\_kahire*).

Na protiváhu norskému přístupu byla zvažena také univerzita v Káhiře, která se ve výše zmíněném seznamu univerzit umístila na jedné z posledních

pozic.

### **Přístup k hodnocení**

Přidělení počtu bodů v rozsahu od 1 do 100, což se ve výsledném hodnocení převádí na známky od F ( $\leq 49$ ), C s minusem (50–54) do A (90–93), A s plusem (94–100). Vedoucí a oponent práce posuzují přidělení bodů v rámci níže popsaných kritérií, a to maximálně do 30 bodů pro každého hodnotitele. Vedoucí pak navíc rozhoduje o extra 40 bodech vyplývajících z průběžné práce na projektu.

Jako pomůcka pro vyhodnocení se používá speciální tabulka rozdělená do třech kategorií: dodané výstupy, kvalita, prezentace. To se hodnotí každý týden v rozsahu 1 až 10 (nebo 15 – za kvalitu). Výsledné hodnocení se pak sestavuje jako průměr průběžných známek. Takový přístup pomáhá průběžné kontrole práce studenta a pozitivně přispívá k zachycení co nejvíce detailů v průběhu rozvoje projektu.

### **Kritéria hodnocení:**

- Kvalita zprávy – jedná se o kvalitu výsledného textu, kontroluje se správnost využití odborného jazyka, pravopis, interpunkce, formátování a logičnost konzistence textu.
- Rozvoj, dosažený výsledek a závěr – analyzuje se naplnění všech faktorů stanoveného problému a správnost řešení.
- Kvalita průběžných zpráv – kvalita dodaných výsledků v průběhu práce a kontrola, že všechno bylo uděláno včas.
- Kvalita práce.
- Samostatnost.

Z toho lze posoudit, že hodnocení je sepsáno velice stručně a nedává možnost ani studentovi, ani hodnotiteli zpřehlednit individuálnost práce v hodnocení. Z obecného hlediska největší váhu má „Kvalita písemné zprávy“ a „Kvalita práce na projektu“ vyhodnocená vedoucím práce. Nejmenší podíl známky má „Samostatnost“.

## **4.4 Vybraný model hodnocení bakalářských závěrečných prací v dalších oborech mimo ČR**

Tento bod stojí za zmínku v rámci tvorby kritérií hodnocení v oboru informačních technologií, i když se toho přímo netýká. I tyto příklady mohou obsahovat zajímavé podklady a poskytnout širší náhled do existujících přístupů hodnocení.

#### ■ 4.4.1 Univerzita aplikovaných věd v Mikkeli, Finsko, obor Mezinárodního podnikání

Schéma hodnocení se nachází v elektronické příloze práce B (viz soubor *univerzita\_v\_mikkeli\_model*).

Důvodem pro zvolení dané univerzity je nejen její zařazení do států s jedním z nejvyšších indexů vzdělání, ale i způsob, který univerzita používá pro hodnocení. Konkrétně se to týká přidělení vah k jednotlivým bodům hodnocení stejně jako v norském přístupu (viz. část 4.3.1).

##### **Přístup k hodnocení**

Známka je přidělena v rozsahu od 0 do 5, kde 0 znamená „Neprospěl“. Výsledná známka za bakalářskou práci se vypočítá jako vážený průměr známek za každé měřitelné kritérium. Nula za jakékoliv kritérium se počítá jako nevyhovující, což znamená neúspěch hodnocení pro studenta. Vzhledem k různé důležitosti měřitelných kritérií jim byly přiřazeny různé váhy pro použití v konečném hodnocení. Vedoucí pro hodnocení používá excelový soubor (viz elektronická příloha B soubor *univerzita\_v\_mikkeli\_formular*), ve kterém jsou předem nastavené vzorce pro vypočítání známky s přihlédnutím k vahám kritérií.

##### **Kritéria hodnocení:**

- Motivace, vztah k oboru studia – vysvětlení, jak se studie se týká jevu nebo oblasti zájmu v rámci oboru.
- Míra specifikace a omezení výzkumného problému a otázek.
- Přehled literatury – prokázání kritického myšlení, kreativity a vhledu do problematiky po přehledu již existující literatury.
- Systematičnost a logičnost přístupu k problému – jasné zdůvodnění zvoleného přístupu, porovnání přístupů.
- Koncepční struktura zprávy – jasná a konzistentní struktura práce, syntéza jiných/nových konceptů nebo hledisek.
- Sběr a používání empirických dat/materiálů (pokud je to potřeba) – identifikace vhodných materiálů, správnost a účelnost jejich použití.
- Tlumočení a diskuse o výsledcích, odvození závěrů.
- Akademický styl, využití jazyků a čitelnost.
- Konzistence a souvislost výkladu.

## 4.5 Vybrané modely hodnocení diplomových závěrečných prací

Modely v této sekci se od předchozích příkladů liší pouze náročností vypracování diplomové práce a jejího hodnocení. Jedná se o větší samostatnost a větší rozsah práce. Formuláře představují detailnější seznam požadavků a dílčích bodů, podle kterých jsou studenti hodnoceni. Proto považuji za nezbytné popsat alespoň povrchní seznámení s existujícími modely hodnocení diplomových prací.

### 4.5.1 KU Leuven Fakulta strojí technologie (FET)

Model hodnocení se nachází v elektronické příloze B (viz soubor *ku\_leuven\_fakulta\_strojni*).

#### Přístup k hodnocení

Slovní hodnocení (Nadstandardní kvalita – Velmi dobře Dobře – Uspokojivé Dostatečně – Nedostatečné – Slabé – Velmi slabé). Výsledná známka má také slovní formu a počítá se jako průměr známek přidělených za jednotlivá kritéria.

#### Kritéria hodnocení:

- Průběžná práce – přístup k provedení práce, plánování, iniciativa, samostatnost, dostatečnost komunikace, motivace studenta.
- Metodologie a výsledek – analýza zadání, správnost interpretace výsledků, kvalita výsledků, osobní příspěvek studenta.
- Forma výkazu – dodržení struktury, správnost využití jazyka, správnost citací, kvalita zdrojů.
- Obsah – kvalita odvedené analytické práce, kritické myšlení, kvalita zvolených metod.

### 4.5.2 Další příklady existujících modelů hodnocení

S dalšími příklady seznamů kritérií se lze seznámit v přílohách. Jedná se především o model hodnocení diplomových prací Technické univerzity v Delftu v Nizozemí (viz soubor *technicka\_univerzita\_v\_delftu*) a Fakulty EEMCS Delft Nizozemí (viz soubor *fakulta\_eemcs\_delft*). Rozpad na kritéria je víceméně stejný, hlavní odlišnost je v základním dělení na větší sekce jako třeba vyčlenění teoretických znalostí do zvláštní sekce, což nemá zásadní vliv na hodnocení bakalářských prací.

## 4.6 Vyhodnocení rešerše

Tato kapitola analyzovala různé modely hodnocení bakalářského a magisterského studia. Rešerše se zaměřovala jak na univerzitu, ve které bude řešení použité, tak i na další univerzity s různými specializačními zaměřenými.

Na základě rešerše jsme s vedoucím práce definovali úvahy o tom, co může působit problémy pro studenty a hodnotitele z pohledu pochopení požadavků na práci a vyplňování hodnocení. Tyto úvahy jsou uvedené níže.

### ■ Úvaha 1. Velká míra abstrakce modelu

*Příliš abstraktní a krátký seznam hodnotících kritérií. V každém bloku chybí členění na dílčí kritéria, což by vyloučilo přílišnou subjektivitu při hodnocení a zavedlo přesné pochopení toho, co se od studenta požaduje.*

### ■ Úvaha 2. Hlavní bloky modelu

*Hlavní bloky budoucího modelu by měly odpovídat základnímu rozdělení stávajícího formuláře, aby nedocházelo k nedorozuměním mezi studenty, hodnotiteli a ke zbytečnému přecházení z jednoho hodnotícího přístupu do jiného.*

### ■ Úvaha 3. Rozpad hlavních bloků na dílčí kritéria

*Je třeba rozšířit popis požadavků v rámci hlavních bloků (viz. Úvaha 2. Hlavní bloky modelu) o přesně definovaný seznam dílčích kritérií, která budou zohledněna při hodnocení prací – rozšíření Úvahy 1. Velká míra abstrakce.*

### ■ Úvaha 4. Přidělení vah za bloky a dílčí kritéria

*Každý z hlavních bloků by měl být doplněn o možnost pro vedoucího nastavit jeho váhu. Student tak má lepší porozumění přístupu vedoucího a průběhu práce. Vedoucí tak může přizpůsobit model přístupu k hodnocení, na něž je zvyklý. Totéž platí pro kritéria v rámci dílčích bloků.*

Rozsah vah bude stanoven až po rozhovorech s vyučujícími. Inspirace je čerpána z přístupu Norské asociace vysokých škol (více v části 4.3.1).

Počet prací vedených vyučujícími může být poměrně velký, což vede ke snížení pravděpodobnosti, že si vedoucí bude pamatovat všechny detaily průběžné práce studenta. To může v konečném důsledku znamenat nepřesnosti v hodnocení. Na příkladu Německé univerzity v Káhiře 4.3.2) přicházíme s následující úvahou:

### ■ Úvaha 5. Průběžné monitorování práce

*Pro přesnější kontrolu průběžné práce studenta by jako pomůcka k hodnocení měla být poskytnuta tabulka s monitorováním práce, která by*



vycházela z harmonogramů povinných schůzek a zpráv o odvedené práci.

Úvaha 5 po rozhovoru s vedoucím byla odložena. Dohodli jsme se, že tento předpoklad nebudeme dál rozvíjet, protože to není prioritou daného výzkumu, ale mohlo by být využito jako návrh pro další rozšíření projektu.

#### ■ Úvaha 6. Celkové a blokové hodnocení

*Každý blok bude ohodnocen body, které celkem dávají 100 bodů. Tento přístup je uživatelsky přívětivější a snadněji implementovatelný. Výsledný počet bodů bude převeden na známku A – F dle klasifikační stupnice ČVUT[10], aby vyhovoval aktuálně používaným formulářům posudků.*

Výsledky dané rešerše plně naplňují Dcíl1. Analýza existujících modelů hodnocení a poskytuje základnu pro lepší pochopení Dcíl2. Definice problémů při hodnocení. Stanovené úvahy budou ověřené v dalších kapitolách na základě provedených pohovorů.



## Kapitola 5

### Vliv typu práce na její hodnocení

Kromě struktury modelů hodnocení a využívaných přístupů k hodnocení existuje další parametr, který výsledné hodnocení ovlivňuje. Jedná se o zaměření práce (viz kapitola 1). Tato kapitola zohledňuje vazbu zaměření práce a jejího hodnocení. Na začátku kapitoly jsou uvedeny důvody toho, proč zaměření musí být zváženo při návrhu hodnocení. Pak se uvádí příklady existujících zaměření v programu softwarového inženýrství na FEL. Na konci kapitoly je uvedena úvaha, která rozšiřuje úvahy stanovené v závěru předchozí kapitoly.

#### 5.1 Důležitost zaměření práce při hodnocení

Zaměření práce je důležitým kritériem při přidělování známek za odvedenou práci v průběhu hodnocení. Konkrétně má silnou váhu při rozhodování o odborné úrovni studenta a zvoleném přístupu k řešení problémů. Zvláště musí být zváženo pro Úvahu 3. Rozpad hlavních bloků na dílčí kritéria (viz stránka 30). Jedním z důvodů je to, že se program Softwarové inženýrství aktuálně nedělí na obory. Jedná se v podstatě o získání základů v různých oborech, které do tohoto programu spadají. A proto si studenti jako téma závěrečného projektu mohou vybrat z poměrně široké nabídky: od výzkumných/analytických prací po práce v oboru IoT. Specifika těchto zaměření jsou důležitým faktorem, jež hodnocení ovlivňuje.

Stanovení všech možných povinných a specifických kritérií pro dané zaměření je příliš obtížný úkol kvůli jejich velkému počtu a častému překrytí kritérií mezi zaměřeními. V podstatě by se mohlo jednat o další rozvoj této práce nebo o téma pro další závěrečnou práci. Vzhledem k tomu bylo po dohodě s vedoucím rozhodnuto, že v rámci dané práce uvedeme jen základní rozpad existujících zaměření prací do šesti typů. Nejedná se o finální výčet zaměření, ale o seznam, kterým jsme se zabývali během další analýzy. Upřesnění toho seznamu může být bráno jako další možnost budoucího zlepšení projektu.

## 5.2 Dělení závěrečných prací podle zaměření

### Návrh softwaru

Jedná se o návrh rozhraní, funkcí, datových struktur, architektury chování, řízení a bezpečnosti softwaru.

### Ověření softwaru

Práce se může zabývat zkoušením/simulací provozu softwaru, analýzou existujícího softwaru s cílem nalézt chyby a problémy, měřením funkcionality a kvality softwaru; zhodnocením atributů a schopností softwaru, zda dosahují požadovaných či akceptovatelných výsledků, inspekci, prováděním testování kódu.

### Aplikační

Implementace programu pro zpracování a řešení konkrétní úlohy řešené uživatelem. Příkladem mohou být textové editory a procesory; tabulkové procesory; prezentační program; databáze; grafický software; technický software; internetové aplikace; pomocný software; ekonomický software; plánovací software; multimediální software; hry a zábavní aplikace.

### Hardware

Práce s technickými prostředky, jejich propojení prostřednictvím počítačové sítě, napojení na paměťové jednotky, doplnění o potřebné periferní jednotky, zkoumání fyzického vybavení ve vztahu k jeho spolehlivosti, bezpečnosti a použitelnosti se softwarovým vybavením.

### Internet Of Things (IoT)

Nastavení komunikace v síti fyzických a elektronických zařízení, která jsou vzájemně schopna si odesílat a přijímat data.

### Výzkumné (analytické)

Práce zabývající se analýzou požadavků na procesy a potřeby, napsáním podrobné technické dokumentace, sběrem a analýzou dat s předložením rozumného výstupu, provádění průzkumů, chatboty.

Jak je z krátkého výčtu zaměření vidět, je možné realizovat práce od čistě teoretického zaměření, až po ryze praktické. Čím více teoretické téma, tím více úsilí je třeba věnovat studiu problematiky a hledání obecných řešení. U praktičtějších je zase vyžadováno větší úsilí na vytvoření konkrétního výstupu

a jeho ověření. Na základě toho lze vytvořit následující úvahu, která upřesňuje Úvahu 3. Rozpad hlavních bloků na dílčí kritéria (viz stránka 30):

■ **Úvaha 7. Volitelná kritéria pro odbornou úroveň**

*Kritéria vztahující se k bloku „Odborná úroveň“ nelze omezit na přesný seznam dílčích kritérií. Seznam se musí lišit v závislosti na zaměření práce (viz 5. Zaměření práce). Seznam se definuje po diskuzi mezi vedoucím a studentem na základě schváleného zadání. Doporučením je stanovení seznamu na začátku práce s možností jeho budoucí korekce.*



## Kapitola 6

### Vnímání stávajícího přístupu k hodnocení absolventy

Po dohodě s vedoucím bylo rozhodnuto, že v dalším kroku výzkumu budou provedeny pohovory s absolventy programu SIT. Pohovory byly zaměřeny především na pochopení toho, s jakými problémy se absolventi při tvorbě svých projektů potýkali, co jim dělalo problémy a co pomáhalo v jejich práci. Zpětná vazba absolventů byla taky zaměřená na analýzu stávajícího modelu hodnocení (viz 4.2.1). Výstupy těchto pohovorů pomohly potvrdit Úvahu 1. Velká míra abstrakce modelu (viz stránka 30). V neposlední řadě jsme se chtěli dozvědět, jestli názory studentů a jejich vnímání závěrečné práce jsou ovlivněny zaměřením práce (zvažování Úvahy 7).

Hlavním cílem bylo dostat zpětnou vazbu na výše zmíněné úvahy. Výsledky setkání poskytly základnu jak pro vytvoření prvního prototypu modelu hodnocení, tak i pro následné pohovory s hodnotiteli. Jelikož tyto pohovory byly odvedeny jako první, neměly pevnou strukturu. Navazující konverzace se vyvíjela na základě poskytnutých odpovědí respondentů.

Celkem se rozhovorů zúčastnilo 6 absolventů bakalářského studia v programu Softwarové inženýrství.

#### 6.1 Shrnutí připomínek absolventů

Po rozhovorech jsem došla k následujícím závěrům spojeným s mým výzkumným projektem:

- Rozložení počtu výzkumných a aplikačně zaměřených prací bylo rovnoměrné.
- Typ práce, ani zaměření neměly vliv na odpovědi studentů. Zkušenosti s plněním závěrečného projektu v obou případech byly téměř stejné.
- Studenti potvrdili, že na začátku práce na projektu často měli problém s porozuměním tomu, co se od nich ve finále očekává. Rozhodně by uvítali přesnější popis požadavků, a pokud je možné, i kategorizaci dle zaměření práce. To potvrzuje Úvahu 1. Velká míra abstrakce modelu a poskytuje základ pro další postup v Úvaze 3. Rozpad hlavních bloků na dílčí kritéria a 4.Přidělení vah za bloky a dílčí kritéria (viz stránka 30).





## Kapitola 7

### Návrh prototypů modelu hodnocení

Před uskutečněním pohovorů s další skupinou respondentů (více v části 3.7 Respondenti) bylo rozhodnuto nejprve vytvořit prototyp (viz kapitola 1) první verze modelu hodnocení. Během pohovorů tak zjistíme obecný názor na aktuální stav formulářů a přístupů k hodnocení. Spolu s tím dostaneme první zpětnou vazbu k navržené verzi modelu, což urychlí práci. Jako základ návrhu jsme použili úvahy stanovené v předchozích kapitolách. Navíc jsme společně s vedoucím během práce rozšířili předchozí úvahy o nové. Tyto úvahy také byly zohledněny v návrhu a předloženy ke zpětné vazbě hodnotitelům.

