

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra počítačů

## Softwarová podpora pro vzdálenou výuku

Daniel Koten

Vedoucí: Ing. Petr Aubrecht, Ph.D.  
Obor: Softwarové inženýrství a technologie  
Květen 2021



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Koten** Jméno: **Daniel** Osobní číslo: **483828**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra počítačů**  
Studijní program: **Softwarové inženýrství a technologie**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Softwarová podpora pro vzdálenou výuku**

Název bakalářské práce anglicky:

**Software Support for Remote Education**

Pokyny pro vypracování:

Během lockdownu v roce 2020 trpí učitelé nedostatkem softwarové podpory pro vzdálenou výuku, speciálně na základních školách. Zvlášť problematické je odevzdávání a kontrola domácích úkolů.

Úkolem je navrhnout systém, který by modelově řešil danou oblast. Není smyslem řešit problematiku komplexně (např. videohovory), ale soustředit se speciálně na domácí úkoly, které jsou ve stávajících systémech nejčastěji kritizovány uživateli ve všech rolích.

1. Zanalyzujte a popište aktuální situaci a problematické úkony ze strany učitele. Stávající systémy by měly zahrnovat Google Classroom, MS Teams, Škola v pyžamu.
2. Navrhněte řešení, které by učitelům usnadnilo práci, zaměřte se na domácí úkoly, snadné zadávání (v první fázi stačí plain text), vypracování (opět v první fázi plaintext), kontrolu (manuální učitelem, hodnocení, komentář, příp. vrácení).
3. Obzvlášť dbejte na přehlednost, aby každá strana věděla, co se po ní požaduje a v jakém stavu plnění se nachází. Seznam úkolů, jestli jsou hotové, termíny odevzdání, jejich překročení. Každá ze stran (učitel, žák) má jiné potřeby na uživatelské rozhraní.
4. Řešení implementujte v technologii JakartaEE.

Seznam doporučené literatury:

- [1] The Jakarta EE 8 Tutorial: <https://eclipse-ee4j.github.io/jakartaee-tutorial/toc.html>
- [2] Jakarta EE Cookbook - Second Edition, <https://www.packtpub.com/programming/jakarta-ee-cookbook-second-edition>
- [3] Patterns of Enterprise Application Architecture — Martin Fowler

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Petr Aubrecht, Ph.D., katedra počítačů FEL**

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

Datum zadání bakalářské práce: **12.02.2021**

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

Platnost zadání bakalářské práce: **30.09.2022**

Ing. Petr Aubrecht, Ph.D.  
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

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

## III. PŘEVZETÍ ZADÁNÍ

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

\_\_\_\_\_  
Datum převzetí zadání

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



## Poděkování

Tímto bych rád poděkoval Ing. Petru Aubrechtovi, Ph.D. za vedení práce, jeho vstřícnost a cenné rady. Stejně tak chci poděkovat všem svým blízkým za podporu při studiu a za pomoc při testování aplikace.

## Prohlášení

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

V Praze, 21. května 2021

## Abstrakt

V současné době roste potřeba po kvalitních nástrojích pro podporu online výuky, zejména v oblasti zadávání, odevzdávání a kontroly úkolů.

Nejhorší situace se zdá být na prvním stupni základních škol, kde je důležité dbát hlavně na jednoduchost používání, což stávající systémy často nesplňují.

Cílem této práce je tedy na základě analýzy existujících řešení a jejich nedostatků navrhnout a implementovat webovou aplikaci umožňující co nejsnazší práci s úkoly.

**Klíčová slova:** Online výuka, Jakarta EE, JSF, PostgreSQL, Web

**Vedoucí:** Ing. Petr Aubrecht, Ph.D.

## Abstract

Currently, there is a growing need for high-quality tools to support online learning, especially in the field of assignment, submission and control of tasks.

The worst situation seems to be at the first stage of primary schools, where it is important to pay particular attention to the ease of use, which the current systems often do not meet.

Thus, this work aims to analyze existing solutions and their shortcomings and based on the results to design and implement a web application that allows the easiest possible work with tasks.

**Keywords:** Online teaching, Jakarta EE, JSF, PostgreSQL, Web

**Title translation:** Software Support for Remote Education

## Obsah

<b>1 Úvod</b>	<b>1</b>
---------------	----------

### Část I Analýza

<b>2 Požadavky na systém</b>	<b>5</b>
------------------------------	----------

2.1 Funkční požadavky .....	5
-----------------------------	---

2.2 Nefunkční požadavky .....	8
-------------------------------	---

<b>3 Případy užití</b>	<b>9</b>
------------------------	----------

<b>4 Existující řešení</b>	<b>13</b>
----------------------------	-----------

4.1 Microsoft Teams .....	13
---------------------------	----

4.2 Google Classroom .....	15
----------------------------	----

4.3 Bakaláři .....	17
--------------------	----

4.4 Škola OnLine .....	19
------------------------	----

4.5 Škola v pyžamu .....	20
--------------------------	----

4.6 Moodle .....	21
------------------	----

4.7 Online cvičení .....	23
--------------------------	----

4.8 Umíme to .....	24
--------------------	----

4.9 Shrnutí .....	25
-------------------	----

### Část II

## Implementace

<b>5 Architektura</b>	<b>29</b>
-----------------------	-----------

5.1 Použité technologie .....	29
-------------------------------	----

5.1.1 Jakarta EE .....	29
------------------------	----

5.1.2 Primefaces .....	34
------------------------	----

5.1.3 PostgreSQL .....	34
------------------------	----

5.1.4 Payara Server .....	34
---------------------------	----

5.1.5 Docker .....	35
--------------------	----

5.1.6 NGINX Reverse Proxy Server	35
----------------------------------	----

5.2 Datový model .....	36
------------------------	----

5.3 Stavový diagram úkolu .....	36
---------------------------------	----

<b>6 Gamifikace</b>	<b>39</b>
---------------------	-----------

6.1 Body .....	39
----------------	----

6.2 Žebříček .....	40
--------------------	----

6.3 Avatar .....	40
------------------	----

6.4 Třídní chat .....	40
-----------------------	----

<b>7 Uživatelské rozhraní</b>	<b>43</b>
-------------------------------	-----------

7.1 Přihlašovací obrazovka .....	43
----------------------------------	----

7.2 Rozhraní pro učitele .....	44
--------------------------------	----

7.3 Rozhraní pro žáky .....	47
-----------------------------	----

7.4 Rozhraní pro administrátory . . .	50
<b>8 Průchody aplikací</b>	<b>51</b>
8.1 Zadání nového úkolu . . . . .	51
8.2 Odevzdání úkolu . . . . .	52
8.3 Schválení nebo vrácení pokusu . .	52
8.4 Opravné odevzdání úkolu . . . . .	53
8.5 Shrnutí . . . . .	54
<b>9 Testování</b>	<b>55</b>
<b>10 Závěr</b>	<b>57</b>
<b>Přílohy</b>	
<b>A Literatura</b>	<b>61</b>
<b>B Instalace</b>	<b>63</b>
<b>C Seznam zkratk</b>	<b>65</b>



## Obrázky

3.1 Diagram případů užití – žák a rodič . . . . .	10	5.4 Datový model . . . . .	37
3.2 Diagram případů užití – učitel a administrátor . . . . .	11	5.5 Stavový diagram úkolu . . . . .	38
4.1 Ukázka uživatelského rozhraní Microsoft Teams, vlastní screenshot	14	6.1 Ukázky avatarů, staženo pro nekomerční použití z [9] . . . . .	41
4.2 Ukázka uživatelského rozhraní Google Classroom, převzato z [1] .	16	7.1 Přihlašovací obrazovka . . . . .	43
4.3 Ukázka uživatelského rozhraní Bakaláři, převzato z [2] . . . . .	18	7.2 Učitel – seznam úkolů . . . . .	44
4.4 Ukázka uživatelského rozhraní Škola OnLine, převzato z [3] . . . . .	19	7.3 Učitel – detail úkolu . . . . .	45
4.5 Ukázka uživatelského rozhraní Škola v pyžamu, převzato z [4] . . .	21	7.4 Učitel – detail pokusu . . . . .	45
4.6 Ukázka uživatelského rozhraní Moodle, vlastní screenshot . . . . .	22	7.5 Učitel – chat . . . . .	46
4.7 Ukázka uživatelského rozhraní Online cvičení [5], vlastní screenshot	23	7.6 Učitel – profil . . . . .	46
4.8 Ukázka uživatelského rozhraní Umíme to [6], vlastní screenshot . .	24	7.7 Student – seznam úkolů . . . . .	47
5.1 Jakarta EE 8 – Diagram architektury, převzato z [7] . . . . .	30	7.8 Student – nový pokus . . . . .	47
5.2 Zjednodušená architektura technologie JSF, převzato z [8] . . .	33	7.9 Student – schválený úkol . . . . .	48
5.3 Životní cyklus JSF, převzato z [8]	34	7.10 Student – avataři . . . . .	49
		7.11 Student – žebříček . . . . .	49
		7.12 Student – profil . . . . .	50
		7.13 Administrátor – seznam žáků . .	50
		9.1 Ukázka testů v Selenium IDE . .	56

## Tabulky

8.1 Zadání nového úkolu . . . . .	52
8.2 Odevzdání úkolu . . . . .	52
8.3 Schválení nebo vrácení pokusu . .	53
8.4 Opravné odevzdání úkolu . . . . .	53



# Kapitola 1

## Úvod

Cílem této práce je navrhnout a implementovat platformu pro snadné zadávání a kontrolu online úkolů primárně pro základní školy. Právě na oblast domácích úkolů jsem se zaměřil z důvodu, že v existujících systémech není vždy řešena příliš šťastně – například není často na první pohled jasné, v jakém stavu úlohy jsou. Rád bych, aby pro každého uživatele bylo naprosto zřejmé, co se od něj očekává – žáci vidí, které úkoly musí ještě vypracovat nebo které byly vráceny k přepracování. Učitelé zase vidí, které úlohy jsou odevzdané a je tedy potřeba je zkontrolovat. Hlavním cílem je, aby byl systém maximálně přehledný a jednoduchý na použití. Každá operace má být tedy zvládnutelná pouze na několik kliknutí a uživatelské rozhraní by nemělo být přehlcené rozptýlujícími prvky.

V první části práce jsou nejprve vymezeny požadavky, které by měl systém splňovat. Na základě těchto požadavků jsem posléze provedl analýzu existujících řešení, kde každé z nich obsahuje podrobnější rozbor včetně silných a zejména slabých stránek. Z výsledku této analýzy nakonec vznikne souhrn, který poslouží jako podklad pro návrh a implementaci nové platformy.

Návrhu a implementaci se pak věnuje druhá část textu. Je zde ukázána například architektura systému, datový model, použité technologie včetně odůvodnění jejich použití, ukázky uživatelského rozhraní a další. Důraz je kladen zejména na pokrytí slabých stránek existujících řešení. Nakonec jsem se nad rámec zadání rozhodl implementovat i různé herní prvky, jak je podrobněji rozebráno v kapitole 6.





**Část I**

**Analýza**



## Kapitola 2

### Požadavky na systém

Pro začátek jsem se rozhodl stanovit požadavky, které by měl systém splňovat. V první řadě se jedná o funkční požadavky, ale v druhé polovině jsem směřoval pozornost i na nefunkční požadavky. Většina bodů vzešla z rozhovorů s učiteli, s vedoucím práce a nebo z mé vlastní zkušenosti a potřeby. Několik jich ale zpětně vyplynulo i z analýzy existujících řešení v kapitole 4, Existující řešení.

#### 2.1 Funkční požadavky

Tato sekce obsahuje číslovaný seznam všech funkčních požadavků na aplikaci. Jedná se o jeden z hlavních podkladů pro implementaci a dále jej rozvíjí kapitola 3, Případy užití, kde je vše rozebráno z více uživatelského úhlu pohledu.

##### **FP1 – Přihlášení**

Systém umožní uživateli se přihlásit, rozpozná jakou má roli a podle toho mu umožní přístup na jemu dostupnou část webu.

##### **FP2 – CRUD operace s úkolem**

Systém umožní učitelům provádět CRUD operace s úkolem u konkrétního předmětu v konkrétní třídě.

### **FP3 – Zobrazení seznamu úkolů**

Systém umožní zobrazit seznam úkolů u konkrétní výuky s možností řazení podle jednotlivých sloupců.

### **FP4 – Odevzdání úkolu**

Systém umožní žákovi opakovaně odevzdávat řešení (pokus) úkolu, pokud je mu vždy vrácen k přepracování.

### **FP5 – Vrácení úkolu**

Systém umožní učiteli pokus úkolu vrátit k přepracování s nepovinnou poznámkou.

### **FP6 – Schválení úkolu**

Systém umožní učiteli schválit pokus úkolu s nepovinným komentářem, čímž je celý úkol u studenta považován za splněný.

### **FP7 – Zobrazení statistik úkolů**

Systém umožní učiteli u každého úkolu zobrazit počty studentů v jednotlivých stavech (nový, odevzdáno, vráceno, znovu odevzdáno, schváleno, nesplněno, omluveno).