#### ■ Úvaha 9. Průnik náročnosti zadání a jeho splnění

*Bloky Zadání a Splnění zadání nelze rozdělovat a hodnotit samostatně, protože jeden na druhý navazují.*

Poznámka: Čím náročnější je zadání, tím větší počet menších výhrad může být "odpuštěn" studentovi a naopak.

#### ■ Úvaha 10. Dopad nevyhovujícího blokového hodnocení na výslednou známku

*Pokud alespoň jedna z částí posudku byla označena za nesplněnou (odpovídá známce F), výsledná známka nemůže nabýt jiné hodnoty než F.*

Poznámka: Tuto úvahu jsme zvažovali již během analýzy existujících přístupů na jiných univerzitách. Stejný postup měla univerzita aplikovaných věd v Mikkeli (viz 4.4.1). V rámci této práce bylo rozhodnuto tento postup vyčlenit do samostatné úvahy, abychom zjistili názor hodnotitelů na tento předpoklad.

Po diskuzi s vedoucím práce jsem přišla na to, že přínosnější bude rozdělit první verzi modelu hodnocení do dvou prototypů. Oba dva budou mít společnou kostru neboli jádro, ale budou zohledňovat různé přístupy hodnocení pro lepší přehlednost. Respondenti by tak dostali více podkladů pro srovnání a ve výsledků by poskytli kvalitnější zpětnou vazbu, která buď stanovené úvahy potvrdí, nebo vyvrátí. V dalších částech kapitoly postupně popíšu výše zmíněnou kostru a rozeberu hlavní odlišnosti dvou prototypů.

## 7.1 Kostra navržených prototypů

Kostra prototypů vychází z následujících charakteristik:

- Formulář je rozdělen do pěti bloků: 1. Aktivita a samostatnost, 2. Odborná úroveň, 3. Formální a jazyková úroveň, rozsah práce, 4. Výběr zdrojů, korektnost citace, 5. Průnik zadání/splnění zadání.

Charakteristika zohledňuje Úvahu 2. Hlavní bloky modelu (viz stránka 30) a poskytuje možnost hodnotit práci s přihlédnutím k Úvaze 9. Průnik náročnosti zadání a jeho splnění (viz stránka 39)

- Prvním čtyřem blokům dále říkáme **základní bloky**. Základní bloky jsou rozděleny na dílčí kritéria (viz tabulka 7.1).

Blok	Dílčí kritéria
1. Aktivita a samostatnost	1. Dodržení domluvených termínů 2. Přípravenost k setkáním 3. Iniciativnost 4. Schopnost samostatně postupovat v rozvoji projektu 5. Pečlivost
2. Odborná úroveň	1. Použitelnost, aplikovatelnost v praxi 2. Možnosti dalšího rozvoje 3. Systematičnost analýzy 4. Replikovatelnost závěru Další kritéria se stanovují po dohodě s vedoucím na základě zaměření práce
3. Formální a jazyková úroveň, rozsah práce	1. Adekvátnost rozsahu 2. Formátování 3. Gramatické chyby, interpunkce, překlady 4. Souvislost, konzistence textu 5. Čitelnost 6. Použití odborného jazyka 7. Reference v textu
4. Výběr zdrojů, korektnost citace	1. Dodržování konvencí 2. Dostatečnost citací 3. Kvalita zdrojů (stáří a charakteristika literatury)

**Tabulka 7.1:** Rozpad základních bloků na dílčí kritéria

Seznam kritérií byl definován na základě analýzy použitých formulářů z jiných univerzit (viz kapitola 4 a stanovené úvahy). Byla použita ta kritéria, která se nejvíc shodovala napříč modelům, a ta, která jsou relevantní pro hodnocení bakalářských prací. Tento rozpad zohledňuje Úvahu 1. Velká míra abstrakce modelu a 3. Rozpad hlavních bloků na dílčí kritéria (viz stránka 30).

- Blok Odborná úroveň obsahuje jak pevná kritéria, tak i volitelná. Tato kritéria může, ale nemusí doplnit vedoucí po diskuzi se studentem v závislosti na zadání a zaměření práce (zohledňuje Úvahu 7. Zvláštní požadavky na odbornou úroveň, viz stránka 35)
- Student může získat 0–100 bodů, které mu budou převedeny do známky A–F (zohledňuje Úvahu 6. Celkové a blokové hodnocení, viz stránka 31)
- Nedostatečně náročné zadání nebo jeho nesplnění se okamžitě počítá jako F pro celý formulář posudku (zohledňuje Úvahu 2. Hlavní bloky modelu, Úvahu 10. Dopad nevyhovujícího blokového hodnocení na výslednou známku, viz stránka 39)

## 7.2 Hlavní rozdíly prototypů

Se dvěma variantami prototypů modelu se lze seznámit v elektronické příloze B (viz soubor *prototypes\_v1*). Tabulka č. 7.2 znázorňuje hlavní rozdíly navržených prototypů vzhledem k jednotlivým částem modelu hodnocení (první sloupec tabulky). První prototyp modelu se nazývá "Pevný" kvůli přesně danému rozdělení bodů mezi bloky (viz druhý sloupec tabulky). Druhý prototyp se nazývá "Dynamický", jelikož nabízí možnost manuálního nastavování vah (viz třetí sloupec). Poslední sloupec popisuje to, která úvaha byla zohledněná využitím příslušného přístupu.

Pro zohlednění Úvahy 9. Průnik náročností zadání a jeho splnění zavedeme pojem koeficientu.

**Koeficient** – vypočítaná hodnota, která přímo závisí na splnění a náročnosti zadání a počítá se na základě následujícího vzorce:

$$(\text{Incr}/\text{Upper} + \text{Incr}/\text{Lower}) / 2 + 1, \text{ kde}$$

*Incr* – hodnota, kterou lze známku navýšit dle tabulky 7.3 Inkrement;

*Upper* – horní hranice intervalu dle stupnice ČVUT[10], do které spadá dosažený počet bodů studenta;

*Lower* – dolní hranice intervalu dle stupnice ČVUT[10], do které spadá dosažený počet bodů studenta.

Tento vzorec jsem sama odvodila na základě analýzy již existujících hodnocení, pohovorů vedoucím. Došla jsem k závěru, že známka vzhledem k náročnosti a splnění může být upravena maximálně o 8 bodů, což je dáno incrementem (viz tabulka 7.3). Horní a dolní hranice zajistí, že increment bude stejný pro všechny bodové rozsahy výsledné známky.

Koeficientem se automaticky vynásobí studentem dosažené body za základní bloky. Se všemi koeficienty je možné se seznámit v již zmíněné elektronické příloze B (viz soubor *prototypes\_v1* tabulka "Koeficient").

Část	Pevný	Dynamický	Poznámka
Zadání/ splnění zadání	Koeficient, kterým se vynásobí dosažený počet bodů za základní bloky. Musí být doplněn o řádné zdůvodnění vedoucího. Závisí na náročnosti a splněnosti zadání.	Je ohodnoceno známkou A–F, která závisí na dvou podmínkách: náročnosti a splnění zadání. Musí být zdůvodněno vedoucím. Celkem má váhu 20% z výsledné známky (max. 20 bodů).	Zohledňuje Úvahu 9.
Základní bloky	Rovnoměrné rozdělení bodů (25 za blok) mezi bloky bez zadávání vah.	Nastavitelné váhy v předem daném rozsahu. Váhy se počítají v procentech, dohromady musejí dávat 80 % (max. 80 bodů).	Zohledňuje Úvahu 4.
Dílčí kritéria	Nehodnotí se	Přidělení vah z předdefinovaného rozsahu. Dohromady musejí dávat 100% . Počítají se od počtu bodů za blok.	Zohledňuje Úvahu 1 a 4.
Výsledná známka	Počet bodů za první čtyři bloky vynásobený koeficientem z posledního bloku.	Počítá se jako součet bodů (max. 100 bodů) za všechny bloky a převádí se do příslušné známky A–F.	Zohledňuje Úvahu 6 a 9.

Tabulka 7.2: Hlavní rozdíly prvních verzí prototypů

	Splněno	Splněno s menšími výhrami	Splněno s většími výhrami	Nesplněno
Mimořádně náročné	8	4	0	F
Náročnější	6	2	-2	F
Průměrně náročné	0	-2	-6	F
Lehčí	-2	-4	-8	F
Nedostatečně náročné	F	F	F	

Tabulka 7.3: Inkrement

## Kapitola 8

### Vyhodnocení navržených prototypů hodnotiteli

Po vytvoření prototypů modelu hodnocení bylo možné přistoupit ke druhému kolu sběru dat. Do druhého kola sběru dat se zapojili zaměstnanci univerzity, kteří se buď aktuálně věnují vedení prací studentů a jejich oponentuře, nebo jsou součástí komise, která přímo rozhoduje o závěrečném hodnocení práce. Pohovorů se zúčastnilo 5 vyučujících zastupujících různé obory a programy studia, a to jak FEL, tak i FIT. Respondentům byly předloženy dva navržené prototypy modelů hodnocení závěrečných prací, které jsou popsány v předchozí kapitole. Pohovory byly provedeny především za účelem získání odpovědí a názorů na následující otázky:

- Mají navržené prototypy hodnocení na první pohled smysl, aniž bychom se zabývali podrobnostmi metodiky? Odpověď zdůvodněte.
- Po vyslechnutí vysvětlení každého přístupu, který z nich by byl pro Vás v praxi nejvhodnější, proč?
- Souhlasíte s rozdělením základních bloků do dílčích kritérií? Co byste přidali nebo odstranili?
- Jaký je Váš názor na přidělení vah základním blokům a dílčím kritériím.
- Souhlasíte s přístupem ke spočítání výsledné známky a proč?
- Měl by mít oponent přístup k dílčím kritériím a dalším detailům hodnocení stanoveným vedoucím a studentem?
- Obecné návrhy na vylepšení formuláře. Existují části, které jsou v rozporu s Vaším chápáním přístupů k vyhodnocení?

Další otázky a směřování pohovorů byly doplňovány v průběhu diskuze. Výsledky těchto diskuzí pomohly zhodnotit a upřesnit v předchozích částech stanovené úvahy.

#### 8.1 Připomínky k navrženým prototypům

Z pohovorů vplynuly následující závěry:

- Všichni respondenti se shodli, že oponent by neměl vidět, jak si vedoucí a student práci naplánovali. To znamená, že by neměl mít přístup k modelu hodnocení, který vedoucí přizpůsobil konkrétně pro daného studenta vzhledem k jeho zaměření – vyvrácení Úvahy 8. Přístup oponenta k modelu (viz stránka 38)
- V rámci jednoho bloku je zbytečné doplňovat váhy pro každé kritérium bloku. Je to obtížné na počítání – doplnění Úvahy 4. Přidělení vah (viz stránka 30).
- Většina respondentů se přiklání k variantě s možností nastavení vah základních bloků, jeden z respondentů měl za to, že rozsah vah by měl být větší – doplnění Úvahy 4. Přidělení vah.
- Půlka respondentů hlasovala pro kombinaci dvou tabulek, kde by se používaly koeficienty z průniku Zadaní a Splnění zadání a bylo by umožněno nastavování vah pro základní bloky.
- Náročnost zadání by se měla definovat po konzultaci s osobou, která má s tímto typem práce zkušenosti, například externista – přímo se netýká záměru práce a dál se nerozebírá.
- Od jednoho z respondentů padla připomínka, že zadání a jeho splnění nemůže být hodnoceno jako zvláštní bod a musí vycházet z odborné úrovně.

Přesný seznam těchto kritérií byl spolu s respondentem upřesněn a vyjmenován níže v rámci úvahy.

Poslední připomínka byla přímo vyčleněná do nové úvahy:

■ **Úvaha 11. Vliv kritérií odborné úrovně na splnění zadání**

*Kritéria, která spadají do odborné úrovně (a to použitelnost, aplikovatelnost v praxi, systematickosti analýzy a volitelná), přímo ovlivňují splnění zadání. Body odebrané za tato kritéria by měly být v poměru odebrané i za splnění zadání.*

Obecně lze říci, že hodnotitelé jsou často nuceni sami vymýšlet, co musí být naplněním jednotlivých bloků posudků. To výrazně prodlužuje jejich práci. Vzhledem k tomu se hodnotitelé shodli, že by rozhodně uvítali přesnější definici kritérií hodnocení a celkového modelu hodnocení. Musím ale zmínit, že se přístup k hodnocení z pohledu vedoucích, kteří mají na starosti větší počet prací, liší od přístupu vedoucích, u kterých je těchto prací méně. To se projevilo v tom, že druhá skupina nechtěla moc měnit stávající model, jelikož je schopna držet v hlavě všechny potřebné náležitosti na vedení práce.

Výsledky této kapitoly rozšiřují Dcíl2: Definice problémů při hodnocení a naplňují 5. krok harmonogramu (viz tabulka 3.1).

## Kapitola 9

### Finalizace přístupů k provedení hodnocení

Tato kapitola uzavírá návrh prototypů modelů hodnocení z kapitoly 7 a zabývá se jejich rozšířením o další přístupy k hodnocení na základě zpětné vazby z předchozí kapitoly. Na začátku se popisuje kostra všech definovaných přístupů, pak se rozebírají odlišnosti každého z přístupů a uvádějí se jejich možné kombinace. Vytvořené řešení je na konci kapitoly ohodnoceno pomocí jeho aplikace na již ohodnocené práce.

#### 9.1 Upřesnění obsahu modelu hodnocení

Z výsledků rozhovorů s hodnotiteli je patrné, že názory na otázku, který přístup k hodnocení zvolit jako hlavní a jediný, byly rozdílné. Někteří také navrhovali vytvoření různých kombinací prototypů. Analýza tedy nepřinesla očekávaný výsledek, na základě kterého bychom mohli navrhnout jednotný model. Abychom uspokojili většinu požadavků respondentů, rozhodli jsme se po debatě s vedoucím práce nesjednocovat navržené prototypy do jednoho modelu hodnocení, ale dát uživatelům prostor zvolit si vhodný přístup k hodnocení s ohledem na jejich zvyklosti. Z toho vznikla další úvaha:

##### ■ Úvaha 12. Možnost konfigurace přístupů k hodnocení

*Místo jednotného modelu hodnocení by uživatelům měla být poskytnuta možnost kombinovat různé přístupy k hodnocení dílčích částí modelu hodnocení.*

Jádrem modelu tak zůstávají základní bloky stávajícího modelu (viz 4.2.1): 1. Aktivita i samostatnost, 2. Odborná úroveň, 3. Formální a jazyková úroveň, rozsah práce, 4. Výběr zdrojů, korektnost citace; a dílčí kritéria sestavená v předchozích fázích (viz tabulka 7.1. Dílčí kritéria). Hodnocení (jak celkové, tak i za každý blok) mohou mít různé formy v závislosti na přístupu, který je pro uživatele vhodnější. Možné rozdíly v přístupech k hodnocení a sestavování modelu jsou popsány v další části kapitoly. Možné kombinace navržených přístupů a jejich hodnocení ukazuje sestavená tabulka v elektronické příloze práce B (viz soubor *approach\_combinations*).

## 9.2 Možné přístupy k ohodnocení dílčích částí modelu

Dále uvádím přístupy k hodnocení dílčích částí modelu.

### 9.2.1 Bloky: zadání a splnění zadání

1. Dva samostatné bloky: stejný přístup jako ve stávajícím modelu hodnocení
  - a. slovní hodnocení (Mimořádně náročné až Nedostatečně náročné; Splněno až Nesplněno) a komentář ke každému bloku;
  - b. známka A až F;
  - c. bodové hodnocení: možné maximum záleží na zvolení přístupu k hodnocení dalších částí modelu;
  - d. koeficient podílu na známce.
2. Průnik dvou bloků:
  - a. hodnocení známkou A až F;
  - b. bodové hodnocení: možné maximum záleží na zvolení dalších částí modelu;
  - c. koeficient podílu na známce.

Pro každý ze dvou přístupů lze navíc zvolit možnost, kde se splnění zadání bude vypočítávat automaticky na základě kritérií definovaných v Úvaze 11. (viz stránka 44).

### 9.2.2 Základní bloky

1. Známkové: známka A až F za každý blok;
2. Bodové:
  - a. fixní počet rovnoměrně rozdělených bodů za každý blok (možné maximum záleží na zvolení dalších částí modelu, viz body níže);
  - b. nastavitelný počet bodů pomocí vah v předem daném rozsahu.

### 9.2.3 Dílčí kritéria

1. Nehodnotí se;
2. Procentuální hodnocení v rozsahu 0 až 100%.

Hodnocení základních bloků se může vypočítávat automaticky dle průměru dosažených procent za dílčí kritéria bloku.



## 9.3 Možné kombinace stanovených přístupů

Možnosti kombinace přístupů a jejich ohodnocení je zohledněno v elektronické příloze B v souboru *approach\_combinations*. Příloha obsahuje dva listy. První kombinuje možnosti hodnocení, kde bloky zadání a splnění zadání představují dva samostatné bloky. Ve druhém listu jsou tyto bloky sjednocené a hodnotí se jejich průnik. Další struktura listu je podobná. Řádky tabulek odpovídají možnostem hodnocení bloků zadání a splnění zadání, sloupce představují možnosti hodnocení základních bloků. Buňky průniku řádků a sloupců odpovídají příslušným kombinacím hodnocení, obsahují detailnější informace o kombinaci a mohou být označeny jednou ze třech barev. Červená znamená, že kombinace neexistuje kvůli obtížnosti nebo nepraktičnosti pro použití. Černá znamená, že kombinace by mohla existovat, ale dál se nebere v úvahu a neanalyzuje se. Neexistence kombinace přístupů je nejčastěji zdůvodněná zbytečným převodem známek na body a naopak. Nebo kombinace již má ekvivalent v jedné z dalších kombinací uvedených v tabulkách. Celkově bylo odstraněno 6 přístupů. Zelená označuje kombinace, které budou podrobeny další analýze. Poslední sloupec tabulek obsahuje informace o vzorci pro spočítání výsledné známky.