### **FP8 – Zobrazení statistik žáků**

Systém umožní učiteli u každého žáka zobrazit počty úkolů v jednotlivých stavech (stejně jako ve FP7).

### **FP9 – Počty záležitostí k vyřešení**

Systém ukáže učiteli u každé třídy a každého předmětu počty záležitostí k vyřízení (počty studentů, kteří odevzdali řešení nějakého úkolu a je tedy potřeba provést kontrolu řešení). Žákovi systém naopak ukáže u každého předmětu počty nových nebo vrácených úkolů k přepracování.

### **FP10 – Seznam pokusů (žák)**

Systém umožní žákovi zobrazit seznam jeho pokusů u konkrétního úkolu.

### **FP11 – Detail pokusu**

Systém umožní uživateli zobrazit detail řešení (pokusu) úkolu.



### **FP12 – Seznam žáků s pokusy**

System umožní učiteli u každého úkolu zobrazit seznam žáků s jejich řešeními (pokusy).

### **FP13 – Zobrazení spolužáků**

System umožní uživateli zobrazit seznam studentů v aktuální třídě.

### **FP14 – CRUD operace s žákem**

System umožní učiteli provádět CRUD operace s žákem ve třídě, kde učí.

### **FP15 – Poslední pokusy**

System ukáže učiteli posledních 10 odevzdaných pokusů v daném předmětu.

### **FP16 – CRUD operace se třídou**

System umožní administrátorovi provádět CRUD operace se třídou.

### **FP17 – CRUD operace s učiteli**

System umožní administrátorovi provádět CRUD operace s učitelem.

### **FP18 – CRUD operace s žáky**

System umožní administrátorovi provádět CRUD operace s žákem.

### **FP19 – CRUD operace s avatary**

System umožní administrátorovi provádět CRUD operace s avatary.

### **FP20 – Třídní chat**

System umožní učitelům a žákům zobrazit třídní chat pro každý předmět a psát do něj nové zprávy.

### **FP21 – Nákup avatara**

System umožní žákovi koupit si na určitou dobu za své body avatara z nabídky.

### **FP22 – Volba avatara**

System umožní žákovi přepínat aktivního avatara mezi těmi, které aktuálně vlastní. Případně si může nastavit základního, který je zdarma.

## 2.2 Nefunkční požadavky

Oproti předchozí sekci se zde zaměřím na požadavky, které nedefinují přímo funkčnost aplikace, ale spíše její kvalitu. Mohou také klást určitá omezení.

### NP1 – Přehlednost

Nejdůležitějším bodem je co nejvyšší přehlednost a jednoduchost celé aplikace. Uživatel musí stále vědět, kde se zrovna nachází a co se po něm na dané stránce požaduje.

### NP2 – Konzistence

Neméně důležitá je konzistence dat. Když uživatel vloží nějaká data, musí je opět najít tam, kde je očekává v očekávané podobě.

### NP3 – Odolnost proti uživateli

System by měl chránit uživatele před nechtěnými akcemi. Například mazání dat by mělo být umožněno pouze tam, kde je to opravdu potřeba a mělo by být vyžadováno potvrzení.

### NP4 – Bezpečnost

System by měl být odolný proti různým útokům jako například SQL injection. Hesla by měla být ukládána v bezpečné podobě. A uživatel nesmí mít možnost dostat se do části webu, kam nemá přístup.

### NP5 – Výkon

Je potřeba také počítat s tím, že základní školy mají stovky žáků a není přípustné, aby pod tímto větším náparem přestával systém fungovat tak, jak má.



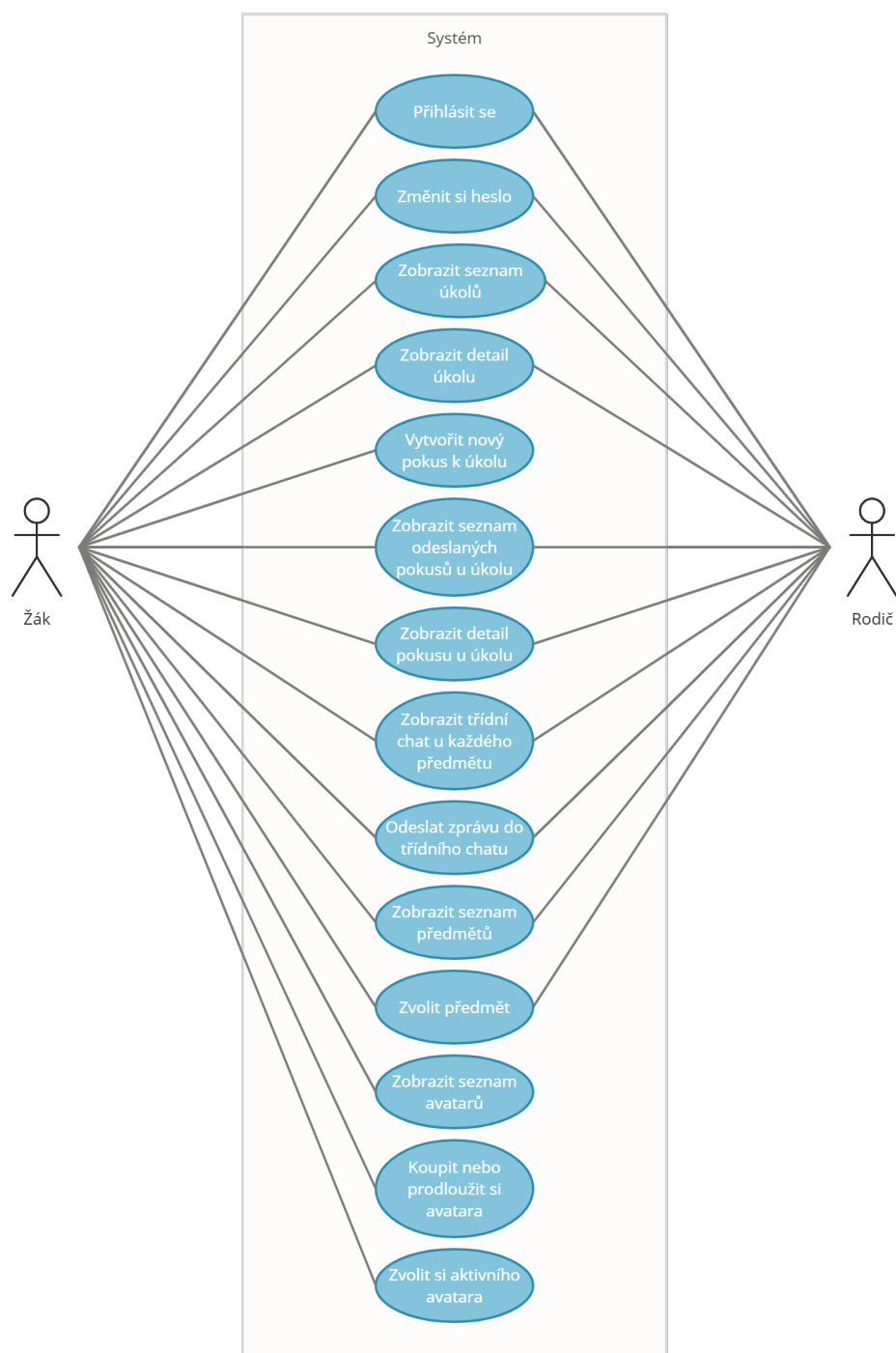
## Kapitola 3

### Případy užití

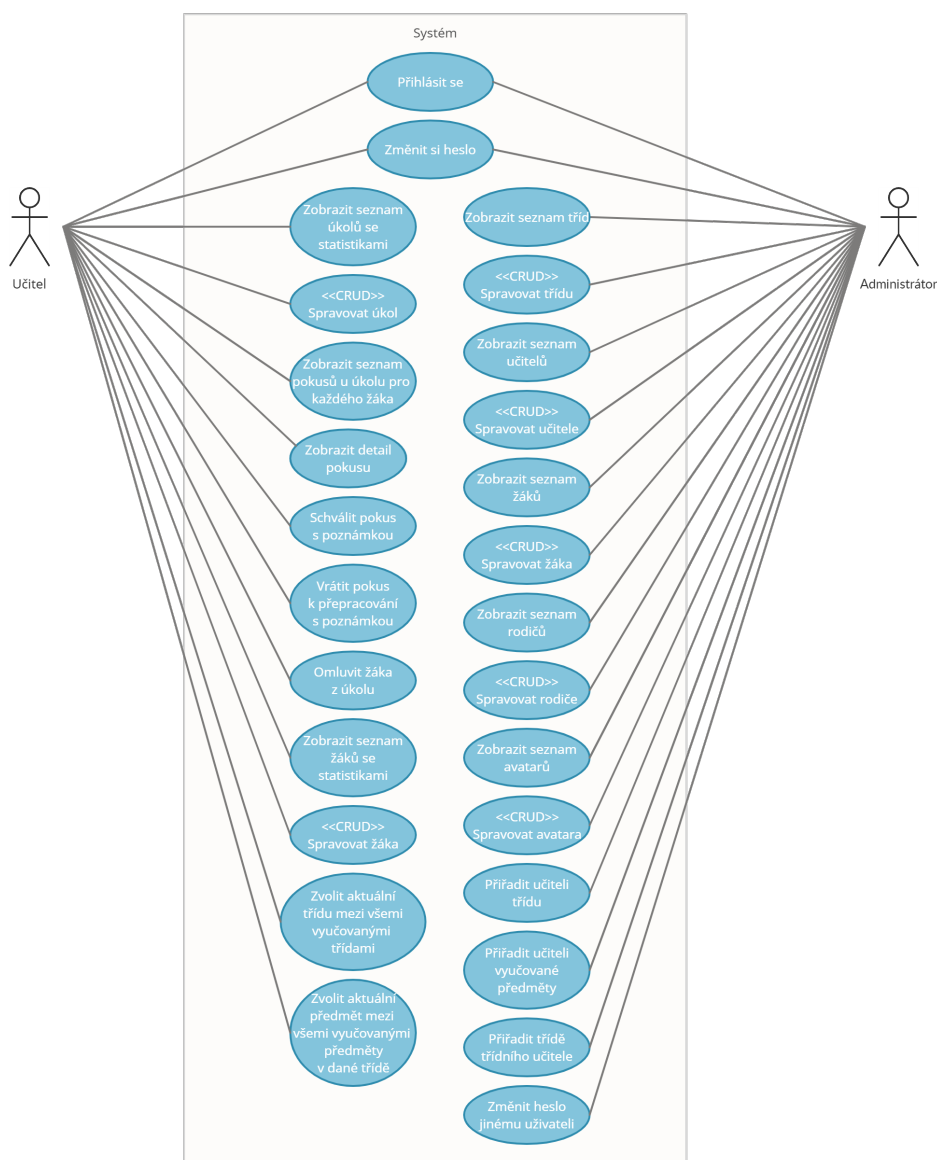
V předchozí kapitole jsem zadefinoval funkční a nefunkční požadavky. Nyní z nich odvodím jednotlivé případy užití, které poskytnou ucelenější pohled a dobře pak poslouží jako podklad pro implementaci.

Zároveň jsem podle požadavků definoval 4 role uživatelů, ke kterým jsem jednotlivé případy užití rozdělil.

V neposlední řadě, pro některé nejdůležitější případy užití jsem rozepsal detailní scénáře v kapitole 8, Průchody aplikací.



**Obrázek 3.1:** Diagram případů užití – žák a rodič



**Obrázek 3.2:** Diagram případů užití – učitel a administrátor



## Kapitola 4

### Existující řešení

V následující kapitole rozeberu konkurenční systémy, které se již pokouší požadovanou funkcionalitu implementovat. U každého z nich zmíním nejprve obecné informace k čemu slouží a následně se zaměřím na jejich klady a zápory. Zároveň bych rád zmínil, že u některých systémů nebylo zrovna snadné si je vyzkoušet a provést nějakou analýzu. Důvodem je to, že ne všechny zmiňované systémy jsou zdarma a přístupné široké veřejnosti. Abych uvedl příklad, tak Google Classroom je možné využívat pouze skrz školní účet registrovaný u některé školy, která si tento systém platí. Podobný problém je i pro většinu dalších systémů. Tuto nepříjemnost jsem řešil většinou tak, že jsem vycházel například z různých návodů pro užívání těchto systémů dostupných na internetu.

#### 4.1 Microsoft Teams

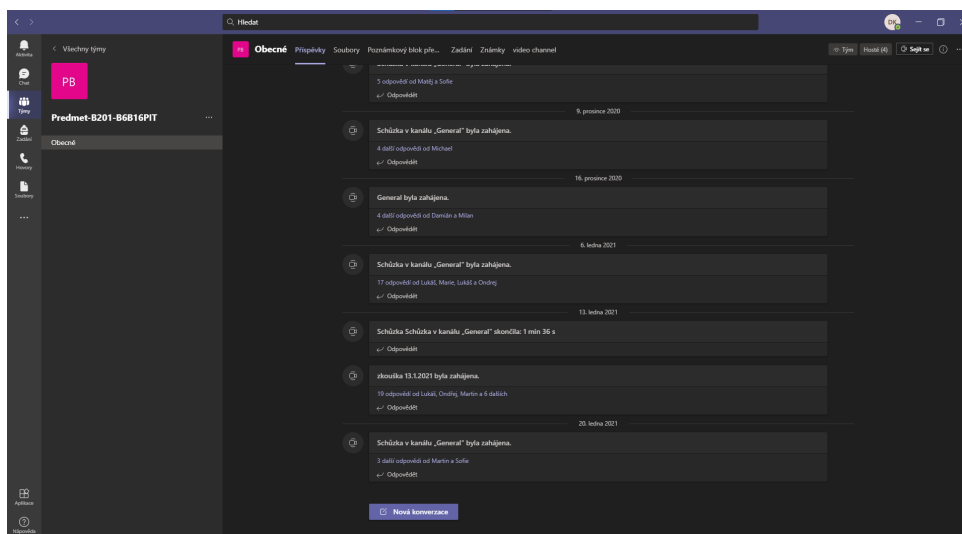
V současné době pro nás, jakožto studenty ČVUT, velmi známý a hojně využívaný nástroj. Ukázkou uživatelského rozhraní lze vidět na obrázku 4.1.

Rozhodně tedy patří k těm populárnějším, a to nejen ve školách, ale zejména i ve firemní sféře. Jedná se totiž o produkt z dílny Microsoftu, díky čemuž nabízí nadstandardní možnosti integrace s dalšími nástroji jako například cloudové úložiště, poznámkové bloky a hlavně umožňuje napřímo editovat dokumenty ve formátu .docx.

Samozřejmě jej lze doplnit i o další pluginy jako například Microsoft Planner pro plánování nejrůznějších úkolů a podobně.

Všechno toto lze samozřejmě považovat za obrovskou výhodu, a to převážně v již zmiňovaných firmách. Nebo obecně tam, kde si lidé s takto komplexním nástrojem poradí. Například tedy i vysoké či některé střední školy.

Tak velké množství funkcí se ovšem může stát nevýhodou v těch oblastech, kde je vyžadována spíše jednoduchost a co nejvyšší uživatelská přívětivost, a to rozhodně základní školy jsou. Takoví uživatelé ocení spíše systém, přesně šitý na míru jejich potřebám. Budou rádi, když nebudou muset nikde nastavovat žádné oprávnění a podobné věci. Nakonec by také mohli mít problém s vícero Microsoft účty, mezi kterými by potřebovali přepínat. Například pracovní a osobní.



**Obrázek 4.1:** Ukázka uživatelského rozhraní Microsoft Teams, vlastní screenshot

Za výhody lze tedy považovat:

- Vysoká možnost integrace s dalšími nástroji
  - MS Office
  - OneNote
  - OneDrive
  - a další...
- Podpora velké firmy a kvalitní zázemí
- Poměrně rozšířený a prozkoušený systém.
- Dostupná mobilní aplikace



Na druhou stranu, nedostatky jsou například:

- Přílišná komplexnost vzhledem k základním školám (nabízí příliš mnoho nepotřebných možností a funkcí)
- Naopak postrádá například propracovanější systém pro odevzdávání a kontrolu úkolů
- Nedostatečné možnosti nastavení práv
  - Skrýt některé kanály v týmu pro určité uživatele
  - Vymezit pro hosta, co přesně může a nemůže vidět
- Nutnost vytvářet si sám strukturu týmů a skupin. Toto často vede k nekonzistentním stavům, kde například každý předmět je trochu jinak uspořádaný a uživatel je poté velmi zmatený.
- Při posílání dokumentů je zde potřeba myslet na adresářovou strukturu, aby zůstal obsah přehledný. Toho nemusí být malé děti schopné a i pro dospělé to může být obtěžující.

## 4.2 Google Classroom

O něco více se potřebám škol snaží přiblížit platforma Google Classroom. Ukázkou jejího rozhraní lze vidět na obrázku 4.2. Nicméně i ta má několik nedostatků.

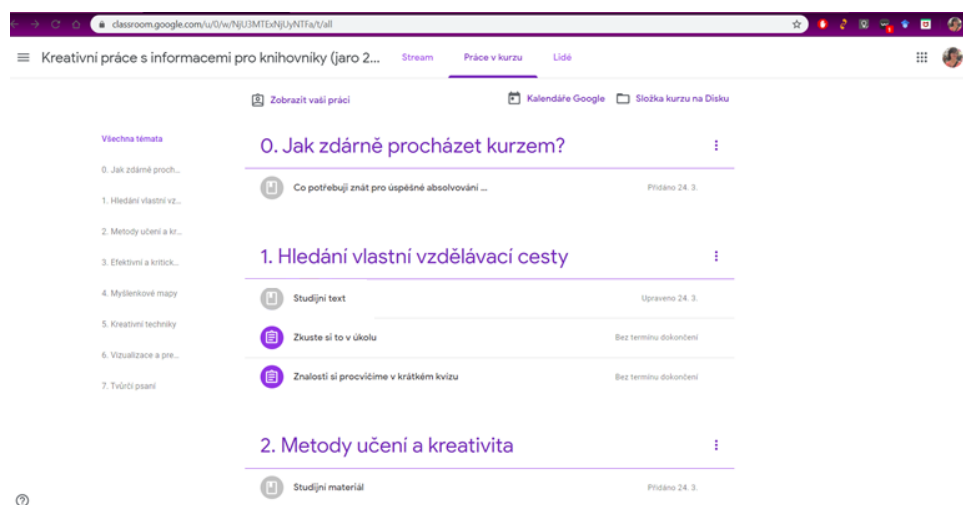
Abych začal výhodami, tak musím zmínit poměrně čisté a hezké uživatelské rozhraní, které i přes svou jednoduchost nabízí spoustu skvělých funkcí.

Učitel má možnost vytvářet nové kurzy, do kterých může přidávat témata s jednotlivými úkoly nebo může jen psát příspěvky.

Do těchto kurzů se musí žáci přihlašovat sami pomocí kódu. Už to by se dalo považovat za nevýhodu, protože sice se může zdát, že to ušetří práci učiteli, nicméně dovedu si představit, že s tím některý žák bude mít problém a ve finále učitel stejně žádný čas neušetří, protože bude muset žákovi pomoci se do kurzu dostat. V tomto ohledu se mi tedy jeví jako lepší varianta to, že žák dostane již vytvořený účet, který je připojen do všech předmětů ve třídě.

Výhodou může být integrace s dalšími nástroji od společnosti Google. Například Google dokumenty, tabulky, Google disk a podobně. Nicméně i to s sebou nese spoustu problémů, které popisují v dalším odstavci.

Mezi velké nevýhody patří to, že se uživatel musí přihlašovat svým školním Google účtem. Z toho plyne několik problémů. Jednak jsem se často setkal s uživateli, kteří ani nevědí, že nějaký Google účet mají. Ale mnohem horší je pak přepínání účtů. Téměř každý uživatel má, ať už nevědomky nebo vědomě, svůj osobní Google účet, ale najednou se musí přihlašovat pomocí jiného. Už to může být pro velkou část uživatelů dost zmatečné, a navíc se k tomu pak přidávají podobné problémy například při použití Google disku nebo Google dokumentů či fotek. Uživatel je například na svém mobilním telefonu přihlášen pod svým osobním účtem, něco vyfotí, myslí si, že to najde na počítači ve svých Google fotkách, ale pod jiným účtem tyto fotky nenajde.



**Obrázek 4.2:** Ukázka uživatelského rozhraní Google Classroom, převzato z [1]

Výhody jsou tedy podobné jako u MS Teams:

- Integrace s dalšími Google nástroji
  - Google fotky
  - Google dokumenty, tabulky
  - Google disk
  - a další...
- Podpora velké společnosti a kvalitní zázemí
- Poměrně rozšířený a zaběhlý systém
- Čisté a moderní uživatelské rozhraní
- Dostupná mobilní aplikace

Ovšem i zde existují nedostatky:

- Přepínání Google účtů (osobní, školní)
- Žáci se musí dostat do kurzů sami
- Odevzdávání řešení
  - Žák může buď psát přímo do zadání, pak si ho ale může, byť jen omylem, změnit.
  - Nebo může mít práva zadání pouze zobrazit. Pak je pro něj ale situace složitější v tom, že musí vytvořit nový dokument a poradit si s formou sám. Buď zadání zkopíruje a dopisuje do něj nebo se na otázky nějak odkazuje např. čísly.
- Nepropracované multiple choice úkoly v porovnání např. s Moodle (chybí kupříkladu poměrné bodové hodnocení za částečné řešení)

## 4.3 Bakaláři

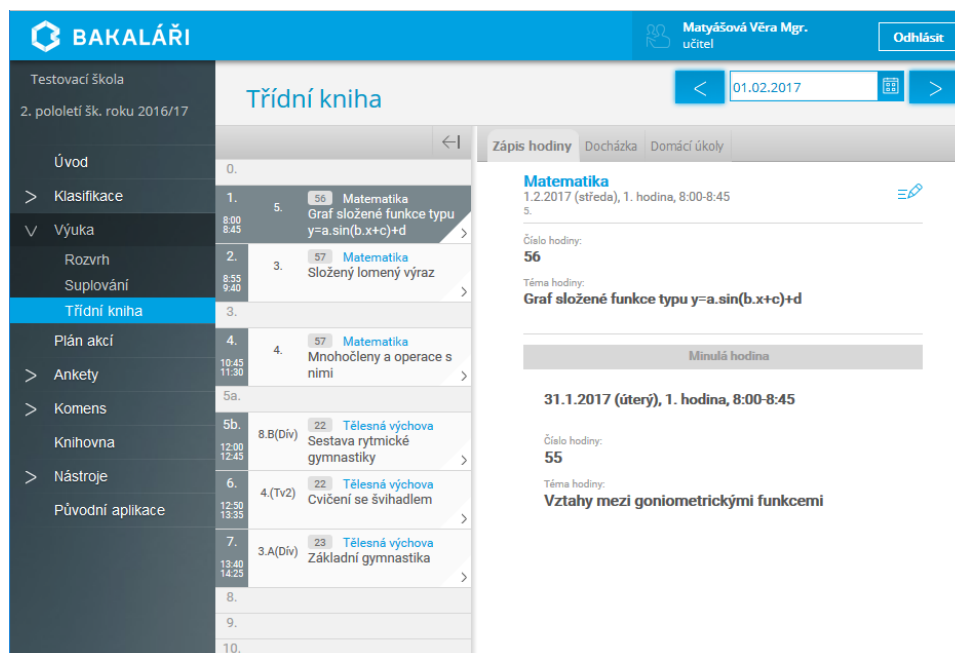
Nyní už se dostávám k nástroji, který se skutečně snaží pokrýt potřeby škol. Uživatelům poskytuje nepřeberné množství funkcí včetně například plánovaných akcí školy, knihovny, suplování a další. Právě tyto, z mého pohledu, trochu nadbytečné funkce činí ovšem tento systém hovorově řečeno "přeplácáný" jak lze vidět na ukázce 4.3.

Samozřejmě netvrdím, že je to úplně špatně. Tento nástroj je přeci jen stavěn spíše pro kompletní pokrytí všech požadavků škol, což splňuje dobře. Nicméně předmětem mé práce je přijít s řešením, které bude velmi jednoduché a intuitivní pro všechny druhy uživatelů a to jde trochu proti sobě s pokrytím všech myslitelných požadavků.

S předchozím bodem také souvisí historický vývoj této platformy, která byla původně koncipována jako desktopová aplikace. V současné době je ovšem zájem spíše o webové rozhraní, a tak vzniká nekonzistence mezi těmito dvěma prostředími. Jednak vypadají rozdílně, a navíc poskytují rozdílné funkcionality, což může být velice matoucí.

Také je nutno podotknout, že systém Bakaláři je v některých ohledech vcelku "nedotažený". Například bych rád zmínil proces zadávání a schvalování úkolů. Ze sesbíraných referencí vím, že učitelé často zapomenou označit úkol za dokončený, když ho odevzdají všichni studenti. Pak se stává to, že ho i nadále

všichni studenti vidí ve svém seznamu úkolů a časem se z tohoto seznamu stane nepořádek. Lepší by bylo, kdyby se žákovi úkol po schválení přestal zobrazovat v seznamu a případně se přesunul do nějakého méně nápadného seznamu odeslaných či hotových úkolů. Učitel by pak viděl úkol také jen do té doby než ho dořeší všichni studenti a následně by se automaticky označil jako dokončený a také by se přesunul do jiného seznamu.



**Obrázek 4.3:** Ukázka uživatelského rozhraní Bakaláři, převzato z [2]

Výhody:

- Snaha o kompletní pokrytí všech požadavků škol
- Dostupné výukové materiály pro práci se systémem
- Dostupná mobilní aplikace
- Vhodné spíše pro úplnou digitalizaci školy

Nedostatky:

- Náročnější k naučení se všech možných funkcí
- Příliš komplexní (vzhledem k požadavku na co největší jednoduchost nabízí tento systém příliš nepotřebných funkcí, například akce školy, knihovny, atd.)