Jak je vidět z přílohy, z 15 různých vytvořených možností kombinací přístupů k hodnocení po odstranění nepoužitelných přístupů zbylo 9 kombinací. Ty jsou sekundárně analyzovány v následující sekci.

## 9.4 Ověření vytvořených přístupů

Celkem 9 přístupů k hodnocení z předchozí sekce bylo otestováno na dvou pracích dvou různých vyučujících FEL SIT. Záměrem analýzy bylo na základě výsledku testování omezit seznam přístupů a odstranit ty, které nemají dostatečnou přesnost. Ačkoliv jsem předpokládala, že takových přístupů bude alespoň půlka, výsledky testování moje předpoklady vyvrátily. Žádný z přístupů nebyl eliminován. Ve stejné elektronické příloze v souboru *approach\_combinations* modrou barvou u každého přístupu jsou poznamenané výsledné známky dvou prací. Pořadí známek u každého přístupu odpovídá pořadí testovaných prací. Jak je vidět, body se pohybují v rozmezí čtyř bodů a ve výsledku dávají stejnou známku, což u obou příkladů je B. Vzhledem k tomu bylo po diskuzi s vedoucím rozhodnuto, že základem budoucí aplikace bude všech 9 přístupů. To znamená, že uživatelům bude nabídnut konfigurátor, ve kterém si budou moci vybrat z dostupných možností svůj preferovaný přístup. Výsledky dané kapitoly naplňují Dcíl4. Finalizace modelu.

Tímto uzavírám analytickou část práce. Úvahy stanovené během této části práce byly zohledněné v rámci navržených kombinací přístupů hodnocení dílčích částí modelu. Příloha, která byla v této části rozebrána (viz soubor *approach\_combinations*), je základem pro další aplikační práci související s návrhem a implementací aplikace. Po provedení testování vytvořeného řešení je

ve finální fázi výzkumu vyhodnocen stav všech úvah výzkumu a analyzována naplněnost cílů (viz kapitola 13).



## Část II

### Aplikační část práce



# Kapitola 10

## Návrh aplikace

Kapitola se zabývá návrhem aplikace pro pomoc v hodnocení závěrečných prací a také výběrem vhodných technologií pro její následnou implementaci. Na začátku kapitoly jsou stanoveny požadavky na budoucí aplikaci. Dále je popsána zvolená architektura a zdůvodněn výběr technologií pro její realizaci. Kapitola končí definicí datového modelu a návrhem struktury webu.

### 10.1 Obecný popis požadavků na budoucí aplikaci

Na základě odvedené výzkumné práce (viz část I) bylo navrženo několik přístupů k hodnocení závěrečných prací studentů, které byly ověřeny na dvou již ohodnocených pracích. Vzhledem k tomu, že se výsledky testů od sebe lišily maximálně o 4 body, což v podstatě nezměnilo stupeň hodnocení celé práce, bylo rozhodnuto, že základem budoucí aplikace budou všechny definované přístupy (viz elektronická příloha B soubor *approach\_combinations*). Možnost hodnotit práci pomocí přístupu zvoleného vedoucím je základním požadavkem pro první verzi aplikace. Navíc aplikace musí obsahovat popis modelu a navržených přístupů pro lepší pochopení aplikace uživateli. Detailní seznam všech stanovených požadavků je uveden v následující kapitole (viz sekce č.10.3).

Je třeba poznamenat, že cílem této práce není vytvořit dokonale propracovanou aplikaci se všemi technickými specifikacemi. Cílem je vytvořit první zjednodušenou pracovní verzi neboli prototyp a ověřit, zda by navrhované řešení hodnocení (model a přístupy) bylo užitečné pro hodnotitele a jejich studenty. Kromě toho testování dané verze aplikace pomůže zjistit, zda by bylo možné práci dále rozvíjet, aby se zlepšila z hlediska implementace a funkcionalit.

Daná kapitola se zabývá naplněním dvou dílčích cílů projektu č. 5 a 6 stanovených v kapitole 3.3. Cíle výzkumu. Konkrétně se jedná o výběr technologií a návrh aplikace.

## 10.2 Business cíle a business požadavky

Před definicí konkrétních systémových požadavků na první verzi aplikace, bylo rozhodnuto sestavit seznam obecnějších business cílů a požadavků na budoucí aplikaci, protože ty se v softwarovém inženýrství využívají pro popis použitelnosti aplikace a smyslu.

V obecné definici business cíle udávají směr budoucí aplikace. Vysvětlují, čeho její implementací a kladeným úsilím chceme dosáhnout, ale nepopisují, jakým způsobem bude těchto cílů dosaženo [11]. Business požadavky popisují detailnější rozpad cílů na jednotlivé podmínky, které je třeba splnit, aby stanovené cíle byly dosažené [12]. Na rozdíl od systémových požadavků se nejedná o definici konkrétního systémového řešení, ale chápání problému z obecnějšího hlediska.

Vytvořený seznam vychází z úvah definovaných na začátku a v průběhu práce a dalších předpokladů stanovených spolu s vedoucím. Business cíle a požadavky byly vypracovány pomocí nástrojů Enterprise Architect. Do diagramu byly zahrnuty cíle, kterých má být realizací dosaženo, i cíle, jež se mohou týkat budoucího rozšíření a zlepšení aplikace. Ty nejsou klíčové pro úspěšné dokončení této práce, a proto se dál nerozebírají. Kompletním diagram business cílů a požadavků znázorňující jejich hierarchii a realizaci je k dispozici v příloze na obrázku A.1 a v elektronické příloze B v souboru *business\_cile\_a\_pozadavky*. Cíle, které budou v průběhu implementace naplněné, jsou v diagramu označeny zeleně. Částečně realizované cíle jsou označeny žlutě. Cíle, jež by mohly být v budoucnu užitečným rozšířením aplikace, jsou označeny modře. Níže jsou uvedeny pouze cíle, které ke konci práce budou realizované.

### ■ BG1<sup>1</sup>: Zrychlení a zefektivnění procesu vyhodnocení prací

Chceme, aby vedoucí a oponenti měli pohodlný systém pro vyhodnocení prací studentů, který by odpovídal jejich osobním přístupům a zvykům, tím by umožnil zpracovávat více posudků za kratší dobu.

#### ■ BR1.1<sup>2</sup>: Kombinace přístupů

Jako vedoucí práce chci mít přehled o možných kombinacích přístupů k hodnocení, abych při práci s konkrétním studentem mohl zvolit tu, která mi nejvíc vyhovuje a která umožní přizpůsobit se specifikům dané práce.

#### ■ BR1.2: Přizpůsobení přístupu k hodnocení

Jako vedoucí chci přizpůsobit hodnocení konkrétní práce svému obvyklému postupu pomocí zvolení vhodné kombinace přístupů k hodnocení jednotlivých bloků modelu.

#### ■ BR1.3: Ohodnocení prací

Jako vedoucí chci rychle vytvořit hodnocení konkrétní práce na

<sup>1</sup>BG je zkratka pro anglický pojem *business goal*.

<sup>2</sup>BR je zkratka pro anglický pojem *business requirement*.

základě předem daných kritérií a bodových ohraničení s možností ovlivnit výslednou známku.

- **BG2: Zvýšení povědomí studentů o kritériích hodnocení a průběhu práce**

Chceme, aby studenti již na začátku práce na projektu měli jasný přehled o průběhu, požadavcích prací a kritériích hodnocení a to včetně specifik závislých na zaměření jejich prací.

- **BR2.1: Jasný přehled kritérií a bodových rozsahů**

Jako student chci mít definitivní seznam kritérií, podle kterých bude práce hodnocena, a bodové rozsahy, aby se neztrácel čas pochopením struktury projektu, ale rovnou se mohlo pustit do práce na projektu.

- **BR2.2: Upřesnění požadavků**

Jako student chci mít možnost na začátku práce na projektu spolu s vedoucím přizpůsobit a upřesnit požadavky vzhledem k zaměření práce, aby výsledné hodnocení reflektovalo specifika projektu.

## 10.3 Analýza systémových požadavků

Systémové požadavky ověřují, jestli potřeby zákazníka definované pomocí business požadavků mohou být implementovány. Analýza systémových požadavků umožňuje dohodnout se mezi stakeholdery a dodavateli tím, že poskytne přesnější seznam funkčních a nefunkčních požadavků na systém, které zaručí nebo vyvrátí stanovené cíle [13]. Mezi stakeholdery pro daný projekt spadají studenti a hodnotitelé.

Přístup k tvorbě systémových požadavků je stejný jako i v předchozí sekci. Diagram znázorněný v příloze na obrázku A.2 obsahuje návaznost jednotlivých systémových požadavků na již popsané business požadavky. Ty se také odlišují pomocí barev, kde zelená znamená plnou realizaci, žlutá částečnou realizaci a modrá možné vylepšení pro další rozvoj projektu. Níže jsou pro seznámení uvedeny pouze požadavky, které ke konci práce jsou realizovány.

- **NFR1: Autorizace**

Systém umožní přihlášenému uživateli provádět operace, ke kterým má příslušná oprávnění. Tato oprávnění jsou jednoznačně určena uživatelskými rolemi. Mezi tyto role patří vedoucí a student.

- **FR1: Výběr vhodné kombinace přístupů**

Systém vedoucímu umožní vybrat mezi existujícími přístupy vhodnou kombinaci nebo postupně nakonfigurovat model pomocí odpovědí na otázky v dynamickém seznamu.

- **FR2: Přehled existujících přístupů**

Systém umožní zobrazit seznam existujících přístupů a kombinací včetně jejich popisu a způsobu vypočítání výsledné známky.

- **FR3: Nastavování metrik hodnocení**  
Systém vedoucímu umožní nastavit váhy, bodové rozsahy, odebrat či přidat kritéria, pokud to umožňuje zvolená kombinace přístupů.
- **FR4: Správa prací**  
Systém vedoucímu umožní zobrazit seznam všech prací vedených vedoucím a přidávat do seznamu nové registrované studenty.
- **FR5: Ohodnocení konkrétní práce**  
Systém vedoucímu umožní otevřít, prohlédnout si, upravit a ohodnotit konkrétní práci.
- **FR6: Komentování**  
Systém vedoucímu umožní ponechat odůvodnění ke každému dílčímu kritériu, bloku a celému formuláři hodnocení.
- **FR7: Modifikace finální známky**  
Systém vedoucímu umožní ovlivnit finální známku ponecháním odůvodnění a zvolením příslušné bodové / známkové odchylky. Odůvodnění se přidá do výsledného komentáře.
- **FR10: Přehled požadavků a metrik**  
Systém umožní přehledně zobrazit seznam požadavků, vah, bodových rozsahů relevantních pro danou práci.
- **FR11: Přehled dokončeného hodnocení**  
Systém umožní zobrazit již dokončené hodnocení včetně komentářů vedoucího.
- **FR12: Přidání custom požadavků**  
Systém vedoucímu umožní přidat povinné požadavky vzhledem k zaměření práce, jejichž naplnění bude ve výsledku hodnocené.

## 10.4 Architektura

Jako návrh architektury jsem zvolila jednoduchou klient–server aplikaci s pro dané aplikace typickou třívrstvou архитектурou. Pro vypracování prvního funkčního prototypu aplikace je daný typ architektury plně dostačující. Daná architektura se charakterizuje poměrně rychlým vývojem a možností navýšení škálovatelnosti, jelikož každá vrstva se vyvíjí samostatně a není závislá na ostatních vrstvách. S klient–server архитектурou jsem již měla zkušenosti během studia, což považuji za velkou výhodu v procesu implementace.

Princip fungování klient–server architektury je plně zachycen na obrázku 10.1. Obecně klient–server aplikace fungují dle přístupu požadavek–odpověď. Klienti přes svoje počítače interagují s uživatelským rozhraním aplikace, což umožňuje požadovat od serveru vykonávání různých služeb. Klient čeká, až server připraví a odešle odpověď. Poté je uživatelské rozhraní přizpůsobeno pro zobrazení obdržené odpovědi [15]. Třívrstvá architektura je nejběžnějším



přístupem k implementaci klient–server aplikací. Umožňuje jasně oddělit logiku aplikace od zobrazování a ukládání dat. Kvůli tomu se každá vrstva může vyvíjet nezávisle na ostatních.

- **Prezentační vrstva** je představená uživatelským rozhraním, které poskytuje zobrazení a přenos informací mezi koncovým uživatelem a aplikační logikou. Tato vrstva běží ve webovém prohlížeči a obvykle se vyvíjí na základě jazyku JavaScript, HTML a CSS.
- **Aplikační vrstva** představuje business logiku aplikace. Provádí hlavní výpočty spojené se zpracováním vstupních dat uživatelů z prezentační vrstvy. Obvykle aplikační server poskytuje veřejné rozhraní pro komunikaci. Proto prezentační vrstva nemusí přesně znát specifika systému. Aplikační vrstva pak komunikuje s datovou pro aktualizaci, uchování a mazání dat. Aplikační servery jsou obvykle postavené v jazycích Java, Python, PHP, Ruby.
- **Datová vrstva** bezprostředně pracuje z daty. Vystupuje místem, kde se tato data ukládají a spravují. Existuje několik druhů databází. Nejznámějšími jsou MySQL, PostgreSQL pro relační databáze, MongoDB pro dokumentově orientované databáze a Neo4j pro grafové databáze [16].

## 10.5 Použité technologie

Tato sekce se zabývá volbou vhodné technologie pro implementaci každé ze třech vrstev architektury popsanych v sekci výše. Výběr technologie je převážně založen na zkušenosti mimo frontend, které jsem získala během studia.

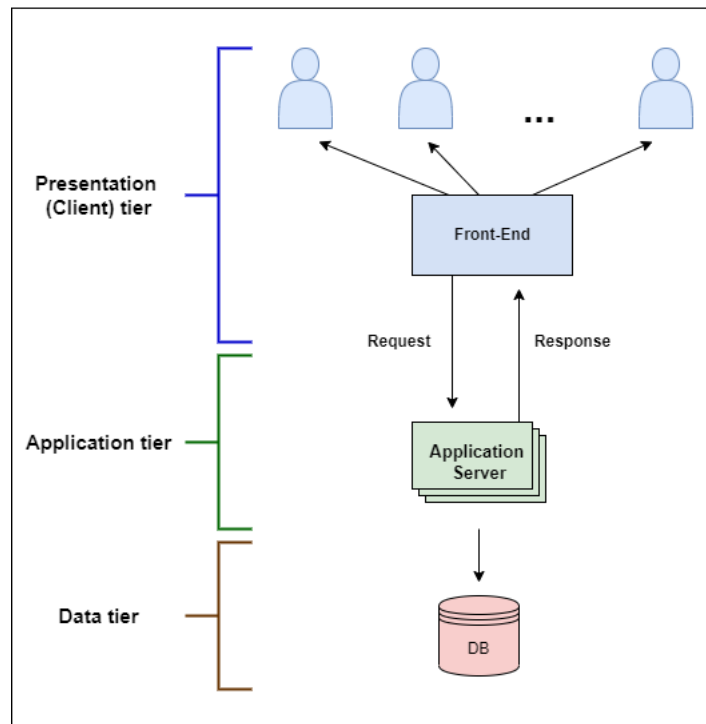
### 10.5.1 Srovnání frameworků Vue, Angular, React

S implementací prezentační vrstvy za použití JavaScriptu jsem dosud neměla žádnou zkušenost. Vyvíjet aplikace v čistém JavaScriptu dá hodně práce navíc. Existuje mnoho různých pomocných frameworků, které tuto práci zjednodušují a umožňují znovupoužití již vyvinutých komponent na různých místech aplikace. Příklady nejpopulárnějších frameworků jsou Angular, React a Vue, mezi nimiž jsem vybírala to nevhodnější řešení.

React je knihovna uživatelského rozhraní, Angular a Vue jsou plnohodnotné frontend frameworky. K vytváření frontendových aplikací je lze používat téměř zaměnitelně, ale nejsou stoprocentně stejné. Každý z těchto frameworků je založen na komponentách a umožňuje rychlé vytváření funkcí uživatelského rozhraní. Všechny tyto frameworky jsou zaměřené na tvorbu singlepage aplikace, což je základem pro vytváření progresivních webů [17]. Dále jsou uvedené základní principy každého frameworku zvlášť.

- **Angular** je vývojová platforma postavená na jazyce TypeScript. Jedná se o komponentový framework integrovaných knihoven. Tyto knihovny

### Client-server application with a three-tier architecture



Obrázek 10.1: Architektura klient-server [23]

poskytují funkce pro směrování, správy formulářů, komunikace klient-server a další. Komponenty jsou stavební bloky, ze kterých se skládá aplikace. Komponenta obsahuje třídu jazyka TypeScript s dekorátorem, šablonu HTML a styly. Angular má z těchto tří frameworků nejsložitější strukturu projektu, a protože se jedná o plnohodnotný frontend framework, spoléhá se na více konceptů [18].

- **React** je javascriptová knihovna pro vytváření uživatelských rozhraní na základě komponent. Protože logika komponent je napsána v jazyce JavaScript, framework umožňuje snadno předávat nejrůznější data v celé aplikaci a udržovat stav mimo DOM. Většinou se ale používá v kombinaci s JSX (JavaScript XML), syntaktickým rozšířením, které umožňuje vytvářet prvky obsahující současně HTML a JavaScript [20].
- **Vue** je progresivní framework, protože jeho funkce mohou být rozšířené pomocí dalších knihoven. Šablonová syntaxe Vue umožňuje vytvářet komponenty a kombinuje známé HTML se speciálními direktivami a funkcemi. Tato šablonová syntaxe je upřednostňována, i když jsou podporovány také JavaScript a JSX. Komponenty ve Vue jsou malé, samostatné a lze je opakovaně používat v celé aplikaci. Vue je o něco složitější na nastavení než React, ale snadno se učí.