- V některých částech "nedotažené"(např. splnění úkolu)
- Různý vzhled a funkce na různých platformách (zejména desktopová a webová aplikace)
- Často padá či nefunguje.
  - To může být způsobeno historicky použitím různých programovacích jazyků a technologií

## 4.4 Škola OnLine

Tato platforma je poměrně dost podobná zmiňovaným Bakalářům. Možná i to je důvod proč se tyto platformy v roce 2019 spojily. Také se snaží pokrýt většinu požadavků škol a i když není zatím tolik komplexní, také je poměrně těžkopádná. Ukázku uživatelského rozhraní lze vidět na obrázku 4.4

The screenshot displays the Škola OnLine web application interface. At the top, there is a navigation bar with the logo on the left, the user's name 'Adamkovičová Adéla Administrátor' on the right, and the school year '2017/2018'. Below the navigation bar is a horizontal menu with various options: 'Třídní kniha', 'Hodnocení', 'Výuka', 'Rozvrh a suplování', 'Evidence osob', 'Výstupy', 'Komunikace', 'Přijímací řízení', 'Ostatní moduly', and 'Správa aplikace'. The main content area is titled 'Kalendář' and shows a calendar for the month of September 2017. The calendar is a grid with days of the week (Po, Út, St, Čt, Pá) and dates (1-30). Below the calendar, there are checkboxes for 'Možnosti zobrazení': 'Zobrazit rozvrh', 'Zobrazit hodnocení', 'Zobrazit sobotu a neděli', and 'Zobrazit názvy skupin tříd'. The footer contains contact information for ŠKOLA ONLINE a.s., including the address 'Karlovo nám. 288/17, Praha 2', phone number '378 229 460', and email 'hotline@skolaonline.cz'.

**Obrázek 4.4:** Ukázka uživatelského rozhraní Škola OnLine, převzato z [3]

Výhody:

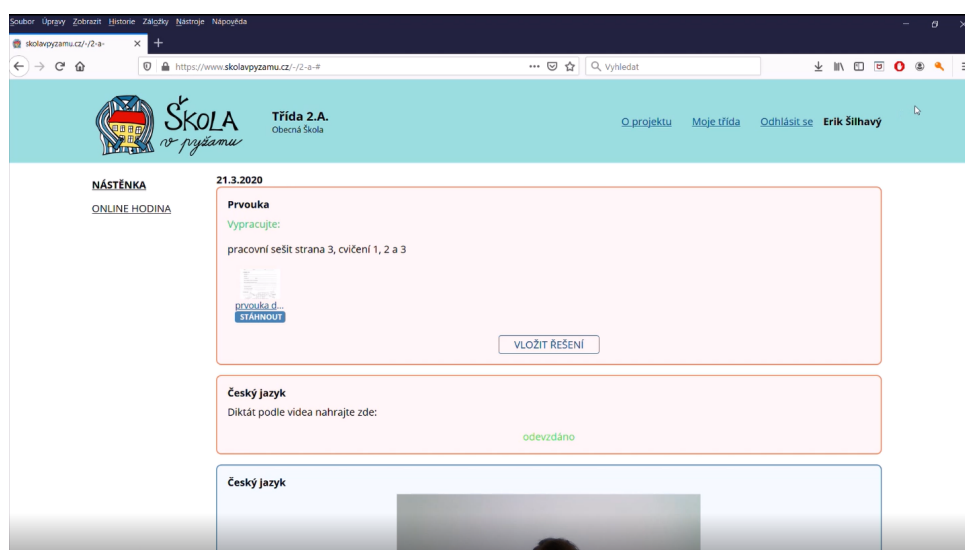
- Snaha o kompletní pokrytí všech požadavků škol
- Dostupné výukové materiály pro práci se systémem
- Dostupná mobilní aplikace
- Vhodné spíše pro úplnou digitalizaci školy

Nedostatky:

- Příliš komplexní (podobně jako Bakaláři, tak i Škola OnLine nabízí vzhledem k požadavku na co nejvyšší jednoduchost příliš nepotřebných funkcí, například knihovna, školní akce)
- Neintuitivní uživatelské rozhraní (Pomohly by například jen drobnosti jako, aby názvy úkolů a podobné nápisy fungovaly jako odkazy)

## 4.5 Škola v pyžamu

Poměrně hezkým projektem je Škola v pyžamu, jehož uživatelské rozhraní lze vidět na obrázku 4.5. Zde se autoři snaží právě o jednoduchost, kterou rodiče a malé děti ocení nejvíce. Nicméně zde na druhou stranu zase několik věcí chybí. Například učitel si nemůže psát k žákům různé poznámky či nevidí přehledné shrnutí všech úkolů. Vidí pouze úkoly pod sebou a u každého z nich seznam žáků se stavem vypracování. Také by bylo vhodnější rozdělení na jednotlivé předměty a snazší přepínání tříd pro učitele.



**Obrázek 4.5:** Ukázka uživatelského rozhraní Škola v pyžamu, převzato z [4]

Výhody:

- Velmi jednoduché prostředí (dáno ovšem absencí některých funkcí)
- Zdarma

Nedostatky:

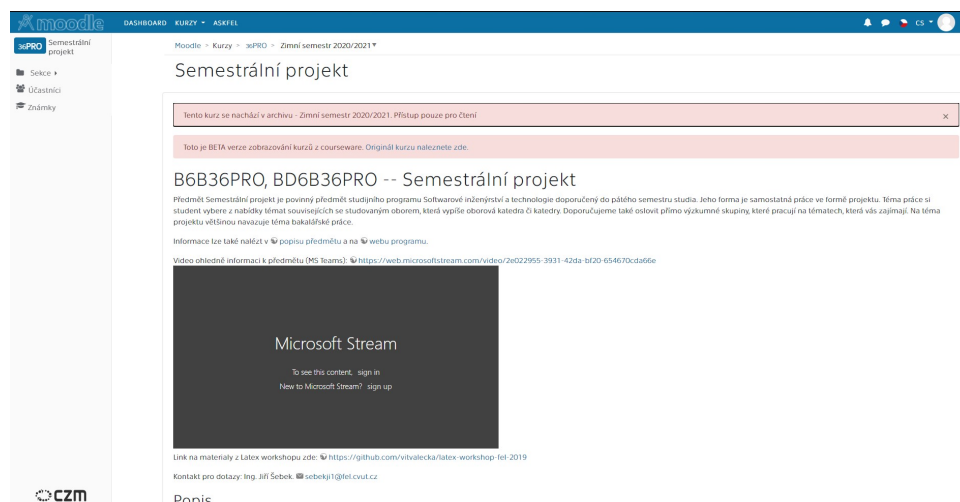
- Chybí mobilní aplikace
- Chybí lepší rozdělení jednotlivých předmětů
- Složitě přepínání mezi třídami

## 4.6 Moodle

Tento systém je znám spíše na vysokých školách. Jedná se o poměrně komplexní nástroj, který umožňuje i tvorbu testů různých druhů, systém známkování, multijazyčnost a další. Nicméně, jak již vyplývá z faktu, kde se využívá, není jeho využití pro základní školy příliš vhodné. Přeci jen není tolik intuitivní, jak by bylo potřeba. Navíc je vhodné pro jeho konfiguraci mít k dispozici

#### 4. Existující řešení

IT odborníka, kterých nebývá na základních školách dostatek. Ukázkou jeho uživatelského rozhraní lze vidět na obrázku 4.6.



**Obrázek 4.6:** Ukázka uživatelského rozhraní Moodle, vlastní screenshot

Výhody:

- Poměrně dobré možnosti tvorby online testů
  - Průběžné odesílání řešení pomocí AJAX
  - Možnost nastavit poměrné bodové hodnocení částečně správných řešení
- Různé role uživatelů

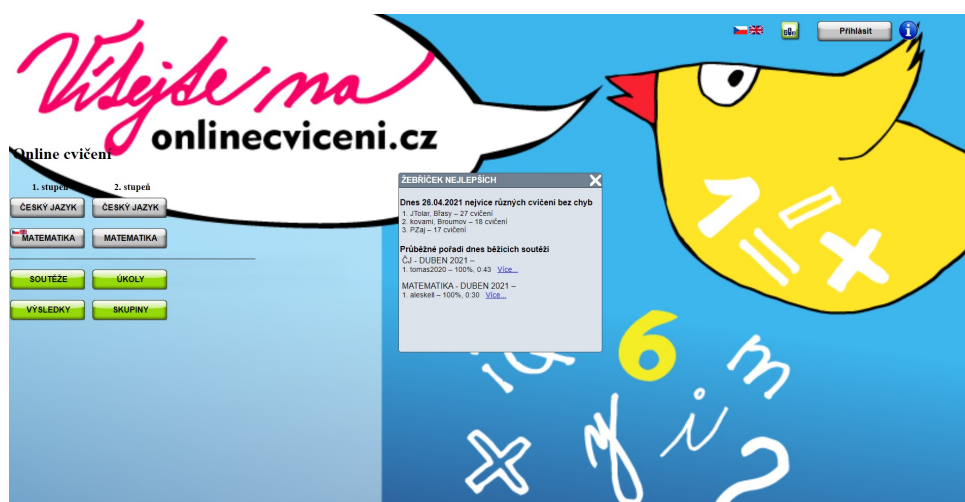
Nedostatky:

- Nepříliš intuitivní
  - Například na úvodní stránce by bylo dobré u každého předmětu vidět přehled toho, co je potřeba udělat.
- Má problémy na mobilních zařízeních, kde někdy dokonce není vidět část obsahu.
- Složitější konfigurace



## 4.7 Online cvičení

Tento web je zaměřen trochu jinak. Je vhodný spíše pro procvičování různých typů cvičení. Žák zde nalezne spoustu dostupných úkolů, ale tím výhody v podstatě končí. Celkově systém není příliš "dotažený". Například přidávání do skupin probíhá zbytečně složitým způsobem, kdy se žák musí sám registrovat, vyhledat podle kódu skupinu, žádat admina o přidání a pak doufat, že ho admin přidá. Mnohem jednodušší se mi jeví řešení, kdy toto vše vyřídí za žáky sám učitel. Ukázkou tohoto webu lze vidět na obrázku 4.7.



**Obrázek 4.7:** Ukázka uživatelského rozhraní Online cvičení [5], vlastní screenshot

Výhody:

- Velké množství obsahu (různá cvičení)

Nedostatky:

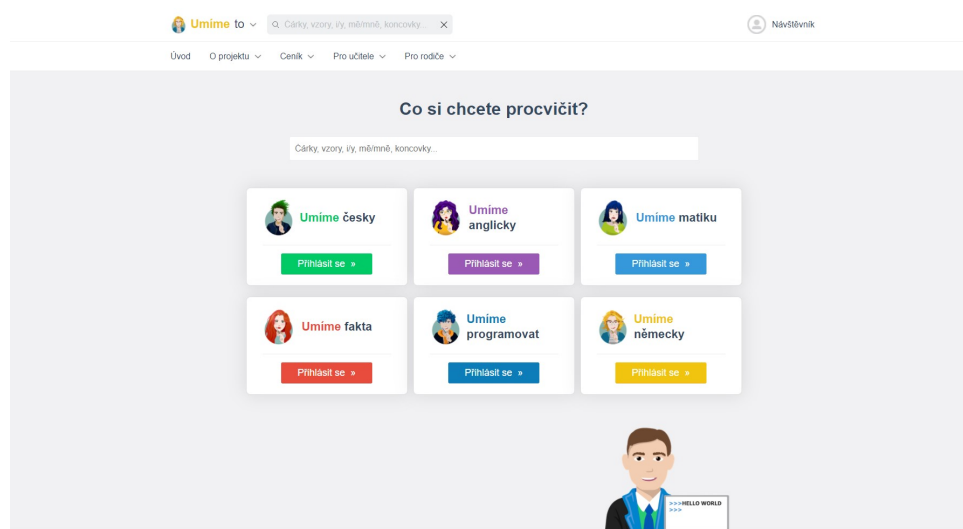
- Nedotažené (zbytečně složité a nepřehledné vzhledem k tak malému počtu funkcí)
- V dnešní době odrazující uživatelské rozhraní

## 4.8 Umíme to

Podobným směrem jako předchozí systém se vydává i Umíme to. Také je zde k dispozici ještě větší škála různých úkolů z různých předmětů. Tento nástroj je ovšem mnohem lépe propracovaný. Nabízí poměrně hezké a čisté uživatelské rozhraní (vizte obrázek 4.8) a působí uceleným dojmem.

Na druhou stranu i zde vidím několik nedostatků. Stejně jako u předchozího, je zde problém s registrací žáků. Ti se musí registrovat a připojit do třídy opět sami. Může to znít jako maličkost, ale umím si představit, že u dětí na prvním stupni základních škol to nemusí být pro všechny rodiče jednoduché a může to v nich hned od začátku vyvolat ze systému nepříjemný pocit. Pak ani nemusí mít snahu a chuť pochopit samotné funkce, které jsou jim po přihlášení k dispozici.

Také by bylo vhodné učitelům nabídnout více statistik a například psát si ke studentům různé poznámky. Vhod by přišel také třídní chat.



**Obrázek 4.8:** Ukázka uživatelského rozhraní Umíme to [6], vlastní screenshot

Výhody:

- Hezké a čisté uživatelské rozhraní.
- Velká spousta předpřipravených úkolů různých typů

Nedostatky:

- Omezení jen na předpřipravené úlohy
- Málo statistik
- Registraci a připojení ke třídě si musí řešit žáci sami
- Chybějící chat

## 4.9 Shrnutí

Z předchozích implementací vychází několik poznatků, které bych zde rád shrnul.

V první řadě většina z nich je příliš složitá pro žáky prvního stupně. Svou implementaci bych tedy měl směřovat hlavně k tomu, aby byla jednoduchá, intuitivní a přehledná.

S tím souvisí další častý problém s registracemi. Bude mnohem jednodušší, když si žáky přidá do třídy rovnou učitel a žáci se pak jen přihlásí uživatelským jménem, které jim je sděleno a zvolí si pouze své heslo.

To samé by mohlo platit i pro samotné učitele a třídy. Ty bude registrovat a vytvářet administrátor a rovnou je k sobě přiřadí. Učitel se tedy jen přihlásí a zvolí si pouze své heslo stejně jako žák. Tím nemusí nic dalšího řešit a může jít rovnou vytvořit účty žákům.

Dále bude potřeba propracovat zadávání a odevzdávání úkolů. Mělo by platit, že zadání může měnit pouze učitel a žák k němu posílá pouze své řešení. To může následně učitel vrátit k přepracování s komentářem, ovšem nesmí upravovat žákovo řešení, aby ho například omylem nesmazal.

Všichni uživatelé musí jasně vidět přehled toho, co musí ještě udělat. Pokud tedy například žák odešle řešení úkolu, tak by se měl tento úkol rovnou přesunout do méně důležité sekce, aby se nepletl s aktuálními povinnostmi.

Snadné by také mělo být přepínání mezi předměty (v případě učitele i mezi třídami).

Vhodnou funkcí je i třídní chat. Ideálně pro každý předmět zvlášť.





## Část II

### Implementace



# Kapitola 5

## Architektura

Nedílnou částí návrhu implementace je architektura samotného řešení, které se věnuje tato kapitola.

### 5.1 Použité technologie

V této sekci bych rád představil technologie, které jsem se rozhodl použít. U každé z nich uvedu stručný popis a důvody, proč má v tomto projektu své místo.

#### 5.1.1 Jakarta EE

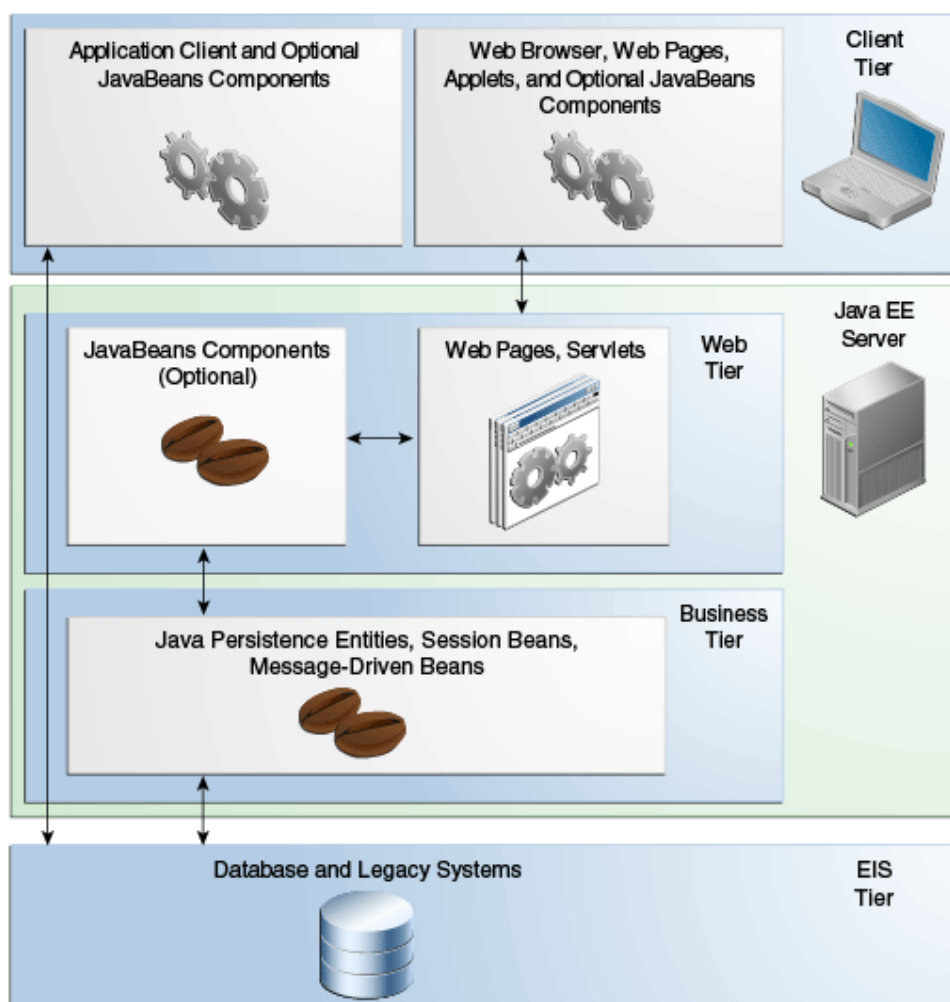
Jakarta EE je souhrn specifikací, který umožňuje světové komunitě vývojářů vyvíjet jednotlivé navzájem kompatibilní implementace [10]. Ve finále se tedy jedná o velkou platformu ulehčující vývoj enterprise aplikací. Celá tato platforma nabízí nepřehledné množství nástrojů, frameworků a knihoven, které poskytují velmi ucelené prostředí pro vývoj od základu dobře navržených systémů.

*Obecně se všechny Jakarta EE specifikace skládají z:*

- *API a dokument specifikace – definuje a popisuje samotnou specifikaci*

- *TCK (Technology Compatibility Kit)* – slouží k testování jednotlivých implementací založených na výše zmíněných API a specifikačním dokumentu
- *Kompatibilní implementace* – ty implementace, které úspěšně splnily testy TCK [10]

Velmi pěkný přehled nabízí obrázek 5.1. Ukazuje architekturu vícevrstvé webové aplikace na platformě Jakarta EE. Je na něm vidět, jak je vše promyšleně rozdělené, které vrstvy s čím komunikují a v podstatě přesně popisuje principy, kterých jsem se držel při implementaci. Často se tedy na tento diagram budu dále odkazovat z následujících dílčích podkapitol, kde popisují jednotlivé technologie zapadající do tohoto ekosystému.



**Obrázek 5.1:** Jakarta EE 8 – Diagram architektury, převzato z [7]



## ■ CDI

Řízení životního cyklu samotné aplikace je často vhodné delegovat na nějaký framework, který je pro to většinou velmi dobře navržený. Kód, který je pak potřeba napsat, se v podstatě stává jakousi knihovnou pro tento framework. A přesně k tomuto účelu slouží právě CDI (Contexts and Dependency Injection).

## ■ Jakarta Security 2.0

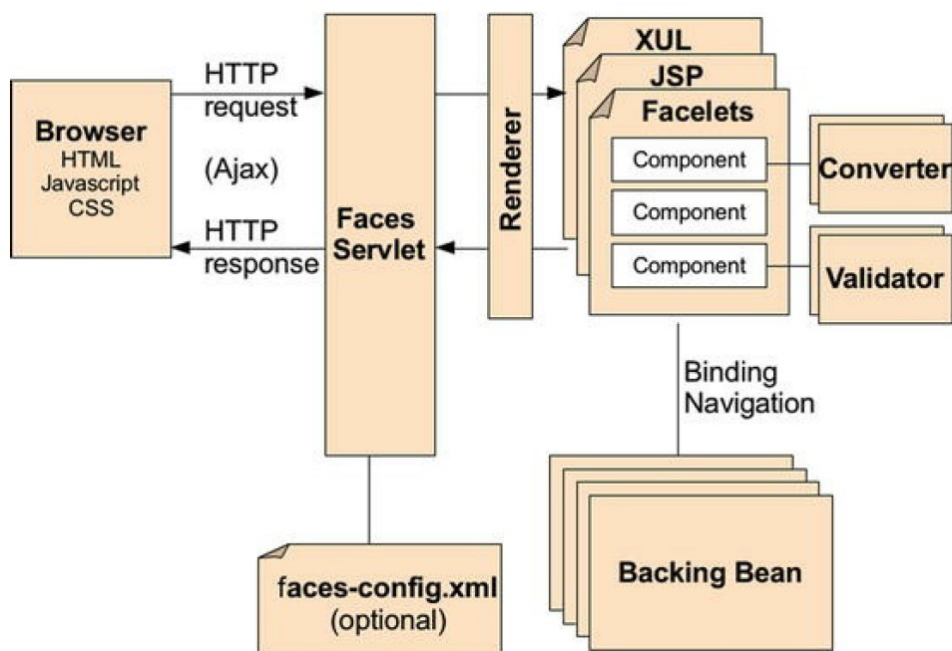
V aplikaci jsem také musel řešit jak autentizaci, tak i autorizaci. Pro tyto účely jsem využil standard Jakarta Security 2.0, pro který má skvělou podporu aplikační server Payara, o kterém se zmiňuji dále. Konkrétně jsem využil JDBC Security realm, což je mechanismus, který umožňuje přímé propojení s databází a díky kterému je možné zabezpečit různé webové prostředky pouze za pomoci konfigurace (vizte příklad konfigurace 5.1). V té je možno mimo jiné nadefinovat například i různé role a jejich oprávnění, kam mohou přistupovat.

```

1      <login-config>
2          <auth-method>FORM</auth-method>
3          <realm-name>onlineEduRealm</realm-name>
4          <form-login-config>
5              <form-login-page>/login.xhtml</form-login-page>
6              <form-error-page>/login.xhtml</form-error-page>
7          </form-login-config>
8      </login-config>
9      <security-role>
10         <description>student</description>
11         <role-name>student</role-name>
12     </security-role>
13     <security-constraint>
14         <display-name>Student pages</display-name>
15         <web-resource-collection>
16             <web-resource-name>student</web-resource-name>
17             <url-pattern>/student/*</url-pattern>
18         </web-resource-collection>
19         <auth-constraint>
20             <role-name>student</role-name>
21         </auth-constraint>
22     </security-constraint>
23 
```

**Listing 5.1:** Zjednodušený příklad Jakarta Security 2.0 konfigurace





**Obrázek 5.2:** Zjednodušená architektura technologie JSF, převzato z [8]

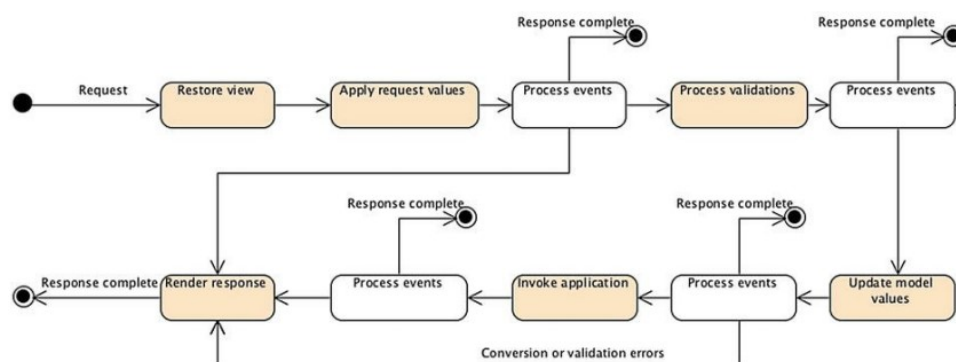
*komponent, umožňují jednoduše definovat rozhraní pro webové služby a podobně. Samozřejmě jsou také velmi dobře integrované s dalšími Jakarta EE technologiemi, jako například JDBC, JPA, atd. Proto slouží mimo jiné jako vstupní bod k byznys logice například z JSF, tedy z prezentační vrstvy.* [8]

V praxi pak tedy stačí například označit třídu anotací `@Stateless` a je vyřešena spousta práce. Například se stane to, že všechny metody v této třídě budou vykonány jako samostatné transakce, což je ideální řešení například pro přístup k databázi.