Vzhledem k tomu, že dané technologie jsou víceméně podobné, při volbě vhodného frameworku pro danou práci jsem vycházela z učící křivky. V nezákladnějším případě použití je React nejméně složitý ze všech tří frameworků. Je tomu tak z důvodu, že stačí pouze importovat knihovnu a poté je možné začít psát aplikaci React s několika řádky kódu [17]. Další dva frameworky jsou komplexnější, vyžadují většího nastudování a v kontextu dané práce nepřinášejí žádné velké výhody, proto jsem zvolila právě **použití React.js**.

Za zmínku taky stojí to, že React umožňuje v rámci stejného projektu lehce importovat a využívat možnosti opensource frameworku **Bootstrap 5**. Ten je vyvinut pomocí jazyků HTML, CSS a JavaScript a skládá se ze šablon pro vytváření různých prvků, jako jsou tlačítka, formuláře, rozbalovací okna a mnoho dalších. Tento framework budu používat pro stylizaci React komponent.

### 10.5.2 Springboot framework

Pro tvorbu aplikační vrstvy jsem zvolila jazyk Java. Základem serverové části je Spring Boot Framework, který je rozšířením Spring Frameworku. Dané řešení poskytuje komplexní programovací a konfigurační model pro aplikace založené na Javě. Jádrem Springu je princip Inversion of Control. Framework obsahuje speciální kontejner, který je zodpovědný za správu životního cyklu objektu: vytváření objektů, volání inicializačních metod, konfiguraci objektů a jejich propojení pomocí anotací. Spring také poskytuje prostředky pro lehké a výkonné testování aplikací (unit, mock, integrační testy), pro práci s daty a transakci (použití knihoven pro objektově relační mapování Hibernate, JPA, EclipseLink) a mnoho další. Pro zjednodušení konfigurace projektu Spring poskytuje generování počátečních závislostí projektu na základě uživatelských předvoleb [21].

### 10.5.3 PostgreSQL

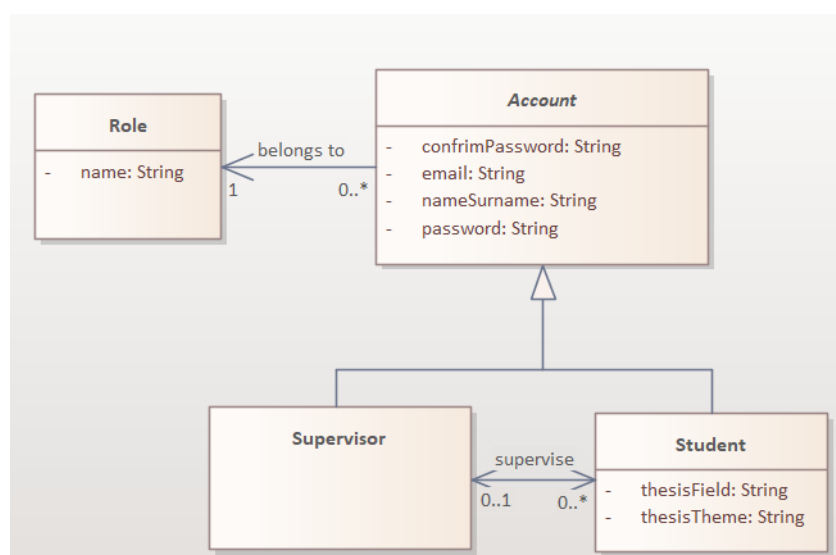
Pro uchování dat jsem zvolila databázi PostgreSQL, s níž jsem během studia mnohokrát pracovala. PostgreSQL je pokročilý relační databázový systém s otevřeným zdrojovým kódem. Aplikace je tvořena pomocí objektů (viz obrázek A.3), které se v relačních databázích ukládají přímo a pomocí vazeb (relací). PostgreSQL podporuje relační dotazování v SQL a taky nerelační JSON dotazování, které pro tuto práci není podstatné. PostgreSQL se používá jako primární databáze pro mnoho webových aplikací i mobilních a analytických aplikací, které jsou postaveny v jazyce Java i v mnoha dalších. Databáze běží na lokálním PostgreSQL Serveru, kam poskytuje přístup speciální rozhraní PgAdmin.

## 10.6 Datový model

Na základě seznamu požadavků z kapitoly 10.2 jsem vytvořila datový model aplikace třetí úrovně abstrakce. Datový model je druhem diagramu tříd,

jež určují, jak se budou data ukládat do databáze. Třetí úroveň abstrakce znamená, že vypracovaný diagram je závislý na použité technologii. Tím, že databáze deklaruje datové typy, vzniká relační schéma, které znázorňuje strukturu dat aplikace. Entity v daném modelu odpovídají příslušným Java třídám. Vazby použité pro propojení entit znázorňují vztahy mezi třídami. Šipky u vztahu popisují směr vazby, neboli objekt, který udržuje odkaz na závislou entitu [14].

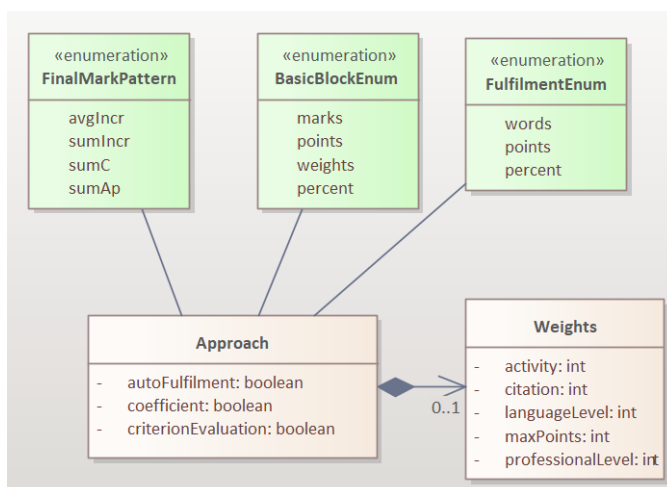
Navržený diagram se nachází v příloze na obrázku A.3. S ohledem na požadavky a úlohu práce není navržený model komplikovaný. Model lze abstraktně rozdělit na tři části. První se zabývá evidencí uživatelů v aplikaci (viz obrázek č.10.2). Z požadavků bylo jasně vidět, že v aplikaci rozlišujeme dvě základní role (vedoucí a student), které jsou představeny příslušnými entitami. Dané entity dědí od abstraktní třídy Account, jež obsahuje společné informace o uživatelích. Uživatelská role je daná další entitou, která je vázaná na abstraktní třídu. Pro každou uživatelskou roli je pak vytvořena zvláštní tabulka pro evidenci záznamů. Z diagramu je taky vidět, že student a vedoucí mají obousměrný vztah. Vedoucí může spravovat studenty, které vede. Student má možnost se podívat na účet svého vedoucího.



Obrázek 10.2: Evidence uživatelů

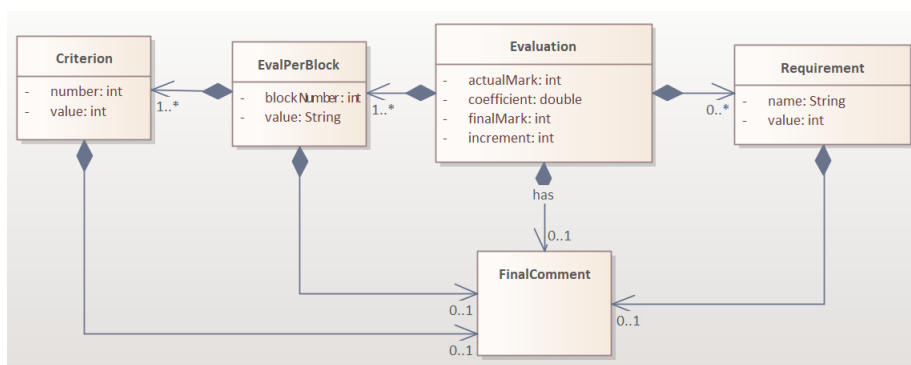
Další část diagramu popisuje evidenci přístupů k hodnocení u jednotlivých studentů. Přístup se váže na konkrétního studenta. V okamžiku založení nového studentského účtu je pro studenta defaultně vytvořen základní přístup hodnocení. Tento přístup odpovídá stávajícímu modelu hodnocení. Vedoucí studenta může daný přístup aktualizovat a nastavit dle svých preferencí. Enum třídy obsahují informace o hodnocení základních bloků (BasicBlockEnum), splnění zadání (FulfilmentEnum) a vzorec pro vypočítání výsledné známky (FinalMarkPattern). Tyto hodnoty se používají pro správnou interpretaci a zobrazení dat v rámci komponent frontendu (více v kapitole 11.2). V závislosti na zvoleném přístupu k hodnocení základních bloků existuje možnost nastavení

jejich vah, což znázorňuje entita Weights.



**Obrázek 10.3:** Evidence přístupů k hodnocení

Poslední část diagramu popisuje ukládání hotového hodnocení (viz obrázek 10.4). Hodnocení je vázáno na studenta. Vedoucí pro každého studenta vytváří hodnocení, kde postupně hodnotí každý blok v závislosti na zvoleném přístupu k hodnocení. Každé hodnocení se skládá z bloku (EvalPerBlock). Jejich seznam odpovídá blokům použitým v posudcích. Bloky se navíc skládají z dílčích kritérií definovaných v průběhu analýzy (Criterion). Vedoucí může, ale nemusí přidávat další požadavky (třída Requirements) specifické pro dané zaměření práce. Ke každé ze zmíněných entit může vedoucí ponechat komentář s odůvodněním zvoleného hodnocení.



**Obrázek 10.4:** Evidence dokončeného hodnocení

## 10.7 Sitemap

Jako sitemap se nazývá hierarchické zobrazení struktury stránek uživatelského rozhraní webové aplikace. Existuje několik druhů takových diagramů. XML sitemap je představen jako speciální soubor, který slouží převážně pro účely

vyhledávače a pro zlepšení navštěvovanosti a optimalizace webu. HTML sitemap se definuje pro větší webové aplikace a poskytuje uživatelům jasnou interaktivní strukturu hlavních a dílčích webových stránek aplikace.

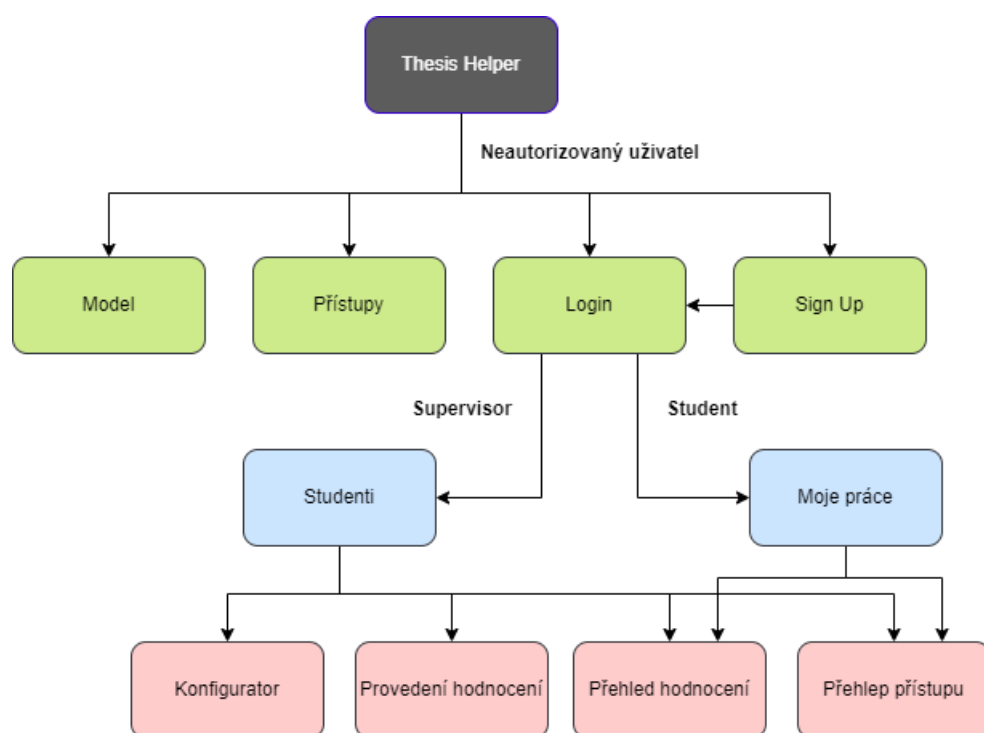
Pro účely dané bakalářské práce byla použita třetí varianta sitemap, což je vizualizovaný sitemap diagram (viz obrázek č.10.5). Takový diagram vypadá jako hierarchický strom a ukazuje informační strukturu budoucí webové aplikace. Pro lepší pochopení jsem do diagramu navíc zařadila rozdělení stránek dle aktuálně přihlášené role.

Celkem jsem návrh rozhraní rozdělila do třech úrovní. První (zelená) obsahuje stránky, které budou dostupné nepřihlášeným uživatelům. Stránka Model, jak je patrné z názvu, obsahuje tabulku modelu hodnocení, a to včetně popisu jednotlivých bloků a rozpadu na dílčí kritéria. Stránka Přístupy bude definovat možné kombinace přístupů k hodnocení základních bloků, dílčích kritérií a výsledné známky podobným způsobem, jak to bylo v souboru *approach\_combinations* elektronické přílohy B. Do stejné úrovně patří stránky pro přihlášení a registraci. Další úroveň (modrá) je závislá na přihlášeném uživateli a je dostupná pouze po úspěšné autorizaci. Pro vedoucího práce bude dostupná záložka se seznamem vedených studentů. Student uvidí záložku Moje práce, která obsahuje rozcestník pro další úroveň aplikace. Třetí úroveň hierarchie (růžová) obsahuje stránky pro interakci a přehled přístupů a samotného hodnocení. Zatímco přehled je dostupný pro obě role, interakce (tzn. konfigurátor přístupů a provedení hodnocení) je dostupná pouze pro vedoucího.

Navržená struktura umožňuje plně zahrnout do aplikace požadavky stanovené na začátku návrhu (viz sekce 10.3). Daná struktura je základem pro návrh komponent frontendu popsanych v kapitole 11.

## 10.8 Shrnutí návrhu aplikace

V této kapitole jsem rozebrala a porovнала hlavní technologie, které jsem použila v implementační části práce (viz kapitola 11). Aplikace je založena na třívrstvé architektuře klient–server, kde prezentační vrstva bude vytvořena pomocí javascriptového frameworku React, aplikační vrstva představuje hlavní serverovou logiku napsanou v Javě pomocí Spring Boot Frameworku, všechna data budou uložena v relační databázi Postgresql. Výsledky popsané v této kapitole plní cíle Dcíl5. Výběr technologií a Dcíl6. Návrh aplikace.



Obrázek 10.5: Diagram sitemap





# Kapitola 11

## Implementace aplikace

Tato kapitola popisuje hlavní implementační detaily aplikace. Začíná popisem implementace backendu, dále rozebírá realizaci frontendové části. Kapitola končí popisem komunikace klienta a serveru.

### 11.1 Backendová část

Na začátku této sekce jsou uvedeny obecné informace o zakládání projektu, popisu SpringBoot architektury a návaznosti struktury projektu na její vrstvy. Dále jsou uvedené konkrétní příklady implementace jednotlivých vrstev, a to včetně použitých knihoven, pluginů a providerů.

#### 11.1.1 Zakládání projektu

Pro implementaci jsem zvolila Spring Boot Framework. Pro založení projektu aplikace jsem využila Spring Initializr<sup>1</sup>. Je to webová aplikace, která umožňuje vygenerovat počáteční strukturu projektu, aniž by to nastavoval sám programátor. Initializr importuje závislosti zvolené při konfiguraci do *pom.xml* souboru a poskytuje základní Maven specifikaci pro build kódu. Vytvořená aplikace běží na lokálním serveru na portu 8080.

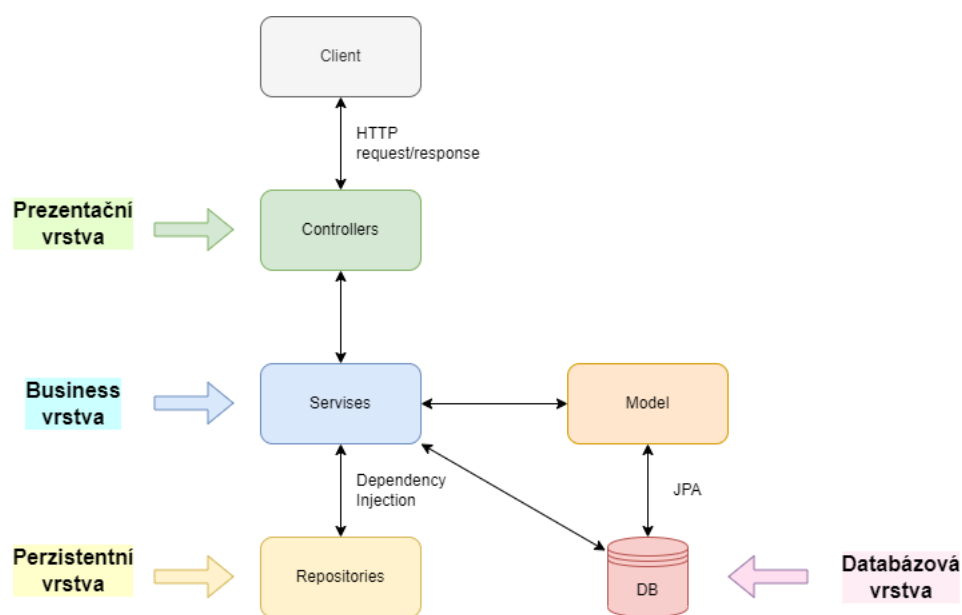
#### 11.1.2 Architektura Spring Boot

Architektura serveru je realizována v podobě vrstev. Spring Boot framework zvolený pro její implementaci se skládá ze čtyř vrstev, jak je znázorněno na obrázku 11.1<sup>2</sup>. HTTP požadavky, které odesílá klient nejprve procházejí prezentační vrstvou.