## ■ Jakarta WebSocket 2.0

Při implementaci třídního chatu jsem řešil problém duplexní komunikace. Nakonec jsem k tomuto účelu použil Jakarta WebSocket 2.0. Jedná se o API specifikaci umožňující vývojáři do aplikace integrovat WebSockets. [12]

WebSocket je protokol umožňující perzistentní TCP spojení mezi serverem a klientem, takže si mohou data posílat libovolně. [13]



Obrázek 5.3: Životní cyklus JSF, převzato z [8]

### ■ 5.1.2 Primefaces

Dále jsem využil knihovnu Primefaces verze 10. Ta nabízí spoustu předpřipravených konfigurovatelných komponent. Nějaké jsou samozřejmě již součástí JSF, nicméně Primefaces poskytují mnohem propracovanější řešení. Například u tabulek je možné nastavit různé filtrování, řazení, styly a podobně. Za zmínku stojí také například Richtext editor, který poskytuje v podstatě všechny důležité možnosti formátování a vkládání obrázků.

### ■ 5.1.3 PostgreSQL

Pro ukládání dat jsem se rozhodl využít osvědčenou relační databázi PostgreSQL [14]. Hlavním důvodem je skvělá podpora ze strany Jakarta EE a také fakt, že nám tato technologie byla často doporučována během různých předmětů ve škole. Pro lepší práci s daty jsem využil ORM ve formě „SQL to Code“. Jinými slovy jsem pomocí jazyka SQL nejprve vytvořil samotnou databázi a z té jsem posléze vygeneroval Java třídy včetně anotací. Tento přístup se mi zdál přirozenější jelikož jsem již znal základy jazyka SQL.

### ■ 5.1.4 Payara Server

Jako aplikační server jsem zvolil Payara Server [15], který vychází z GlassFish Server Open Source Edition. Hlavními důvody bylo, že je nabízen zdarma a má otevřený zdrojový kód. Navíc se snaží držet standardů.

### 5.1.5 Docker

Co se týče běhového prostředí, zde jsem se rozhodl pro kontejnerovou technologii Docker [16]. Ve stručnosti lze Docker chápat jako izolaci procesů. Narozdíl od klasické virtualizace tedy není potřeba virtualizovat celý operační systém, jelikož se využívá kernel samotného hostujícího OS. Důsledkem je to, že start takového kontejneru je téměř okamžitý a pro jeho běh je potřeba RAM a výpočetní prostředky pouze pro samotnou aplikaci, která je v něm izolována.

Konkrétně na produkčním prostředí běží celkem 2 kontejnery. A to Payara server se samotnou aplikací a dále PostgreSQL databáze.

### 5.1.6 NGINX Reverse Proxy Server

Pro zajištění https spojení jsem využil server NGINX Reverse Proxy [17]. Samotnou konfiguraci reverzní proxy lze vidět v ukázce 5.2.

```

1 <IfModule mod_ssl.c>
2   <VirtualHost *:443>
3     ServerName online-edu.aubrecht.net
4     ServerAdmin webmaster@localhost
5     DocumentRoot /var/www/online-edu
6     ErrorLog ${APACHE_LOG_DIR}/online-edu-error.log
7     CustomLog ${APACHE_LOG_DIR}/online-edu-access.log combined
8     ProxyPass / http://127.0.0.1:8780/
9     ProxyPassReverse / http://127.0.0.1:8780/
10    ProxyPassReverseCookieDomain 127.0.0.1 online-edu.aubrecht.
11    net
12    RewriteEngine on
13    RewriteCond %{HTTP:Upgrade} websocket [NC]
14    RewriteCond %{HTTP:Connection} Upgrade [NC]
15    RewriteRule ^/omnifaces.push/(.*) "ws://127.0.0.1:8780/
16    omnifaces.push/$1" [P,L]
17    Header onsuccess edit Set-Cookie ^(.*)$ "$1; Secure"
18    ProxyRequests off # Disable forward proxying
19    SSLCertificateFile /etc/letsencrypt/live/online-edu.aubrecht
20    .net/fullchain.pem
21    SSLCertificateKeyFile /etc/letsencrypt/live/online-edu.
22    aubrecht.net/privkey.pem
23    Include /etc/letsencrypt/options-ssl-apache.conf
24  </VirtualHost>
25</IfModule>

```

**Listing 5.2:** Ukázka konfigurace reverzní proxy

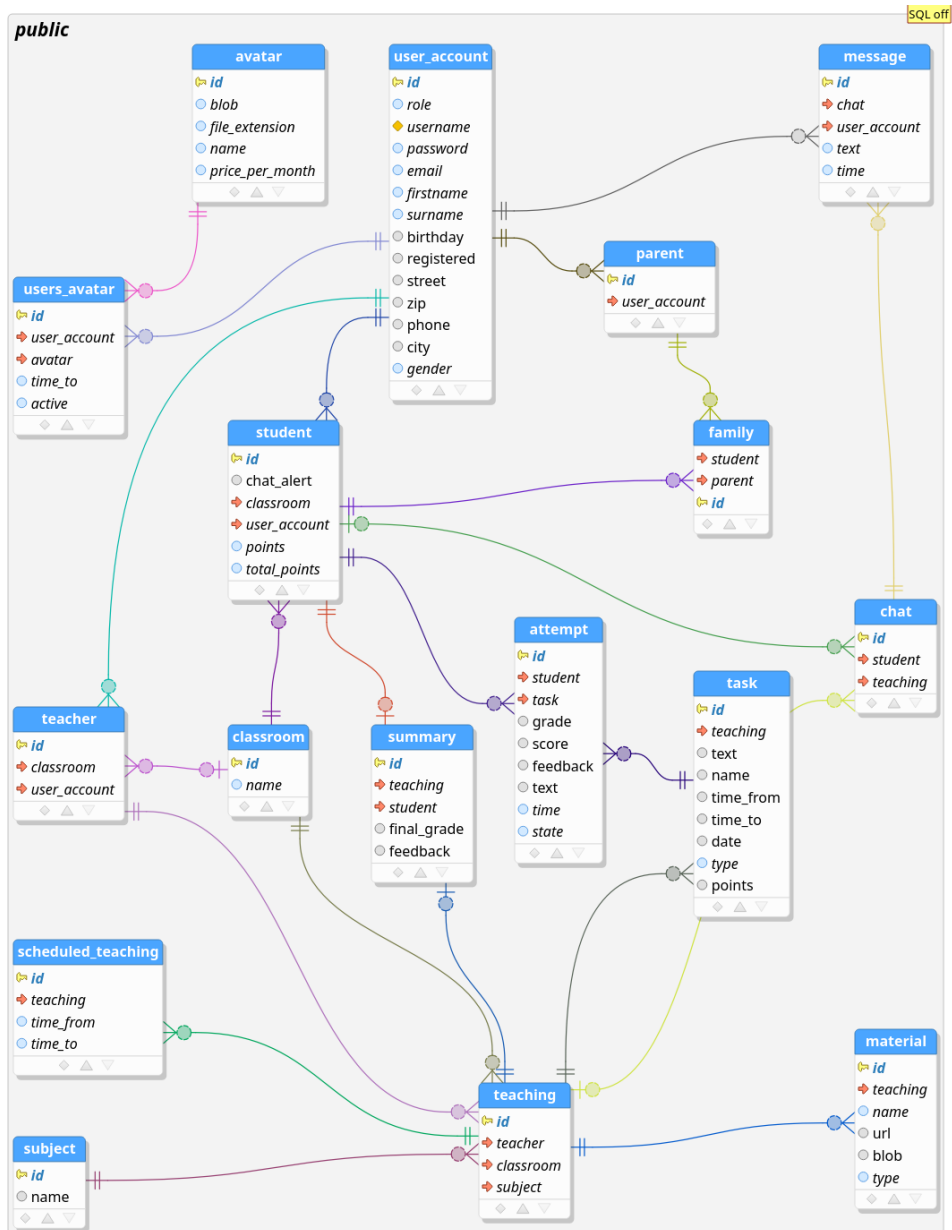
## ■ 5.2 Datový model

Jako základní kámen, od kterého se následně odvíjely další činnosti, je datový model. Jedná se o souhrn entit, které v systému vystupují a vztahy mezi nimi.

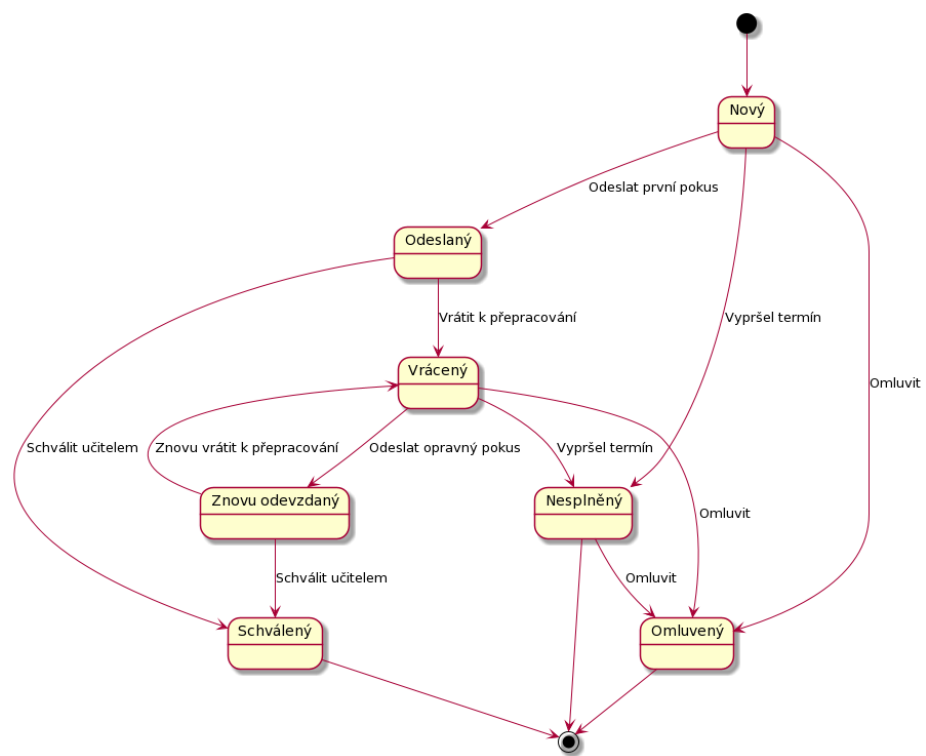
*„Do určité míry lze na dobře sestavený datový model nahlížet jako na funkční prototyp informačního systému, který budou později programátoři dělat.“ [18]*

## ■ 5.3 Stavový diagram úkolu

Jednou z hlavních entit v systému je nepochybně samotný úkol, který učitel nebo učitelka zadává třídě k vypracování. Takový úkol může být buďto povinný a nebo dobrovolný za bonusové body. Ať už se jedná o jeden, či druhý typ úkolu, může se nacházet v různých stavech. Tyto stavy jsou vždy myšleny vzhledem ke konkrétnímu žákovi. Bylo by tedy možná přesněji říkat stav odevzdání konkrétního úkolu u konkrétního žáka. Aby tedy bylo možné rozhodnout stav tohoto úkolu ke konkrétnímu žákovi, je nutná nějaká třetí entita, která bude tento stav určovat. A tou je odeslaný pokus. Těch může žák k jednomu úkolu poslat hned několik, pokud je mu pokaždé vrácen k přepracování. Jakmile je některý z odeslaných pokusů učitelem schválen, tak je tím žákovi schválen celý úkol. Mimo to může vyučující žákovi úkol omluvit a to i v případě, že již vypršel termín odevzdání a žák má tento úkol již ve stavu nesplněný. Souhrnně tyto stavy vysvětluje diagram 5.5



Obrázek 5.4: Datový model



Obrázek 5.5: Stavový diagram úkolu





## Kapitola 6

### Gamifikace

Nad rámec zadání jsem se vynasnažil do systému dostat různé herní prvky. Důvodem je fakt, že se jedná o platformu určenou pro základní školy, a tak se jeví jako dobrý nápad tzv. Škola hrou, kterou prosazoval již Jan Amos Komenský. Cílem je tedy implementace různých zábavných a herních prvků, které by mohly děti motivovat k práci nenásilnou formou.

Jako formu gamifikace jsem zvolil prvky detailně rozebrané v následujících podkapitolách.



#### 6.1 Body

Úplným základem je získávání bodů za dobrovolná cvičení. Právě fakt, že je lze získat pouze za dobrovolné úkoly je velmi důležitý z psychologického hlediska. Pokud by totiž bylo možné získávat body za povinné úkoly pouze na základě toho, jak je jednotlivec chytrý, tak by to mohlo vést k pocitu marnosti u slabších jedinců.

Když je ovšem bude možné získat pouze za dobrovolné úkoly, tak to úplně mění situaci. To, že má někdo více bodů, pak totiž nemusí znamenat, že by byl chytřejší, ale spíše že je snaživější a dělá věci i navíc.

## 6.2 Žebříček

Samotné body by byly víceméně k ničemu, kdyby neexistovalo nějaké srovnání mezi spolužáky, které by vyvolávalo soutěživost. Právě k tomuto účelu slouží třídní žebříček, kde se každý může podívat, kolik kdo získal bodů.

Rozhodně je ale potřeba tříde od začátku vysvětlovat, že zde má každý stejnou šanci a záleží pouze na snaze. A že i když se někdo na chvíli ocitne vespu, nic mu nebrání dodělat nějaké starší cvičení a dostat se tak výše.

## 6.3 Avatar

Nakonec je opět z psychologického hlediska důležité i to, aby body prakticky k něčemu sloužily, například jako měna.

A tak vznikl nápad, že by žáci měli svého avatara, kterého by si za nasbírané body mohli vylepšovat. Takové vylepšení bude vždy na omezenou dobu, například 1 měsíc, aby stále existovala motivace pro průběžnou práci. Kdyby totiž žák získal všechny vylepšení příliš brzy, tak by pak mohl přestat pracovat.

Předměty, které si mohou žáci na svého avatara dokoupit mohou být velmi různorodé. Může se jednat například o ozdobné rámečky, brýle, čepice nebo klidně i celé, úplně nové avatary. Určitě je v této oblasti podle mého názoru obrovský potenciál, a rozhodně se tedy nebráním do budoucna spolupráci s grafikem či zapálenou komunitou. Příklad použitých avatarů lze vidět na obrázku 6.1.

## 6.4 Třídní chat

Aby žák svého avatara někde využil a ukázal jej ostatním, implementoval jsem také třídní chat. Do toho mohou žáci a učitelé psát zprávy ozdobené vlastním avatarem. Z technického hlediska se jedná o plnohodnotný chat s ukládáním zpráv do databáze. Také jsem rád, že jsem si zde vyzkoušel technologii WebSocket (vizte kapitolu Architektura).



**Obrázek 6.1:** Ukázky avatarů, staženo pro nekomerční použití z [9]



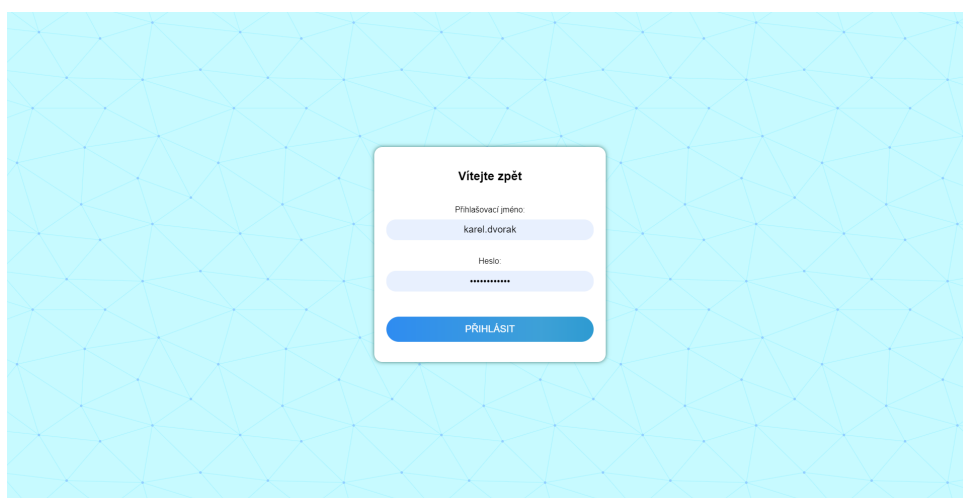
# Kapitola 7

## Uživatelské rozhraní

V této sekci se nachází ukázky uživatelského rozhraní z již existující implementace.

### 7.1 Přihlašovací obrazovka

Přihlašování do systému probíhá pomocí uživatelského jména a hesla v jednoduchém formuláři. Roli uživatele pak systém rozpozná sám.



Obrázek 7.1: Přihlašovací obrazovka

## 7.2 Rozhraní pro učitele

Učitelenské rozhraní nabízí snadné přepínání mezi třídami a vyučovanými předměty v dané třídě. Na obrázku 7.2 je pak konkrétně vidět seznam zadaných úkolů. Ty lze řadit podle různých sloupců a vytvářet úkoly nové. Každý z nich lze následně otevřít.

The screenshot shows the 'Úkoly' (Tasks) section of the 'Online Edu' system. The interface is divided into a left sidebar with navigation options like 'Třídní chat', 'Úkoly (6)', 'Ověření za body (3)', 'Seznam žáků', 'Třídní kniha', and 'Novinky (9)'. The main content area displays a table of tasks with columns for 'Název T1', 'Nedokončeno T1', 'Odevzdáno T1', 'Schváleno T1', 'Nesplněno T1', and 'Termín odevzdání T1'. Below the table is a 'Vytvořit úkol' (Create task) form with fields for 'Jméno úkolu', 'Začátek', and 'Deadline', and a rich text editor for the task description.

Název T1	Nedokončeno T1	Odevzdáno T1	Schváleno T1	Nesplněno T1	Termín odevzdání T1
Počítání příkladů	5		1		20. 05. 2021 14:24
Slovní úloha	5	1			31. 05. 2021 14:33
Násobení	3	3			11. 06. 2021 14:33
Dělení	3	2	1		13. 05. 2021 14:34

Obrázek 7.2: Učitel – seznam úkolů

Po otevření konkrétního úkolu se zobrazí jeho detail, jak je ukázáno na obrázku 7.3. Tam je pro každého žáka k dispozici seznam jeho pokusů, kde každý pokus lze opět otevřít.

The screenshot shows the 'Úkol: Dělení' (Task: Division) page in the Online Edu system. The interface includes a top navigation bar with subject tabs: Matematika (7), Český jazyk (0), Anglický jazyk (0), Prvouka (0), and Informatika (0). A left sidebar contains navigation options like 'Třídní chat', 'Úkoly (5)', 'Cvičení za body (2)', 'Seznam žáků', 'Třídní kniha', and 'Novinky (7)'. The main content area displays a table of student attempts:

Student	Stav LP	Action
> Nová Marie	Neúspěšně	Zobrazit poslední pokus
> Dvořák Karel	Schváleno	Zobrazit poslední pokus
> Malý Jiří	Odevzdáno	Zobrazit poslední pokus
> Novotný Ladislav	Nový	
> Mrázková Michaela	Nový	
> Dolejší Patricie	Nový	

Below the table is the 'Upravit zadání úkolu' (Edit task assignment) section, which includes a rich text editor with a toolbar and a text area containing '8 - 2 ='. A 'Uložit' (Save) button is located at the bottom left of this section. The footer of the page reads 'Vytvořeno pro studijní účely'.

Obrázek 7.3: Učitel – detail úkolu

Na stránce detailu pokusu na obrázku 7.4 lze vidět samotnou odpověď žáka a také prostor pro zpětnou vazbu. Tento pokus lze buďto schválit a tím schválit žákovi celý úkol nebo ho vrátit k přepracování.

The screenshot shows the 'Zadání úkolu (Počítání příkladů)' (Task assignment: Arithmetic examples) page. The top navigation bar is similar to the previous screenshot, but the 'Matematika' tab now shows 11 items. The left sidebar also reflects this change. The main content area displays the task details:

**Zadání úkolu (Počítání příkladů)** Odevzdaný pokus (Odevzdáno) | 30. 04. 2021 14:41

Vypočítejte:

$2 + 3 =$	5
$8 - 1 =$	7
$13 + 8 =$	21

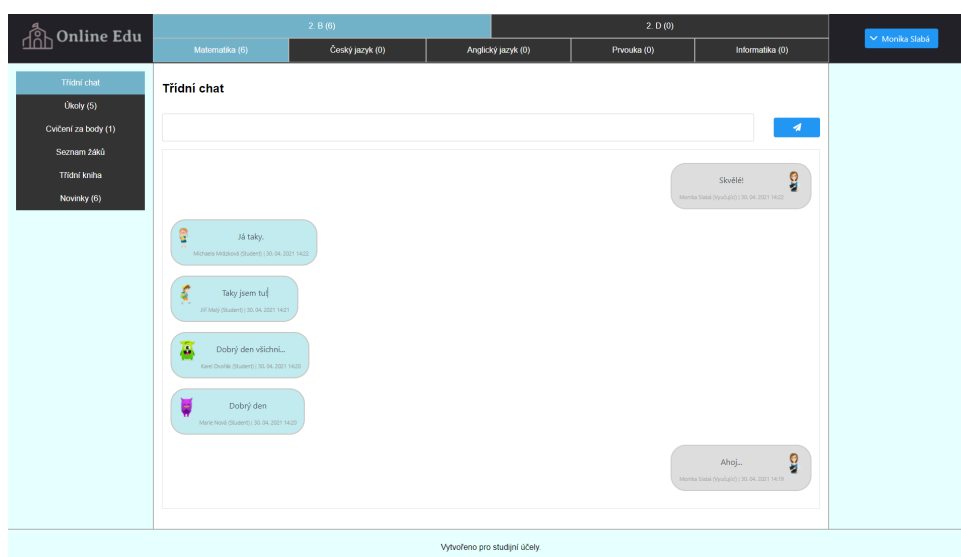
Below the task is the 'Zpětná vazba' (Feedback) section, which includes a rich text editor with a toolbar and a text area containing 'Správně! 😊'. At the bottom of this section are two buttons: 'Schválit' (Approve) and 'Vrátit k přepracování' (Return to rework).

The footer of the page reads 'Vytvořeno pro studijní účely'.

Obrázek 7.4: Učitel – detail pokusu

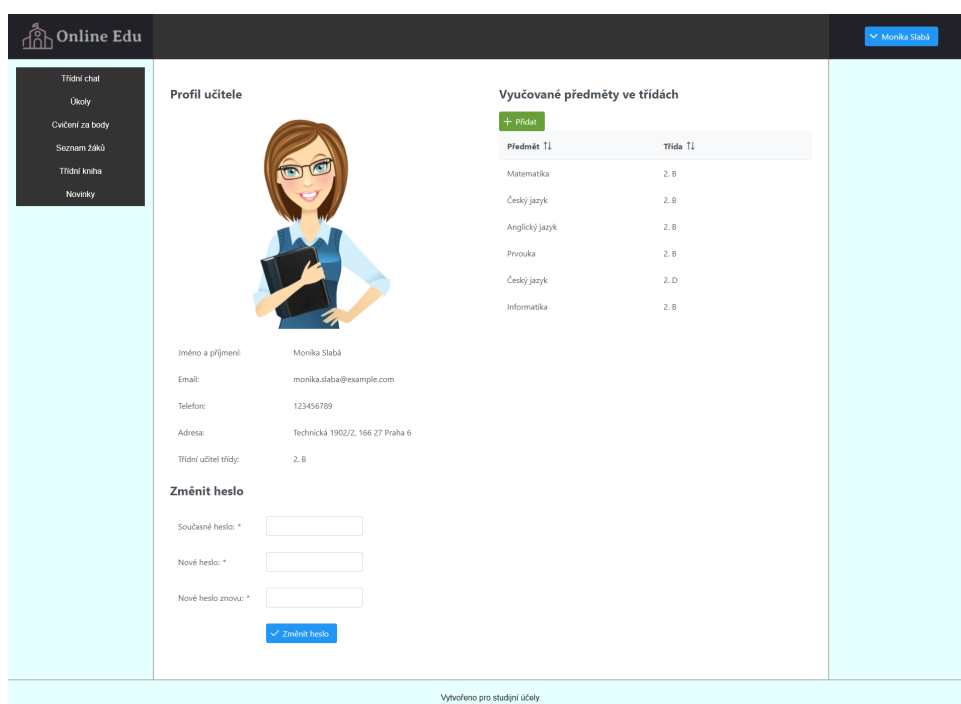
Na obrázku 7.5 je vidět třídní chat. Ten vypadá stejně z pohledu žáka i učitele.

## 7. Uživatelské rozhraní



Obrázek 7.5: Učitel – chat

Učitel si také může například měnit své heslo na stránce svého profilu na obrázku 7.6.



Obrázek 7.6: Učitel – profil



## 7.3 Rozhraní pro žáky

Žákovské rozhraní se velmi podobá tomu učitelskému, jen nabízí jiné funkce. Například žák je fixně ve své třídě a může přepínat jen mezi svými předměty. Konkrétně na stránce seznam úkolů na obrázku 7.7 vidí seznam všech úkolů v daném předmětu, kde na každý může kliknout a zobrazit si jeho detail.

The screenshot shows the 'Úkoly' (Tasks) page for a student. The top navigation bar includes 'Online Edu' and subject tabs: Matematika (0), Český jazyk (0), Anglický jazyk (0), Píseňka (0), and Informatika (0). The user is logged in as 'Karel Dvořák'. On the left, there is a sidebar with 'Třídní chat', 'Úkoly (4)', 'Cvičení za body (2)', 'Žebříček', and 'Avatari'. The main content area is titled 'Úkoly' and has tabs for 'Nesplněné', 'Odeslané', and 'Hotové'. A 'Vybrat skupost' button is visible. Below is a table of tasks:

Název T1	Stav T1	Termín odevzdání T1	
Násobení	Nový	11. 06. 2021 14:33	<a href="#">Detail</a>
Dělení	Nový	13. 05. 2021 14:34	<a href="#">Detail</a>
Počítání příkladů	Nový	20. 05. 2021 14:24	<a href="#">Detail</a>
Slovní úloha	Nový	31. 05. 2021 14:33	<a href="#">Detail</a>

At the bottom, it says 'Vytvořeno pro studijní účely'.