- **Prezentační vrstva** garantuje prvotní zpracování požadavků, ověřuje práva na provedení požadavku a v případě potřeby provádí konverzi mezi JSON objekty, které jsou zaslány v těle požadavku na Java objekty. V případě úspěšného zpracování danou vrstvou požadavky putují dál nižšími vrstvami.

<sup>1</sup>Odkaz na SpringInitializr: <https://start.spring.io>

<sup>2</sup>Obrázek je upraveno autorem na základě diagramu na stránkách: <https://www.geeksforgeeks.org/spring-boot-architecture>



Obrázek 11.1: Architektura Spring Boot

- **Business vrstva** představuje hlavní aplikační logiku a skládá se ze servisních tříd, které dané funkce implementují. Business zabezpečuje validaci a autorizaci.
- **Perzistentní vrstva** zahrnuje třídy, které provádějí manipulaci s databází. Tato vrstva je zodpovědná za vykonávání CRUD operací a převod Java objektu na databázové záznamy a naopak.
- **Datová vrstva** je samotnou databází, která data uchovává.

### 11.1.3 Struktura projektu

Následující tabulka 11.1 uvádí názvy hlavních projektových adresářů a také krátký popis jejich obsahu. Detailnější implementace tříd adresářů je popsána v sekcích níže, začíná sekcí 11.1.4 Kořenový adresář obsahuje hlavní SpringBoot třídu, která umožňuje spouštění aplikace. Důležitým souborem je *application.properties*, který se používá pro uchování neomezeného počtu vlastností aplikace pro spouštění v různých prostředích.

### 11.1.4 Vrstva přístupu k datům

Pro manipulaci se záznamy v databázi je třeba nejdřív zvolit vhodného Java Persistence API providera. Samotné JPA je jenom specifikací, která umožňuje objektivě relační mapování neboli konverzi Java objektů a dat v databázi. Pak existuje několik implementací daného API. Pro tuto práci jsem vybírala mezi EclipseLinkem a Hibernate. Jsou to podobné implementace, ale každá nabízí nějaké výhody a anotace navíc. Hibernate navíc je jednou z nejnávybějších

Adresář	Popis
cvut.fel.cz.thesis_helper.config	Obsahuje konfigurační třídu, která definuje zabezpečení aplikace a autorizaci požadavků.
cvut.fel.cz.thesis_helper.controller	V tomto adresáři se nachází Controller třídy, které přijmou požadavky od klienta, zpracují a odešlou je servisní vrstvě.
cvut.fel.cz.thesis_helper.dto	Adresář obsahuje třídy Data Transfer Objects (DTO) a odpovídá za jejich správnou konverzi do objektu modelu a zpátky.
cvut.fel.cz.thesis_helper.exception	Jde o třídy pro správné zpracování výjimek při vyřízení požadavků.
cvut.fel.cz.thesis_helper.model	Obsahuje třídy doménového modelu.
cvut.fel.cz.thesis_helper.repository	Rozhraní implementující JPARepository pro přístup k datovým zdrojům.
cvut.fel.cz.thesis_helper.security	Třídy, které zabezpečují správnou autorizaci a autentizaci.
cvut.fel.cz.thesis_helper.service	Servisní třídy nebo služby představující hlavní aplikační logiku.

**Tabulka 11.1:** Struktura projektu

implementací JPA s obrovskou komunitou, proto jsem se rozhodla použít právě tuto technologii.

Důležitou částí je přidání závislosti na databázi do properties souboru. Jak jsem již psala, budu používat PostgreSQL databázi. Pro konfiguraci je třeba nastavit vlastnosti pro připojení do databáze a další doplňující vlastnosti, jako je například *spring.jpa.hibernate.ddl-auto*, která definuje, jestli se budou provádět změny v databázi po uzavření/otevření spojení. Dále jsou uvedené vlastnosti použité pro danou aplikaci:

```

1 spring.datasource.url = jdbc:postgresql://localhost:5432/thesis_helper
2 spring.datasource.username=postgres
3 spring.datasource.password=password
4 spring.jpa.hibernate.ddl-auto=update
5 spring.jpa.database = postgresql
6 spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.
  PostgreSQLDialect

```

**Listing 11.1:** Konfigurace datábase a JPA

Následně jsem pokračovala definicí entit. Jejich schéma plně odpovídá doménovému modelu, který jsem vytvořila v kapitole 10.6. Pro zjednodušení práce s jejich definicí jsem použila **JPA Buddy**<sup>3</sup>, což je speciální plugin do IntelliJ. Tento plugin umožňuje snadné generování JPA entit a vazeb mezi nimi

<sup>3</sup>Odkaz na plugin JPA Buddy: <https://www.jpa-buddy.com/>

a pomáhá dodržovat osvědčených postupů při tvorbě doménového modelu. Jako další zlepšení jsem využila knihovnu **Lombok**, která pomocí anotací umožňuje redukovat šablonovitý kód. Nejčastěji jsem používala `@Data` anotaci pro generování getterů, setterů, konstruktorů s parametry a `@NoArgsConstructor` pro automatické generování konstruktoru bez parametrů. Všechny třídy modelu dědí od abstraktní třídy automaticky generovaný identifikátor pro jednoznačné ukládání do databáze.

Dále jsem se zaměřila na implementaci rozhraní **repository**. Každé vytvořené rozhraní implementuje JPA repository a je označené příslušnou anotací. Jeho nejzajímavější vlastností je možnost vytvářet implementace repository automaticky za běhu aplikace. Rozhraní obsahuje úplné API pro CRUD operace, stránkování a třídění. Při implementaci daného rozhraní je třeba specifikovat generické parametry typu třídy, se kterou pracují a její identifikátor. Ve vytvořeném rozhraní není třeba zvlášť definovat základní operace ukládání, aktualizace, hledání a mazání. Složitější metody vyhledávání, například podle různých parametrů, musejí být zvlášť specifikovány. Dále je uveden jednoduchý příklad rozhraní (viz výpis z kódu č.11.2) použitý pro práci s objektem Student. Složitější operace daná aplikace nevyžadovala.

```

1 @Repository
2 public interface StudentRepository extends JpaRepository<Student, Integer> {
3     Student findByEmail(String email);
4     Student findByApproach(Approach approach);
5 }

```

Listing 11.2: Příklad rozhraní repository

### 11.1.5 Servisní logika

Hlavní aplikační logiku představuje servisní vrstva, která poskytuje metody pro zpracování a propojení controllerů s repository. V aplikaci je tato vrstva představená pomocí rozhraní služeb a jejich implementací. Celkem bylo vytvořeno šest tříd implementujících šest různých rozhraní pro zpracování a validaci požadavků:

- **AccountService** – poskytuje metody pro práci s účty uživatelů: registrace, aktualizace účtu, hledání účtů dle emailu a vracení aktuálně přihlášeného uživatele.
- **StudentService** – registrace studentů, hledání a vracení volitelných požadavků na práci.
- **SupervisorService** – registrace vedoucích, jednoduché metody na přidání a odstranění záznamů o vedených studentech.
- **EvaluationService** – metody pro manipulaci s objektem hodnocení.
- **ApproachService** – nabízí metody pro zpracování požadavků týkajících se manipulace s objektem přístupu k hodnocení.

- **RequirementService** – metody pro přidání, odstranění a aktualizaci volitelných kritérií práce.

Každá služba přistupuje k rozhraním repository pro aktualizaci dat v případě úspěšného dokončení metody. Instance daných tříd injektuje a vytvoří Spring automaticky, pokud budou označeny anotací `@Autowired`. Anotovat lze atributy, settery nebo konstruktory. V práci jsem používala poslední možnost:

```

1  @Autowired
2  public StudentServiceImpl(StudentRepository studentRepository,
3  RoleRepository roleRepository, BCryptPasswordEncoder passwordEncoder,
4  ApproachRepository approachRepository, EvaluationRepository
5  evaluationRepository) {
6      this.studentRepository = studentRepository;
7      this.roleRepository = roleRepository;
8      this.passwordEncoder = passwordEncoder;
9      this.approachRepository = approachRepository;
10     this.evaluationRepository = evaluationRepository;
11 }

```

**Listing 11.3:** Injekce repositories do servisní třídy

Pro výkonně jednodušší posílání dat mezi vrstvy aplikace jsem využívala Data Transfer Objects (DTO), jejichž implementace se nachází v příslušném balíčku. Tyto třídy obsahují zjednodušenou strukturu datových objektů a přenáší pouze nezbytné atributy. Složitější závislosti jsou záměrně vynechány. Tyto třídy také nabízí metody pro konverzi entit modelu do DTO objektů a zpátky.

Servisní vrstva rovněž implementuje pomocné třídy pro zpracování chyb, které poskytují jejich uživatelsky pochopitelný popis.

### 11.1.6 REST API, Kontroléry

Pro komunikaci frontendu a backendu aplikace implementuje RESTful API. REST (Representational State Transfer) je architektonický styl, který umožňuje interakci s dalšími webovými službami prostřednictvím posílání HTTP požadavků. Když nějaká externí služba interaguje s rozhraním aplikace, volá speciální endpointy, které jsou představeny URL konkrétní služby nebo serveru. Požadavky jsou pak zpracovány kontrolery, což jsou třídy zodpovědné za interpretaci požadavků, přípravu modelu a vrácení odpovědí.

Kontrolery se anotují pomocí `@RestController`. Třídy kontrolerů jsou navíc označeny anotací `@RequestMapping`, která specifikuje cestu k rozhraní. Příkladem může být zpracování požadavků na registraci a autorizaci, za které nese zodpovědnost `AuthenticationController` s cestou `/thesis_helper`. Před cestou se vždy uvádí URL serveru, což v tomto případě je `http://localhost:8080`.

Třídy kontrolerů byly vytvořeny pro každou servisu s předchozí sekce. Metody kontrolerů jsou označeny jednou z možných anotací `@GetMapping`, `@PostMapping`, `@DeleteMapping` v závislosti na tom, jaký typ požadavku se zpracovává daným endpointem. Tyto anotace (stejně jako i

anotace třídy kontroleru) mohou obsahovat upřesnění cesty, pokud se liší od cesty celého kontroleru. Zároveň nemohou existovat dva endpointy, které by se nacházely na stejné cestě a mapovaly stejný typ požadavku.

Jako parametry metod se mohou posílat proměnné cesty označené anotací `@PathVariable`, pomocí kterých se nejčastěji předávají identifikátory záznamů. Anotací `@RequestBody` jsou označené parametry obsahující tělo požadavků. Obvykle to jsou DTO objekty. Třídy kontrolerů injektují servery, kterým je dál předáno zpracování požadavku. Po dokončení metody se vrací odpověď, která buď obsahuje data serializovaná do JSON, nebo jenom kód statusu HTTP odpovědi.

Správnost fungování API jsem testovala pomocí Postmanu <sup>4</sup>, což je klient-ské API, které umožňuje vytvářet, ukládat HTTP požadavky a číst jejich odpovědi.

### 11.1.7 Autorizace

Dalším krokem je zabezpečení aplikace, aby byl uživatelům umožněn přístup k těm službám, na které mají práva. Po přihlášení se pro každého uživatele na serveru generuje JWT neboli JSON Web Token. Jedná se o otevřený standard pro vytváření přístupových tokenů založených na formátu JSON, který se používá pro ověřování přenosu dat v aplikacích klient-server. Daný token šifruje důležité informace o přihlášeném uživateli: jeho přihlašovací jméno, heslo, dobu platnosti tokenu atd. Tento token se přenáší v hlavičkách HTTP požadavků a je analyzován serverem při obdržení požadavku. V případě, že se jedná o validní token, je provedená služba a vracená odpověď, v opačném případě se vrací odpověď s kódem 403.

Na začátku jsem vytvořila implementaci Spring rozhraní `UserDetails`, které poskytuje základní informace o uživateli. V mém případě je to účet uživatele a také jemu udělené pravomoci. Dále jsem implementovala rozhraní `UserDetailsService`, které má na starosti přístup k informacím o uživateli. Třída obsahuje jedinou metodu, která podle emailu hledá uživatele a v případě úspěchu vrací jeho `UserDetails`. Jinak se vyhazuje autentizační výjimka.

Dále jsem vytvořila třídu `TokenProvider`, která obsahuje metody pro vytváření a vracení tokenu uživatele při úspěšném přihlášení do aplikace, čtení tokenu z hlavičky požadavku a metodu, která validuje platnost tokenu. Třída, která pomocí `TokenProvideru` ověřuje, jestli příchozí požadavek obsahuje platný JWT, se jmenuje `TokenFilter`. Dál je třeba nakonfigurovat webové zabezpečení.

Třída `WebSecurityConfig` je opatřena anotací `@EnableGlobalMethodSecurity`, která zapíná podporu webového zabezpečení Spring Security. Třída rozšiřuje `WebSecurityConfigurerAdapter` a přepisuje jeho metodu pro nastavení některých specifík konfigurace zabezpečení webu. URL pro registraci a login je dostupné všem uživatelům, požadavky na další URL jsou nejdříve podloženy filtrací implementovaným JWT filtrem [19]. Příslušné endpointy jsou opatřeny speciální anotací `@PreAuthorize("hasAnyAuthority('SUPERVISOR')`,

<sup>4</sup>Odkaz: <https://www.postman.com>

'STUDENT')"), která definuje, jaké pravomoci musí mít uživatel pro přístup k dané službě.

## 11.2 Frontendová část

Tato sekce popisuje implementační detaily spojené s tvorbou klientské části aplikace. Sekce začíná návodem pro založení projektu. Pak jsou uvedeny základy principů programování ve frontendu pomocí frameworku React. Sekce pokračuje popisem struktury projektu a přístupem k vytváření komponent. Dále je uveden seznam použitých knihoven a jejich význam v projektu. Na konci je popsán princip autorizace uživatelů a uchování autorizačního tokenu. Pro popis komunikace klienta a serveru je vyhrazena samostatná sekce na konci kapitoly.

### 11.2.1 Založení projektu

Jak jsem již psala v kapitole 10.5.1, React je knihovnou pro vytváření javascriptových aplikací na straně klienta. React kombinuje programování aplikace v jazyce JavaScript se syntaxí podobnou HTML zvanou JSX. Pro vytvoření nového projektu je nejdříve třeba nainstalovat Node.js, což je prostředí pro spouštění kódu JavaScript mimo prohlížeč za běhu programu. Při instalaci Node.js se taky stahuje dispečer npm balíčků pro správu balíčků a spouštění skriptů. Po instalaci v příkazovém řádku je třeba provést příkaz `npm create-react-app` a uvést název projektu pro vytvoření základní struktury projektu. Spuštění aplikace se provádí pomocí příkazu `npm start`. Vytvořený projekt běží na adrese `http://localhost:3000/`.

### 11.2.2 Struktura projektu

Ve veřejném adresáři se nachází `index.html` soubor obsahující element s identifikátorem `root`, do kterého se v průběhu vývoje budou přidávat implementované komponenty. Ve složce `src` se nachází soubor `index.js`, který odpovídá za renderování komponenty `App` do kořenového elementu DOM pomocí následujícího kódu:

```

1   ReactDOM.render(
2     <React.StrictMode>
3       <Router>
4         <App/>
5       </Router>
6     </React.StrictMode>,
7     document.getElementById("root")
8   );

```

Listing 11.4: Vykreslování hlavní komponenty

Složka `src` také obsahuje:

- `App.css` soubor, který definuje rozšíření stylu elementů v případě, že styly nabízené frameworkem Bootstrap nejsou dostačující.

- Složku *components*, která obsahuje strom komponent aplikace a jejich implementaci.
- Složku *services*, jež se skládá ze souboru služeb, které implementují funkci pro správu požadavků, konverzí informačních textů vzhledem ke stavu aplikace a další pomocné metody.

### 11.2.3 Zásadní principy práce s Reactem použité v aplikaci

#### Využití komponent

Komponenty v Reactu lze vytvářet dvěma způsoby. Prvním je použití komponenty třídy, druhým je použití funkční komponenty. Komponenty třídy se obtížně testují a nepřehlednou vizuální reprezentace. Kód funkčních komponent je mnohem lépe čitelný, práce s nimi je jednodušší, a to obzvláště pro nováčky. Proto jsem pro implementaci zvolila právě funkcionální přístup. Každá komponenta vypadá následovně:

```

1  const ComponentName = () => (
2      <div className="name">
3          ....
4      </div>
5  );
6  export default ComponentName;
```

**Listing 11.5:** Šablona pro tvorbu funkcionální komponenty

Tato funkce vrací element stránky, který bude vykreslen prohlížečem na obrazovce.

Největší implementované komponenty jako Model, Approaches, Complete-Evaluation a další plně odpovídají navrženému sitemap diagramu z kapitoly 10.7. Ty mohou být rozdělené do menších celků pro zjednodušení implementace a redukování opakujícího se kódu. Tak komponenta *TextAreaComment* reprezentuje element textového komentáře, který je použit na několika místech při návržení komponenty pro vyplnění formuláře hodnocení. Pojmenování složek s komponentami je intuitivní.

#### Props

Pro přenos dat mezi rodičovskou a podřízenou komponentou se využívá princip *props*, což je jedním ze základních konceptů Reactu. Vlastnosti jsou v podstatě parametry, které se předávají komponentám. Vlastnosti jsou předány komponentě pomocí parametru *props*, který se uvádí do závorek při deklaraci funkcionální komponenty z příkladu výše. Přístup k vlastnostem je umožněn pomocí tečkové notace: *props.vlastnost*.

#### Hooks

Hooky jsou novinka v Reactu 16.8, která umožňuje používat stav a další funkce Reactu bez nutnosti psát třídy. Dále jsou uvedeny některé příklady



hooků, které jsem během práce použila, a jejich popis.

- *useState* se používá pro definici vnitřního stavu komponenty. React bude tento stav ukládat mezi vykreslováním obrazovky, hook reprezentuje pole se dvěma prvky, které obsahují aktuální hodnotu stavu a funkci pro jeho aktualizaci, tuto funkci lze použít kdekoli, například při zpracování události, příkladem může být následující řádek z práce, který ukládá informace o aktuálně přihlášeném uživateli:

```
1  const [currentUser, setCurrentUser] = useState(undefined);
2
```

**Listing 11.6:** Definice stavového hooku

- *useEffect* dává možnost provádět vedlejší efekty ve funkční komponentě, v této práci používám tento hook především ke stahování dat ze serveru,
- *useNavigate* vrací funkci, která umožňuje programovou navigaci (například po odeslání formuláře) nebo přesměrovat uživatele na přihlašovací obrazovku v případě autorizační chyby,
- *useForm* je hook definovaný uživatelem pro správu formulářů.

#### 11.2.4 Použité knihovny

Některé rozšířené možnosti pro práci s Reactem jsou dostupné pouze po instalaci dalších knihoven. Níže uvádím seznam knihoven použitých ve vývoji dané aplikace a rovněž jejich krátký popis.

- **FortAwesome** je internetová knihovna ikon a sada nástrojů, která se používá pro design a tvorbu obsahu stránek. Příkladem může být využití ikony odpadkového koše jako tlačítka pro odstranění položek ze seznamu:

```
1  import { faTrashCan } from '@fortawesome/free-regular-svg-icons'
2  ...
3  <FontAwesomeIcon icon={faTrashCan} className='btn' onClick={()=>
4  removeStudent(student)}>
```

**Listing 11.7:** Příklad využití ikony z knihovny FortAwesome

- **Axios** se používá především k odesílání asynchronních požadavků HTTP na koncové body REST. Tato knihovna je velmi užitečná pro provádění operací CRUD. Axios podporuje rozhraní API Promise. Jako příklad uvádím funkci, která odesílá požadavek pro aktualizaci hodnocení studenta. Jako parametry metody se posílá URL koncového bodu obsahující proměnnou s identifikátorem hodnocení, DTO hodnocení a hlavičky obsahující token přihlášeného uživatele:

```
1  const updateEvaluation = (evaluationId, evaluationDto) =>{
```

```

2   return axios.put(API_URL+'/${evaluationId}', evaluationDto, {
3     headers:
4       user?{"Authorization": 'Bearer_${user.token}':{}});
5   }

```

Listing 11.8: Příklad volání REST API

Po volání se tato funkce vykonává asynchronně. Pomocí *.then* se definuje další chování po úspěšném nebo neúspěšném dokončení volání.

- **Bootstrap** poskytuje základní stylizaci elementů stránky.
- **React Router** je standardní směrovací knihovna pro React. React Router je bohatý na navigační komponenty a umožňuje uspokojení navigačních požadavků ve větších a složitějších aplikacích React. V této práci je použit pro naplnění požadavků daných sitemap diagramem (viz kapitola 10.7) za zachování principu Single Page Web. *App* komponenta obsahuje následující kód, který definuje cesty k větším komponentám aplikaci, což odpovídá dříve zmíněnému diagramu stránek webu:

```

1   <Routes>
2     <Route path="/thesis-helper" element={<WelcomePage />} />
3     <Route path="/model" element={<Model />} />
4     <Route path="/approaches" element={<Approaches />} />
5     <Route path="/students" element={<Students />} />
6     <Route path="/students/:studentId/configurator" element={<
7       Configurator />} />
8     <Route path="/:studentId/evaluation" element={<
9       CompleteEvaluation />} />
10    <Route path="/students/:studentIdParam/evaluation" element={<
11      CompleteEvaluation />} />
12    <Route path="/login" element={<Login />} />
13    <Route path="/register" element={<Register />} />
14    <Route path="/students/:studentId/summary/:approachId" element
15    ={<Summary />} />
16    <Route path="/evaluation-overview/:evaluationId" element={<
17      EvaluationOverview/>} />
18  </Routes>

```

Listing 11.9: Komponenty navigace

Aby směrování bylo umožněno, *App* komponenta musí být umístěna do elementu `<Router></Router>` importovaného z knihovny React Router.

- **Hook Form** a **Validation** pomáhají spravovat a validovat formuláře v Reactu.

### 11.2.5 Autorizace

Pro zaručení zobrazení správných obrazovek klientovi (vzhledem k jeho právomocem) je třeba implementovat jednoduchou proceduru autorizace. K tomu

slouží komponenta *Login*, která obsahuje přihlašovací formulář. Při pokusu o přihlášení se posílá dotaz na server pomocí *AuthService* implementující funkci *login*:

```

1 const login = (data) => {
2   return axios
3     .post(API_URL + "/auth", data)
4     .then((response) => {
5       if (response.data.token) {
6         localStorage.setItem("user", JSON.stringify(response.data));
7       }
8       return response.data;
9     });
10 };

```

**Listing 11.10:** Vyřízení autorizačního požadavku

Funkce přijímá parametr *data*, jež obsahuje JSON objekt s přihlašovacími údaji uživatele (email a heslo). V případě, že přihlašování proběhlo úspěšně, server vrací odpověď obsahující informace o uživateli a jeho autorizační token JWT (viz sekce 11.1.7). V tento okamžik je třeba rozhodnout, kam ukládat token přihlášeného uživatele pro jeho využití při posílání dalších požadavků na server.

Existují dvě možnosti ukládání tokenu na straně klienta. První je použití Cookie Storage. Pro uchování tokenu v cookies server musí poslat odpověď s HTTP hlavičkou *SetCookie*. Pro předání přístupového tokenu zpět rozhraním API na stejné doméně by prohlížeč automaticky zahrnul cookie do požadavku. Druhou možností je uchování tokenu ve Web Storage, kterou jsem zvolila pro realizaci ve své aplikaci. V případě úspěšného volání API je odpověď obsahující autorizační token uchována pomocí **localStorage API**. Navzdory tomu, že posílání požadavků na server pomocí knihovny *Axios* je zcela asynchronní, tuto funkci provádíme synchronně, aby mohla počkat na odpověď serveru a přeměřovat uživatele na obrazovku, která odpovídá jeho roli:

```

1 await AuthService.login(data).then(
2   () => {
3     if (AuthService.getCurrentUserFromLocalStorage().role==='
4     SUPERVISOR')
5       navigate("/students");
6     else navigate("/evaluation");
7     window.location.reload();
8   },
9   (error) => {
10    console.log(error);
11  });

```

**Listing 11.11:** Synchronní volání RESTu

Musíme ještě zajistit mazání tokenu v případě, že server při dalších požadavcích vrátí autorizační chybu. To se dělá kvůli tomu, že *localStorage* uchovává záznamy poměrně dlouho a k jejich mazání dochází třeba v případě přeplnění paměti. Mazání se provádí pomocí metody *removeItem(item)* pro smazání

konkrétní položky nebo `.clear()` pro vyčištění celého úložiště.

### ■ 11.2.6 Seznam vytvořených obrazovek a návaznost na komponenty

Tato sekce popisuje obsah webových stránek a uvádí seznam použitých komponent. Obrazovky jsou znázorněny v elektronické příloze B v souboru *priloha1 > vytvorene\_obrazovky\_aplikace*. Názvy odpovídají názvům z následujícího seznamu.

- **Navigační a spodní panel** – každá stránka nahoře obsahuje webovou navigaci představenou navigačním panelem a spodní panel s informacemi o vývojáři. Implementace komponent se nachází ve složce *HeadersAndFooters* v souborech *AppMainHeader.js* a *Footer.js*. Obsah navigačního panelu se mění v závislosti na aktuálně přihlášeném uživateli.
- **Model** – tato obrazovka obsahuje tabulku s popisem jednotlivých bloků posudku a rozpadem základních bloků na dílčí kritéria a je dostupná pro všechny uživatele. Implementace komponenta *Model.js*.
- **Přístupy** – záložka se skládá z kolapse sekcí obsahujících možnosti hodnocení všech bloků posudku a taky stupnici pro konverzi známky z jednoho přístupu do jiného. Záložka je dostupná pro všechny uživatele a je implementovaná pomocí komponenty *Approaches.js*.
- **Login** – stránka obsahující přihlašovací formulář, implementace komponent a *Login.js*
- **Registrace** – stránka obsahující registrační formulář. Formulář se dynamicky mění v závislosti na zvolené uživatelské roli. Implementace komponenty *Register.js*
- **Studenti** – stránka obsahující informace o profilu vedoucího s možností jeho editace (komponenta *Profile.js*), tabulku se seznamem studentů, jejichž práce uživatel vede (komponenta *Students.js*), a tlačítko, které vyvolá modální okno pro přidání do tabulky nového studenta (komponenta *Modal.js*). Tato stránka je dostupná pouze pro vedoucího. U každého studenta ze seznamu je možnost upravit přístup hodnocení, prohlédnout si nakonfigurovaný přístup, vyplnit hodnocení a prohlédnout si dokončené hodnocení. Každé funkcionalitě odpovídají samostatné komponenty popsané níže.
- **Konfigurator** – úprava přístupu hodnocení začíná prvním krokem konfiguratoru odpovědmi na otázky z nabídky. Tato komponenta je implementovaná v souboru *BasicSelect.js*. V závislosti na zvolených odpovědích se uživateli zobrazují další komponenty konfiguratoru: *Requirements.js* pro přidání volitelných požadavků na práci, *Weights.js* pro konfiguraci vah. Zobrazení správné komponenty řídí rodičovská komponenta *Configurator.js*.

- **Přehled zvoleného přístupu** – tato stránka obsahuje tabulku s popisem přístupu, který uživatel nakonfiguroval pro daného studenta. K této stránce má přístup jak vedoucí, tak i student. Implementace komponenta *Summary.js*.
- **Hodnocení** – provedení hodnocení práce je vždy upraveno vzhledem ke zvolenému přístupu. Začíná hodnocením bloku Zadání, dál jsou postupně hodnoceny všechny bloky posudku. Zobrazení formuláře pro hodnocení řídí komponenta *CompleteEvaluation.js*. Komponenta *Assignment.js* obsahuje formulář hodnocení bloku Zadání, *Fulfilment.js* bloku Splnění zadání, *BasicBlock.js* formulář hodnocení základních bloků. *FinalMark.js* formulář pro potvrzení nebo úpravu finální známky hodnocení. *PercentInput.js*, *TextAreaComment.js* pomocné komponenty formulářů pro zachycení uživatelského vstupu.
- **Přehled dokončeného hodnocení** – tato stránka obsahuje tabulku s vyplněným hodnocením vedoucího. K této stránce má přístup jak vedoucí, tak i student. Implementace komponenty *EvaluationOverview.js*.

## 11.3 Komunikace backendu a frontendu

Prohlížeč provádí odesílání HTTP požadavků na server, přijímá odpovědi, zpracovává přijatá data a používá je k vykreslení stránky. Veškerá komunikace ze strany uživatele probíhá přes jeho prohlížeč. Posílání požadavků na jedné doméně není žádným způsobem omezené. Problém nastává, když se požadavky odesílají napříč různými doménami, čímž se charakterizuje komunikace dané aplikace. K tomu, abychom umožnili podobnou komunikaci, musíme vyřešit problém CORS politiky.

CrossOrigin Resource Sharing (CORS) je mechanismus založený na hlavičkách HTTP, který prohlížečům umožňuje identifikovat, který požadavek pochází ze seznamu povolených domén, a zároveň odfiltrovat neznámé požadavky [22]. To funguje tak, že prohlížeče provedou předběžný požadavek obsahující metodu *OPTIONS* na server s cizí doménou. To se dělá pro ověření, zda server povolí požadavek provést. V tomto předběžném požadavku prohlížeč odesílá informace o metodě HTTP a hlavičkách, které budou později použity v reálném požadavku. Server buď provedení povolí a bude čekat na skutečný požadavek, nebo předběžný požadavek zamítne s chybou *Access to XMLHttpRequest has been blocked by CORS policy*.

Pro umožnění provedení požadavků z jiných domén jsem do konfigurace zabezpečení (*WebSecurityConfig* soubor) přidala řádek *http.cors()*, který dovolí řídit posílání crossdoménových požadavků. Do tříd kontrolerů jsem přidala anotaci:

```
1 @CrossOrigin(allowedHeaders = {"authorization","content-type"}, origins = "
  http://localhost:3000")
```

**Listing 11.12:** Anotace pro cross doménové volání

Ta specifikuje název domény, požadavky které mohou být zpracované a povolené hlavičky.

V této kapitole jsem popsala všechny důležité detaily spojené s implementací jak serverové, tak i klientské strany aplikace. Daná aplikace je hlavním výstupem práce, který v následující kapitole bude podroben uživatelskému testování. Kód aplikace se nachází v elektronické příloze B (viz složka *příloha2*) a v GitHub repositářích. Odkazy na repositáře, seznam použitých knihoven a instalační příručka se nachází v elektronické příloze B (viz soubor *instalacni\_přirucka*). Tato kapitola naplňuje cíl Dcíl7. Implementace.

## Kapitola 12

### Uživatelské testování

Tato kapitola popisuje uživatelské testování aplikace pro hodnocení závěrečných prací, její implementace je obsažena v kapitole 11, a to stejně jako podklady pro vyhodnocení celého výzkumného projektu. Na začátku kapitoly jsou stručně popsány cíle testování a příprava na testování, včetně vytvoření testovacích scénářů a dotazníků. Na konci kapitoly jsou shrnuty hlavní výsledky testování a jsou nastíněny návrhy na zlepšení aplikace i použitého modelu hodnocení.

#### 12.1 Ověření aplikace a vytvořených přístupů

Pro testování byl zvolen přístup uživatelského testování. Jedná se o typ testování, které se provádí přímo s koncovými uživateli vytvořeného systému s cílem ověřit, zda je vhodný pro zamýšlené použití. Existují i další typy testování aplikací – například funkční testování, zátěžové testování, testování konfigurace, testování zabezpečení. Z názvů je hned jasné, co tyto druhy znamenají. Pro testování této aplikace bylo zvoleno uživatelské testování, protože hlavním cílem nebylo najít jen nedostatky systému související s implementací, ale spíše ověřit, zda navržený způsob hodnocení a model přispívá k lepšímu, pohodlnějšímu a rychlejšímu hodnocení.

Hlavní cíle, s nimiž bylo testování prováděno, lze shrnout v následujících bodech:

- dostat zpětnou vazbu na obecný záměr aplikace a zjistit, jestli navržené řešení dává smysl a jestli by bylo vhodné pro použití ve studijním procesu
- zjistit, jestli záměr tohoto typu aplikace má potenciál pro další výzkum;
- ověřit, jestli všechny záložky aplikace (viz 11.2.6) byly pro uživatele navrženy pochopitelně a vhodně pro rychlé a snadné využití;
- zjistit přístup, neboli kombinaci přístupů hodnocení jednotlivých bloků, která se zdá pro vedoucí být nejvhodnější pro použití při hodnocení práce;
- dostat obecné návrhy, náměty, podněty na další zlepšení / úpravu aplikace a navržených kombinací přístupů.

## 12.2 Příprava k testování

Příprava na testování se u hodnotitelů a studentů mírně lišila, a to kvůli odlišnému průběhu testování (více v kapitole 12.3). Před testováním s hodnotiteli byly provedeny následující kroky:

- každému účastníkovi byl vytvořen testovací účet pro práci s aplikací, aby neztrácel čas registrací do testu, a mohl se ihned soustředit na hodnocení;
- byl vytvořen jednoduchý testovací scénář průchodu aplikací (viz sekce 12.2.1);
- každý respondent byl požádán, aby předem zaslal posudky práce, kterou by chtěl v aplikaci vyhodnotit;
- byl vytvořen dotazník, na který každý respondent odpovídal po dokončení testování (více v části 12.2.2).

Před ukázkou aplikace studentům a absolventům:

- byla vytvořena krátká prezentace vysvětlující cíle mé bakalářské práce;
- byl připraven dotazník, který každý student vyplnil po setkání.

### 12.2.1 Návrh testovacího scénáře

Aby bylo zajištěno, že testování bude strukturované a proběhne bez prodlení, byl testerům zadán scénář, jehož jednotlivé kroky jsou popsány v této sekci níže. Tento scénář je pouze doporučením. Testeři se mohli od scénáře odchýlit, jak uznají za vhodné. Téměř všichni testující využili možnosti provést hodnocení několikrát a také několikrát revidovat zvolený přístup k hodnocení, aby zjistili, který z nich bude pro ně nejvhodnější. Kroky scénáře:

1. Přihlášení – tester se přihlásí do aplikace pomocí předem daných údajů.
2. Seznamte se s modelem hodnocení a rozpadem bloku na kritéria – tester přejde do záložky Model a přečte zobrazenou tabulku.
3. Seznamte se s navrženými přístupy k hodnocení prací – tester přejde do záložky Přístupy a seznámí se s uvedenými sekcemi.
4. Zvolte si vhodný přístup k hodnocení – tester se nachází na stejné záložce a po seznámení si zvolí vhodný přístup hodnocení první práce.
5. Přidejte studenta, jehož práci chcete ohodnotit – tester přejde do záložky Studenti a přidá předem registrovaného studenta do svého seznamu.
6. Proveďte konfiguraci zvoleného přístupu k hodnocení – tester pomocí tlačítka Upravit ve sloupci Přístup upraví přístup pro hodnocení práce přidaného studenta.



7. Proveďte hodnocení prací – tester pomocí tlačítka Ohodnotit ve sloupci Hodnocení provede hodnocení práce přidaného studenta.
8. Zobrazte právě dokončené hodnocení – tester pomocí tlačítka Ukázat ve sloupci Hodnocení zobrazí právě dokončené hodnocení.
9. Odhlášení – tester se odhlásí z aplikace.

### 12.2.2 Popis vytvořených dotazníků pro zpětnou vazbu

Byly vytvořeny celkem dva dotazníky, jeden pro studenty a druhý pro hodnotitele.

Dotazník pro studenty se skládá ze dvou částí. První část se týká obecných statistických údajů o studentech. Druhá část je zaměřena na zjištění reakce studentů na navržené řešení.

Pro hodnotitele je dotazník rozšířenější na základě toho, že mohou v aplikaci provádět více interakcí. Proto byla druhá část ze studentského dotazníku rozdělena na dvě samostatné části. První z nich byla zaměřena na sběr názorů na samotnou aplikaci a druhá na sběr zpětné vazby ohledně obecné myšlenky téhle práce.

Odpovědi získané díky dotazníkům jsou hlavním výsledkem testování. V další sekci jsou uvedeny hlavní postřehy testerů. Všechny odpovědi a otázky jsou k nalezení v elektronické příloze B v souborech *dotaznik\_hodnotitele* a *dotaznik\_studenti*.

## 12.3 Průběh testování

Test byl rozdělen do dvou částí. V první části pracovali s aplikací pracovníci univerzity, kteří mají s hodnocením prací přímé zkušenosti. Někteří mají více než 10 prací ročně, jiní mnohem méně. Cílem výběru testerů bylo otestovat aplikaci na různých typech uživatelů a ujistit se, že bude užitečná pro všechny.

Za zmínku také stojí, že se testování účastnili nejen hodnotitelé ze SITu, ale i hodnotitelé z jiných oblastí Fakulty elektrotechnické. Testování s hodnotiteli bylo provedeno jako osobní setkání, kde jsem je nechala pracovat s aplikací pomocí navrženého testovacího scénáře (viz sekce 12.2.1). V průběhu práce jsme vedli diskusi ohledně použitelnosti navrženého řešení.

Druhou částí testování byla skupinová prezentace aplikace studentům a absolventům technické univerzity. Cílem bylo získat jejich zpětnou vazbu o tom, zda by pro ně byl přístup k údajům dostupným v aplikaci užitečný v jejich bakalářském projektu.

Testu se zúčastnilo celkem 6 hodnotitelů (včetně mého vedoucího) a 5 studentů (včetně absolventů). Všichni respondenti na konci testování dostali odkaz na dotazník (viz sekce 12.2.2). Výsledky testování jsou zahrnuté do sekce 12.4.

## 12.4 Výstupy testování

Výstupy jsou rozdělené dle dvou uživatelských skupin, které se testování zúčastnily.

### 12.4.1 Zpětná vazba od hodnotitelů

- Hodnotitelé neměli připomínky k popisu modelu hodnocení (záložka aplikace "Model") a souhlasili s návrhem rozpadu základních bloků na dílčí kritéria – potvrzení Úvahy 2. Hlavní bloky modelu a 3. Rozpad hlavních bloků na dílčí kritéria.
- Většina hodnotitelů souhlasila s návrhem záložky aplikace Přístupy. Jeden měl připomínku, že popis je příliš dlouhý a pro naučení a pochopení by potřeboval více času, což by snížilo jeho ochotu začít používat navrženou aplikaci – upřesnění Úvahy 12. Možnost konfigurace modelu.
- Hodnotitelé se neshodli na tom, který přístup pro hodnocení bloku Splnění zadání by byl pro ně nejvíce použitelný. Hlasy byly rovnoměrně rozdělené mezi slovním, procentuálním a automatickým přístupem. Bodový přístup se nezalíbil žádnému testerovi – upřesnění Úvahy 6. Celkové a blokové hodnocení.
- Pro hodnocení základních bloků by hodnotitelé nejraději používali hodnocení pomocí vah, procent a automatické hodnocení pomocí dílčích kritérií – upřesnění Úvahy 6. Celkové a blokové hodnocení.
- Obecný záměr testeři považují za smysluplný. Doladěná aplikace by podle jejich názorů vedla k objektivnějšímu hodnocení závěrečných prací a byla by přínosná pro méně zkušené hodnotitele.
- Hodnotitelé by uvítali možnost exportování PDF posudků z aplikace, možnost upravovat finální hodnocení přímo v souhrnné tabulce a přidání nápověd v aplikaci.
- Hodnotitelé by stoprocentně běžně využívali možnost přidání dílčích kritérií – potvrzení Úvahy 7. Volitelná kritéria pro odbornou úroveň.
- Hodnotitelé nechtějí, aby nevyhovující hodnocení za jeden blok mělo definitivní dopad na výslednou známku – vyvrácení Úvahy 10. Dopad nevyhovujícího blokového hodnocení na výslednou známku.

### 12.4.2 Zpětná vazba od studentů

- Studenti by rozhodně uvítali rozpad modelu na dílčí kritéria.
- Chtěli by vidět větší strukturaci tabulky nazvané Přehled zvoleného přístupu.
- Chtěli by mít možnost „odsouhlasit“ přístup vedoucího.

- Jeden ze studentů se přímo vyjádřil následovně: „Tato aplikace by byla užitečná při psaní práce. Kdybych ji používala a měla možnost podívat se na způsob hodnocení svého vedoucího, tak bych s ním pravděpodobně měla více konzultací, protože bych věděla, že to má vliv na mé závěrečné hodnocení.“
- Další student uvedl: „Koncept se mi líbí, především to, že student může mít od počátku představu, jakým stylem bude hodnocen. Dále se mi líbí to, že je explicitně definována subjektivita hodnocení, což může přinést přesnější obraz o kvalitě hodnocené práce.“

Zpětná vazba obdržená v průběhu testování a odpovědi na dotazníky naplňují Dcíl8 stanovený během definice projektu výzkumu. Poslední cíl Dcíl9. Odvozování závěrů je naplněn ve finální kapitole 13 ve formě vyhodnocení projektu výzkumu, kde jsem krátce shrnula odvedenou práci a stanovila jsem možné návrhy na rozšíření a zlepšení vytvořeného řešení na základě obdržených výsledků testování.



## Kapitola 13

### Vyhodnocení projektu výzkumu

Tato kapitola se zabývá vyhodnocením projektu výzkumu. Na začátku je krátce popsán aktuální stav projektu a uvedeno vyhodnocení stavu úvah a naplněnosti cílů. Následně se uvádí možnosti dalšího rozvoje projektu.

#### 13.1 Aktuální stav

V rámci projektu výzkumu bylo dosaženo následujících výstupů:

- na základě odvedené analýzy ve výzkumné části projektu byl definován seznam úvah, který se stal základem pro rozšíření stávajícího modelu hodnocení závěrečných prací v programu SIT a pro návrh pomocného nástroje pro jejich vyhodnocení;
- byly vytvořeny první dva prototypy rozšíření modelu (viz elektronická příloha soubor *prototypes\_v1*);
- byly finalizovány kombinace přístupů hodnocení dílčích částí modelu, které vzešly po analýze a obdržení zpětné vazby na vytvořené prototypy (viz elektronická příloha soubor *approach\_combinations*);
- byla vytvořena první verze aplikace, která implementuje navržené přístupy (zdrojový kód se nachází v elektronické příloze B ve složce *příloha2*);
- výstupy práce byly podrobeny uživatelskému testování na vybraných uživateli.

#### 13.2 Interpretace výsledků

Během rozvoje projektu výzkumu bylo celkem definováno 12 úvah, které byly postupně vyhodnoceny a zohledněné v navrženém řešení. Níže uvádím tento seznam a popisují jejich finální stav.

- **Úvaha 1. Velká míra abstrakce modelu – potvrzeno**

Po uskutečnění pohovorů s absolventy a hodnotiteli úvaha byla označena za potvrzenou. Po uživatelském testování se stav úvahy nezměnil (viz kapitoly 6, 8, 12.4).

■ **Úvaha 2. Hlavní bloky modelu – potvrzeno**

Respondenti potvrdili, že by chtěli, aby základní bloky navrženého řešení byly pro ně známé a odpovídaly stávajícímu modelu.

■ **Úvaha 3. Rozpad hlavních bloků na dílčí kritéria – potvrzeno**

Uživatelské testování potvrdilo, že rozpad základních bloků na dílčí kritéria přispívá lepšímu pochopení záměru práce a rychlejšímu hodnocení (viz kapitola 12.4).

■ **Úvaha 4. Přidělení vah za bloky a dílčí kritéria – částečně potvrzeno**

Po provedení pohovorů s hodnotiteli, kde respondentům byly ukázány první prototypy modelu hodnocení, bylo rozhodnuto umožnit nastavování vah pouze za základní bloky, nikoliv za dílčí kritéria (viz kapitola 8).

■ **Úvaha 5. Průběžné monitorování práce – odloženo**

Úvaha vzešla po rešerše existujících přístupů hodnocení různých univerzit a byla odložena, protože se netýká záměru práce (viz kapitola 4).

■ **Úvaha 6. Celkové a blokové hodnocení – částečně potvrzeno**

Úvaha definuje jeden postup k hodnocení práce, který byl rozšířen o další možnosti v průběhu celého výzkumu.

■ **Úvaha 7. Volitelná kritéria pro odbornou úroveň – potvrzeno**

Hodnotitelé se shodli, že by rozhodně využili možnost přidávání dalších kritérií vzhledem ke zaměření a zadání práce (viz kapitola 12.4).

■ **Úvaha 8. Přístup oponenta k modelu – vyvráceno**

Navzdory tomu, že jednou z připomínek od studentů při prvních pohovorech byl požadavek na poskytnutí upřesněného přístupu k hodnocení oponentovi, hodnotiteli se s tím neztotožnili. Hodnotitelé zřejmě v řešení této otázky mají větší prioritu, a proto tato úvaha byla zatím vyvrácena.

■ **Úvaha 9. Průnik náročnosti zadání a jeho splnění – částečně potvrzeno**

Jelikož se v průběhu výzkumu hodnotitelé jednoznačně neshodli na jednom používaném přístupu hodnocení, úvaha byla označena za částečně potvrzenou.

■ **Úvaha 10. Dopad nevyhovujícího blokového hodnocení na výslednou známku – vyvráceno**

Hodnotitelé by chtěli i při nevyhovujícím hodnocení jednoho z bloků mít možnost pro navýšení známky (viz kapitola 8).

■ **Úvaha 11. Vliv kritérií odborné úrovně na splnění zadání – částečně potvrzeno**

Objasnění je stejné jako v Úvaze 9.

■ **Úvaha 12. Možnost konfigurace modelu – částečně potvrzeno**

Hodnotitelé, kteří vedou málo prací a jsou zvyklí na stávající model, by konfigurátor nevyužívali. Opačná situace je u hodnotitelů, kteří mají na starosti větší počet studentů.

Všechny potvrzené a částečně potvrzené úvahy byly zohledněné při návrhu možných kombinací přístupů k hodnocení.

Výsledky práce naplňují hlavní cíle stanovené na začátku výzkumu (viz kapitola 3.3):

1. **Návrh rozšíření modelu** – naplněno během celé výzkumné části projektu, výstupem je soubor *approach\_combinations*.
2. **Implementace první verze modelu** – naplněno v první aplikační části projektu (viz kapitoly 10, 11). Výstupem je kód aplikace, který se nachází v souboru *priloha2*.
3. **Testování a interpretace výsledků** – naplněno v rámci posledních dvou kapitol aplikační částí projektu (viz kapitoly 12 a 13).

## 13.3 Možnosti dalšího rozvoje projektu

Níže uvedený seznam obsahuje návrhy na změny nebo vylepšení aplikace i vytvořených kombinací. Seznam byl vytvořen po diskuzi s vedoucím bakalářské práce na základě uživatelského testování a předchozí analýzy. Tento seznam může být použit jako základ pro vypracování této práce na magisterské úrovni mnou nebo jiným studentem.

1. Zapracovat na připomínkách uživatelů ohledně vzhledu aplikace (viz kapitola 12).
2. Provést detailnější analýzu požadavků označených v diagramu A.2 modrou barvou (tzn. možné návrhy na zlepšení) a provést případné rozšíření aplikace. Tyto požadavky hlavně zahrnují: rozšíření pohledu studenta o možnost sledování a odhadování své průběžné známky, přidání modulu pro plánování práce a času stráveného na projektu, přidání možnosti exportu PDF posudku, jež obsahuje hodnocení vytvořené v aplikaci.
3. Zamyslet se nad zpřístupněním hodnocení posudku oponentovi.
4. Předložit aplikaci většímu vzorku uživatelů, aby se eliminovaly nepoužitelné přístupy k hodnocení bloků posudků. Tento bod by rovněž umožnil provést analýzu hodnocení vytvořených a uložených v databázi s cílem upřesnit seznam povinných požadavků na práci vzhledem k jejímu zaměření.

5. Vytvořit portfolio prací neboli šablony, ve kterých lze definovat doporučený přístup pro hodnocení prací vzhledem k jejich zaměření. Při definici vycházet z již ohodnocených prací uložených na DSpace a zpětné vazby od hodnotitelů.
6. Výstupy projektu dát k dispozici veřejnosti s cílem získání kvalitnějších výstupů a potvrzení navržených přístupů.





## Závěr

Daná práce se zabývala analýzou existujícího hodnocení závěrečných prací studentů posledních ročníků bakalářského studia FEL SIT a návrhem modelu a přístupu k hodnocení, který by vyhovoval co největšímu počtu požadavků uživatelů. Celá práce probíhala ve formě výzkumného projektu, protože nejlépe odpovídal zadání a požadavkům na vypracování. Práce se skládala ze dvou částí: výzkumné a aplikační.

Ve výzkumné části byla provedena rešerše existujících modelů hodnocení různých univerzit. Po analýze byly stanovené první úvahy na rozšíření stávajícího modelu posudků použitého pro hodnocení prací programu SIT. Tyto úvahy byly předloženy ke zpětné vazbě studentům a hodnotitelům, což poskytlo základnu pro vytvoření rozšíření celého modelu hodnocení a navržení možných kombinací přístupů k hodnocení tohoto modelu.

Výstupy první části bakalářské práce byly využity při návrhu v aplikační části. Tato část zahrnovala definici požadavků a dalších specifikací na budoucí aplikaci a výběr vhodných technologií pro její realizaci. Po implementaci byla vytvořena aplikace architektury klient–server za využití frameworků Spring-Boot a React. Vytvořená aplikace byla podrobená návaznému uživatelskému testování. Výsledky testování splnily očekávání a ukázaly, že aplikace a obecný záměr jsou použitelné a mají potenciál pro další rozvoj.

Odvedená práce naplňuje cíle stanovené na začátku výzkumu (viz sekce 3.3), které byly vyhodnocené v předchozí kapitole.

Realizace této práce mi dala možnost vyzkoušet v praxi kompletní provedení jednoduchého výzkumu a rozšířit své analytické a komunikační poznatky. Z obecného hlediska zmapování různých přístupů k hodnocení a rozšíření stávajícího modelu pomůže studentům a hodnotitelům při plnění a hodnocení závěrečných projektů. Tato práce má potenciál pro další rozvoj, jehož možnosti byly popsány při vyhodnocení projektu v kapitole č.13.





## Literatura

- [1] Chapter 2. Concepts and definitions for indentifying RD. In: Frascati Manual 2015 [online]. Paris: OECD Publishing, 2015, s. 4379 [cit. 20211008]. ISBN 9789264239012. Dostupné z: <https://www.oecd.org/sti/frascati-manual-2015-9789264239012-en.htm>
- [2] BENEŠ, Ladislav Dr. Zásady pro zpracování výzkumu v bakalářských a diplomových pracích [online]. Praha: UK ETF, 8. 10. 2020 [cit. 20211008]. Dostupné z: [http://www.etf.cuni.cz/studium/Zasady\\_vyzkum\\_2014.pdf](http://www.etf.cuni.cz/studium/Zasady_vyzkum_2014.pdf)
- [3] VOJTÍŠEK, Petr. Výzkumné metody: Metody a techniky výzkumu a jejich aplikace v absolventských pracích vyšších odborných škol [online]. F. Praha: Vyšší odborná škola sociálně právní, 2012 [cit. 20211020]. ISBN 97880905109-37. Dostupné z: [https://skoly.praha.eu/files/=84121/Skripta%20V%C3%BDzkumn%C3%A9\\_metody.pdf.%20Vy%C5%A1%C5%A1%C3%AD%20odborn%C3%A1%20C5%A1kola%20soci%C3%A1ln%C4%9B%20pr%C3%A1vn%C3%AD,%20Praha](https://skoly.praha.eu/files/=84121/Skripta%20V%C3%BDzkumn%C3%A9_metody.pdf.%20Vy%C5%A1%C5%A1%C3%AD%20odborn%C3%A1%20C5%A1kola%20soci%C3%A1ln%C4%9B%20pr%C3%A1vn%C3%AD,%20Praha)
- [4] Základní pojmy výzkumu a vývoje v OECD a EU. Rada pro výzkum, vývoj a inovace [online]. Česká Republika: Úřad vlády České Republiky, 2015 [cit. 20211020]. Dostupné z: <https://www.vyzkum.cz/FrontClanek.aspx?idsekce=932>
- [5] MOLNÁR, Zdeněk. Pokročilé metody vědecké práce. [Zeleneč]: Profess Consulting, 2012 [cit. 20211109]. Věda pro praxi (Profess Consulting). ISBN 9788072590643.
- [6] JUŘENÍKOVÁ, Petra. Metodika ke zpracování závěrečné práce pro vybrané nelékařské zdravotnické obory: Kvantitativní výzkum [online]. Brno, 2019 [cit. 20211109]. Dostupné z: [https://is.muni.cz/do/rect/el/estud/lf/js19/metodika\\_zp/web/docs/Metodika\\_pro\\_zpracovani\\_zaverecne\\_prace\\_skripta.pdf](https://is.muni.cz/do/rect/el/estud/lf/js19/metodika_zp/web/docs/Metodika_pro_zpracovani_zaverecne_prace_skripta.pdf). Scripta. Masarykova univerzita.
- [7] KNECHTOVÁ, Zdeňka, Andrea POKORNÁ, Edita PEŠÁKOVÁ a Dana DOLANOVÁ. Kvalitativní výzkum [online]. Brno, 2019 [cit. 20211109]. Dostupné z: <https://is.muni.cz/do/rect/el/estud/lf/>

- js19/metodika\_zp/web/pages/06-kvalitativni.html. Scripta. Masarykova univerzita.
- [8] HENDL, Jan. Kvalitativní výzkum: základní metody a aplikace [online]. Praha: Portál, 2005 [cit. 20211215]. ISBN 8073670402. Dostupné z: [https://dl1.cuni.cz/pluginfile.php/882806/mod\\_resource/content/1/06%20Hendl%2C%20J.%20-%20Kvalitativn%C3%AD%20v%C3%BDzkum%3B%2024-63.pdf](https://dl1.cuni.cz/pluginfile.php/882806/mod_resource/content/1/06%20Hendl%2C%20J.%20-%20Kvalitativn%C3%AD%20v%C3%BDzkum%3B%2024-63.pdf)
- [9] U.S.News: U.S. News Best Countries [online]. .S. News World Report L.P., 2021 [cit. 20220710]. Dostupné z: <https://www.usnews.com/news/best-countries/rankings>
- [10] Studijní a zkušební řád pro studenty ČESKÉHO VYSOKÉHO UČENÍ TECHNICKÉHO V PRAZE [online]. Praha: ČVUT, 2022, 8 9 [cit. 20211215]. Dostupné z: <https://fel.cvut.cz/cz/rozvoj/studijni-rad.pdf>
- [11] GHANBARI, Meysam. Business Goals vs. Objectives vs. Strategies vs. Tactics [online]. 16.06.2014 [cit. 20220413]. Dostupné z: <https://www.linkedin.com/pulse/20140616055721-142774715-business-goals-vs-objectives-vs-strategies-vs-tactics/>
- [12] The Guide to the Business Analysis Body of Knowledge™. International Institute of Business Analysis [online]. 17.06.2012 [cit. 20220418].
- [13] Systems and software engineering — Life cycle processes — Requirements engineering [online]. First edition. Switzerland: ISO/IEC/ IEEE, 2011 [cit. 20220418]. ISBN 9780738165912. Dostupné z: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6146379>
- [14] KYBKALO, Anatolij. Datové modelování [online]. Praha, 2013 [cit. 20220418]. Dostupné z: [https://web.archive.org/web/20141204102914/http://www.unicorncollege.cz/european-it-center/kybkalo-anatolij/attachments/PB\\_Kybkalo\\_Anatolij\\_1-4146-1.pdf](https://web.archive.org/web/20141204102914/http://www.unicorncollege.cz/european-it-center/kybkalo-anatolij/attachments/PB_Kybkalo_Anatolij_1-4146-1.pdf). BAKALÁŘSKÁ PRÁCE. UNICORN COLLEGE. Vedoucí práce Ing. Miroslav Žďárský.
- [15] Britannica, The Editors of Encyclopaedia. "clientserver architecture". Encyclopedia Britannica, [online] 13 May. 2021. [cit. 20220501]. Dostupné z: <https://www.britannica.com/technology/client-server-architecture>. Accessed 21 July 2022.
- [16] ThreeTier Architecture. IBM Cloud Education [online]. 28.10.2020 [cit. 20220501]. Dostupné z: <https://www.ibm.com/cloud/learn/three-tier-architecture>
- [17] PATTAKOS, Aris. Angular vs React vs Vue 2022 [online]. In: . 04.01.2022 [cit. 20220502]. Dostupné z: <https://athemes.com/guides/angular-vs-react-vs-vue/#:~:text=React%20is%20a%20UI%20library,them%20and%20understand%20their%20differences.>

- [18] Introduction to the Angular Docs: What is Angular? [online]. Google ©20102022, 2022 [cit. 20220507]. Dostupné z: <https://angular.io/guide/what-is-angular>
- [19] SULEIMANOV, Eugene, 2019, Creating Spring Security REST API using JWT token, YouTube video. [20220507]. Dostupné z [https://www.youtube.com/watch?v=yRnSUDx3Y8k&t=3418s&ab\\_channel=EugeneSuleimanov](https://www.youtube.com/watch?v=yRnSUDx3Y8k&t=3418s&ab_channel=EugeneSuleimanov)
- [20] React: Main Concepts [online]. Meta Platforms, 2022 [cit. 20220721]. Dostupné z: <https://reactjs.org/docs/hello-world.html>
- [21] Spring Boot: Overview [online]. VMware, 2022 [cit. 20220721]. Dostupné z: <https://spring.io/projects/spring-boot>
- [22] Apigee X documentation: CORS policy [online]. Google Cloud newsletter, 20220720 [cit. 20220724]. Dostupné z: <https://cloud.google.com/apigee/docs/api-platform/reference/policies/cors-policy>
- [23] Avgeriou, Paris Zdun, Uwe. (2005). [online] Architectural Patterns RevisitedA Pattern. [cit. 20220507]. Dostupné z: [https://www.researchgate.net/publication/215835792\\_Architectural\\_Patterns\\_Revisited--A\\_Pattern](https://www.researchgate.net/publication/215835792_Architectural_Patterns_Revisited--A_Pattern)





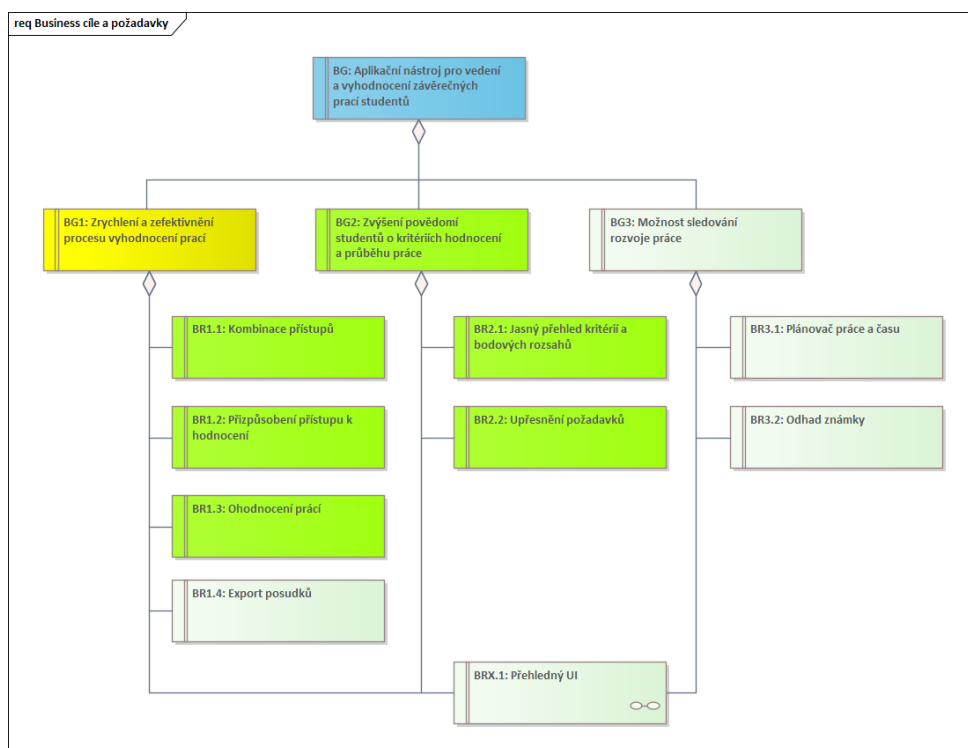
## Přílohy



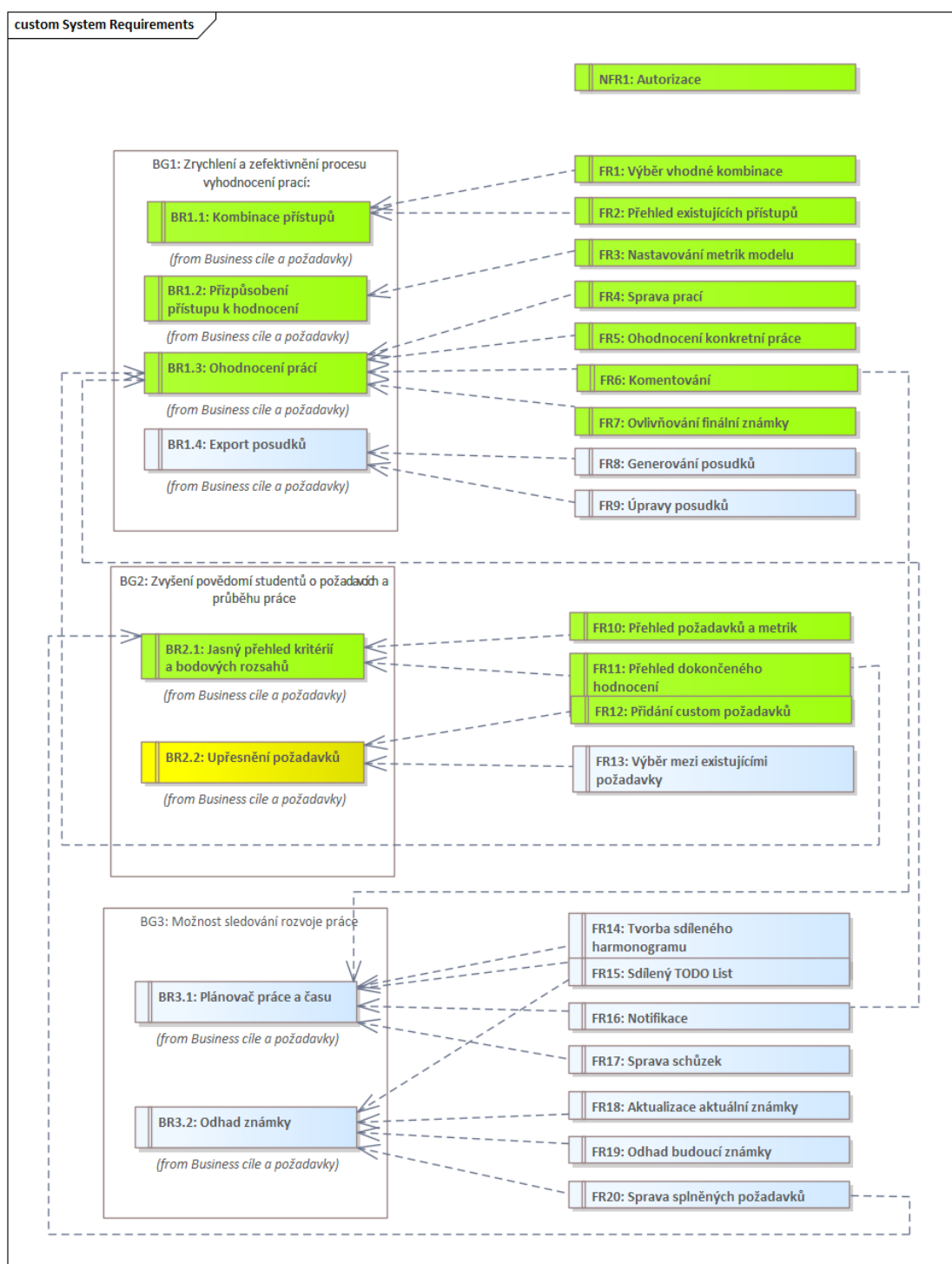


# Příloha A

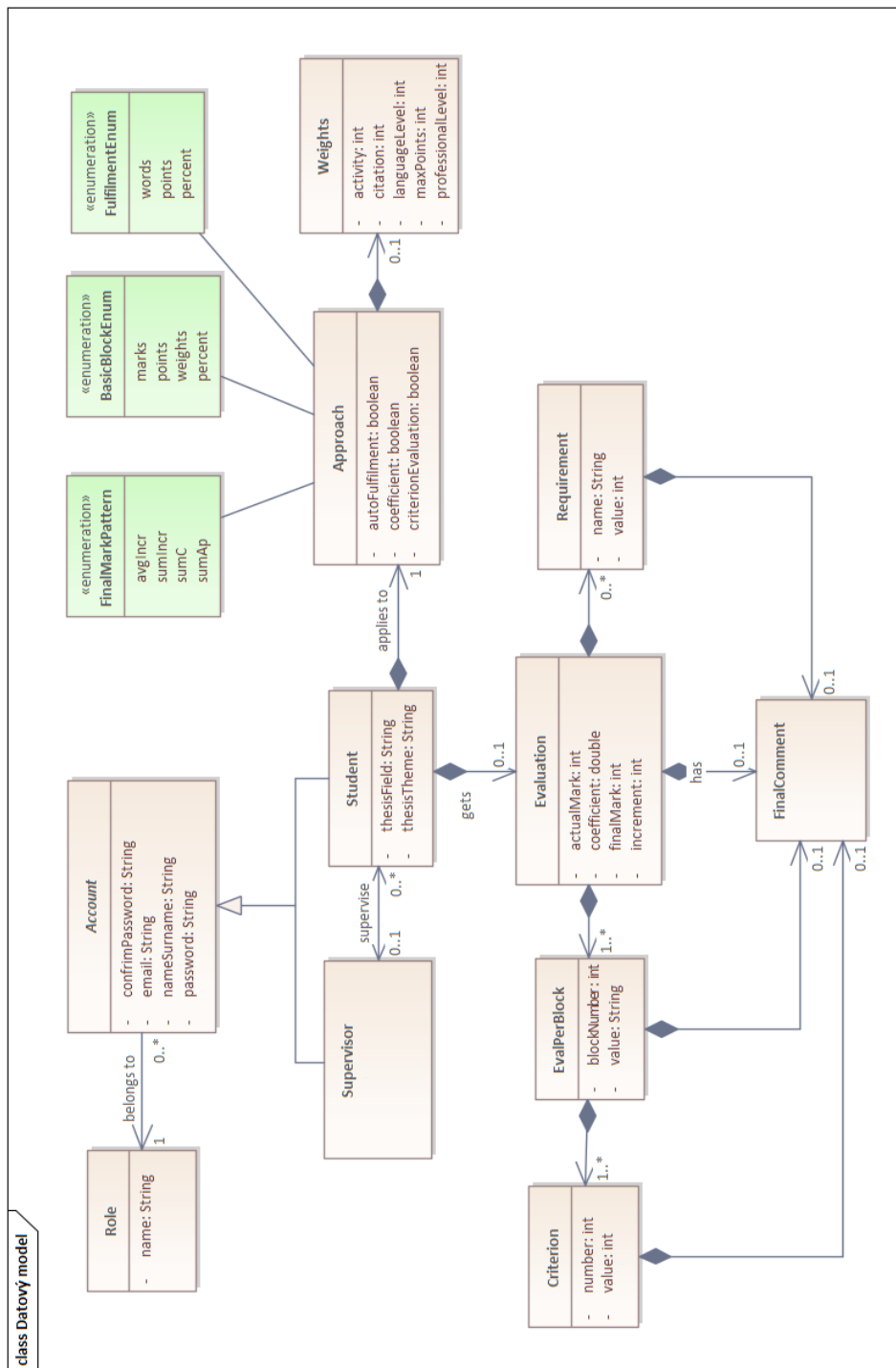
## Diagramy z návrhové části



Obrázek A.1: Business cíle a požadavky



Obrázek A.2: Systémové požadavky



Obrázek A.3: Datový model



## Příloha B

### Elektronické přílohy práce

Práce obsahuje dva soubory s elektronickými přílohami.

```
priloha1
├── latex.zip ..... zdrojová práce ve formátu LATEX
├── analyticke_vystupy
│   ├── prototypes_v1.csv ..... první návrhy prototypů modelu
│   ├── approach_combinations.csv ..... možností kombinace přístupů
│   │   └── hodnocení
│   ├── business_cíle_a_pozadavky.png
│   ├── dotaznik_hodnotitele.docx
│   └── dotaznik_studenti.docx
├── vytvorene_obrazovky_aplikace
│   ├── hodnoceni.png
│   ├── konfigurator.png
│   ├── login.png
│   ├── model.png
│   ├── prehled_dokonceneho_hodnoceni.png
│   ├── prehled_zvoleneho_pristupu.png
│   ├── pristupy.png
│   ├── registrace.png
│   └── studenti.png
├── zkoumane_modely
│   ├── BP_technicke_obory_CR
│   │   ├── oponent_FEL.docx
│   │   ├── vedouci_FEL.docx
│   │   ├── oponent_FIT.pdf
│   │   ├── vedouci_FIT.pdf
│   │   ├── oponent_VUT.pdf
│   │   └── vedouci_VUT.pdf
│   ├── BP_technicke_obory_mimo_CR
│   │   ├── norska_asociace_vsokych_skol.pdf
│   │   └── nemecka_univerzita_v_kahire.pdf
│   └── BP_jinych_zamereni_mimo_CR
│       ├── univerzita_v_mikkeli_model.pdf
│       └── univerzita_v_mikkeli_formular.xls
```

B. Elektronické přílohy práce

- └─ DP\_mimo\_CR
  - └─ ku\_leuven\_fakulta\_strojni.pdf
  - └─ technicka\_univerzita\_v\_delftu.xlsm
  - └─ fakulta\_eemcs\_delft.pdf
- priloha2 ..... zdrojový kód aplikace
  - └─ backend.zip
  - └─ frontend.zip
  - └─ instalacni\_prirucka.txt