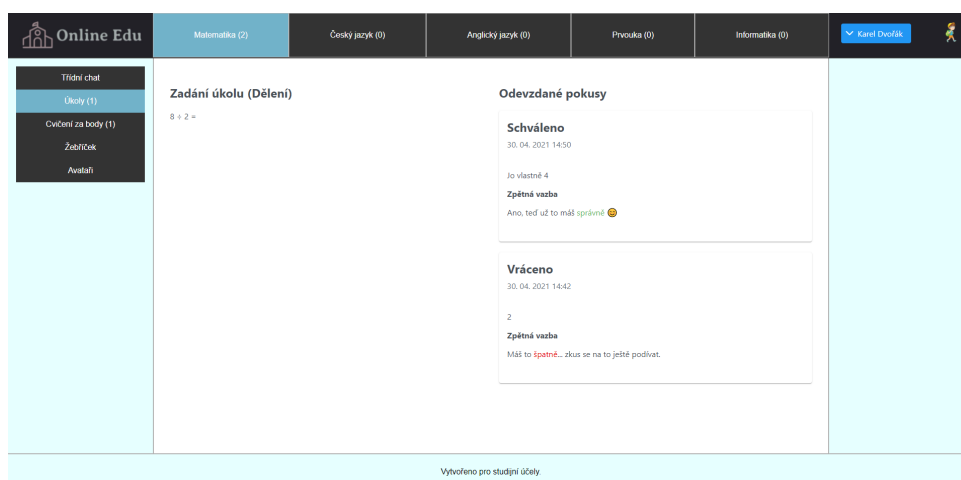
Obrázek 7.7: Student – seznam úkolů

Po zobrazení detailu úkolu k němu může odeslat řešení jak je vidět na obrázku 7.8.

The screenshot shows the 'Zadání úkolu (Počítání příkladů)' (Task assignment) page. The task is 'Vypočítejte:' (Calculate:). The problem is shown as a handwritten-style image with three equations:  $2 + 3 =$ ,  $8 - 1 =$ , and  $13 + 8 =$ . Below the problem is a 'Nový pokus' (New attempt) section with a rich text editor. The editor contains the numbers 5, 7, and 21. At the bottom, there is an 'Odeslat' (Submit) button. The page also shows 'Odevzdané pokusy' (Submitted attempts) on the right. At the bottom, it says 'Vytvořeno pro studijní účely'.

Obrázek 7.8: Student – nový pokus

Jak vidí žák detail schváleného úkolu je vidět na obrázku 7.9. Nemá zde již možnost posílat další pokusy a v pravé části vidí náhledy odeslaných pokusů včetně zpětné vazby.



Obrázek 7.9: Student – schválený úkol

Jak popisuji v kapitole 6, rozhodl jsem se do systému nad rámec řešení implementovat gamifikační prvky. S tím souvisí nakupování avatarů na obrázku 7.10, kde si žák může za své body kupovat avatary a následně si volit, kterého chce zrovna používat. S tím také souvisí třídní žebříček, kde si může žák prohlédnout avatary svých spolužáků na obrázku 7.11

**Avataři**  
Bodů k dispozici: 19

Název	Obrázek	Cena za měsíc	Vypří za	
Black		10		<a href="#">Koupit</a>
Blue		20		<a href="#">Koupit</a>
Brown		15		<a href="#">Koupit</a>
Cat		25		<a href="#">Koupit</a>
Dark Blue		5	29 dní   23 hodin   59 minut	<a href="#">Prodloužit</a>
Light Blue		40		<a href="#">Koupit</a>
Lion King		100		<a href="#">Koupit</a>
Magenta		40	29 dní   23 hodin   59 minut	<a href="#">Prodloužit</a>
Orange		15		<a href="#">Koupit</a>
Pink		10		<a href="#">Koupit</a>
Sitch		80		<a href="#">Koupit</a>
Green		8	60 dní   23 hodin   59 minut	<a href="#">Prodloužit</a>

[Nastavit základního avatara](#)

Vyhofeno pro studijní účely

Obrázek 7.10: Student – avataři

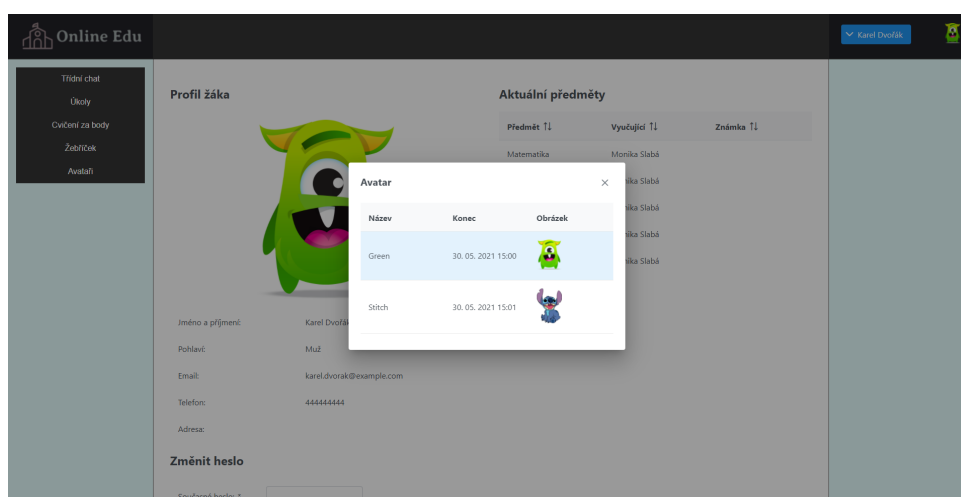
**Žebříček**

Body	Jméno a příjmení	Avatar
90	Karel Dvořák	
80	Marie Nová	
0	Laďslav Novotný	
0	Jiří Malý	
0	Michaela Mláčková	
0	Patrice Dolejší	

Vyhofeno pro studijní účely

Obrázek 7.11: Student – žebříček

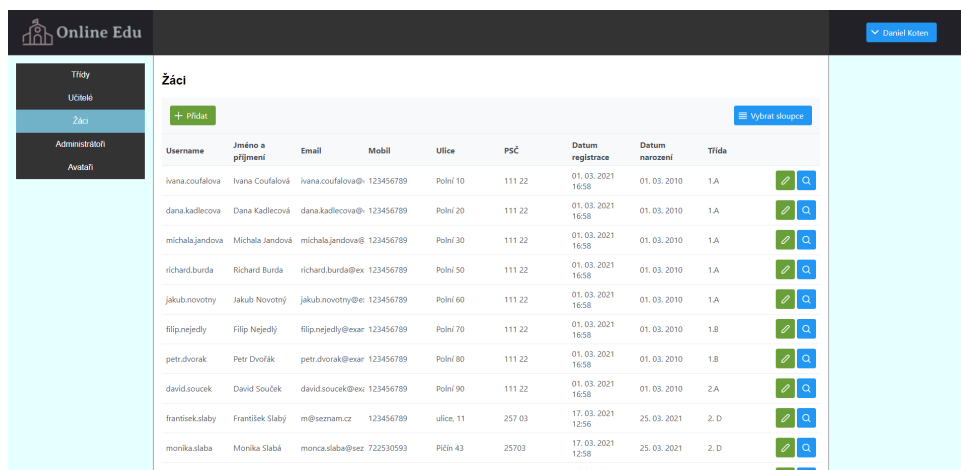
Ze svých vlastněných avatarů může student také vybírat přímo na své profilové stránce na obrázku 7.12. Mimo to si zde i může měnit své heslo.



Obrázek 7.12: Student – profil

## 7.4 Rozhraní pro administrátory

Administrátorské rozhraní umožňuje v podstatě vytvářet a upravovat potřebné entity v systému. Například na obrázku 7.13 je vidět seznam všech žáků, kde každého z nich lze upravit a nebo vytvářet nové.



Obrázek 7.13: Administrátor – seznam žáků

## Kapitola 8

### Průchody aplikací

V této kapitole uvedu několik průchodů aplikací pro nejdůležitější případy užití a také jejich vzájemné porovnání v různých verzích implementace. Jelikož by měl být systém v první řadě co nejjednodušší, tak jsem cítil potřebu tuto jednoduchost také nějak měřit. Samozřejmě, že míra jednoduchosti je z velké části velmi subjektivní a co se někomu může zdát snadné a přehledné, tak pro druhého může být příliš komplexní. Nicméně jako takovou zjednodušenou a vyčíslitelnou míru jednoduchosti jsem použil počet pomyslných kroků pro dosažení určitého cíle. Například počet kliknutí, vyplnění polí, nutnost najít řádek v dlouhé tabulce atd.

První verzi je myšlena verze po dokončení předmětu Semestrální projekt. Tato verze byla sice z velké části kompletní z pohledu funkcionalit, nicméně ještě obsahovala spoustu nedokonalostí, zejména co se týče uživatelského komfortu. Spousta činností byla řešena velmi těžkopádně, zdlouhavě a neintuitivně.

Druhá verze je finální řešení po zaimplementování komentářů a postřehů, které vzešly z konzultací s vedoucím práce, z mé vlastní uživatelské zkušenosti a v neposlední řadě od kamarádů a známých učitelek, které mi pomohly s uživatelským testováním. V této verzi se ohledně počtů kroků počítá s tím, že má uživatel konkrétní stránku uloženou v záložkách prohlížeče.

#### 8.1 Zadání nového úkolu

Pro zadání úkolu nyní stačí pouze přejít na příslušnou stránku uloženou v záložkách, vyplnit text zadání a přidat jej.

První verze	Druhá verze
Kliknout na tlačítko přihlášení	Přejít na stránku seznam úkolů
Zadat údaje	Vyplnit zadání a vytvořit nový úkol
Přihlásit se	
Přejít na hlavní stránku učitele	
Zobrazit seznam úkolů	
Vyplnit zadání a vytvořit nový úkol	

Tabulka 8.1: Zadání nového úkolu

## 8.2 Odevzdání úkolu

Pro přidání řešení již není nutné hledat konkrétní položku v seznamu všech úkolů, ale stačí rovnou přejít na seznam těch nedokončených.

První verze	Druhá verze
Kliknout na tlačítko přihlášení	Přejít na stránku seznam nedokončených úkolů
Zadat údaje	Zvolit úkol
Přihlásit se	Vyplnit a odeslat řešení
Přejít na hlavní stránku žáka	
Zobrazit seznam všech úkolů	
Najít v tabulce nový úkol	
Zvolit úkol	
Vyplnit a odeslat řešení	

Tabulka 8.2: Odevzdání úkolu

## 8.3 Schválení nebo vrácení pokusu

Tento bod prošel asi největším vylepšením. Stačí teď pouze přejít na seznam aktuálních úkolů, zobrazit si pouze ty žáky, kteří odeslali nějaké nové řešení a to zkontrolovat. Nebo je možná ještě kratší varianta přejít na novinky, kde se ukazují posledních 10 odeslaných řešení a zobrazit si ho odtud.

První verze	Druhá verze
Kliknout na tlačítko přihlášení	Přejít na seznam úkolů ke kontrole
Zadat údaje	Zvolit úkol
Přihlásit se	Přejít na žákovo poslední odevzdaný pokus
Přejít na hlavní stránku učitele	Napsat zpětnou vazbu
Zobrazit seznam všech úkolů	Schválit nebo vrátit
Najít v tabulce úkol u kterého přibyly pokusy	
Zvolit úkol	
Podívat se na zadání úkolu	
Rozkliknout žákovo pokusy	
Přejít na poslední odevzdaný pokus	
Napsat zpětnou vazbu	
Schválit nebo vrátit	

**Tabulka 8.3:** Schválení nebo vrácení pokusu

## 8.4 Opravné odevzdání úkolu

Nakonec stojí za zmínku vylepšené opravné odeslání řešení. V první verzi bylo nutné z detailu úkolu přejít přímo na stránku detail pokusu, zde si přečíst zpětnou vazbu, vrátit se a odeslat nové řešení. V nové verzi je vše na jednom místě přímo v detailu úkolu. Zejména minulé pokusy včetně zpětné vazby lze prohlížet v chronologickém pořadí z jednoho místa.

První verze	Druhá verze
Kliknout na tlačítko přihlášení	Přejít na stránku seznam nedokončených úkolů
Zadat údaje	Zvolit úkol
Přihlásit se	Přečíst si poslední komentář ve sloupci s minulými pokusy
Přejít na hlavní stránku žáka	Vyplnit nové řešení a odeslat
Zobrazit seznam všech úkolů	
Najít v tabulce vrácený úkol	
Zvolit úkol	
Přejít na pokus, který byl vrácen k přepracování	
Přečíst si komentář	
Vrátit se zpět na zadání úkolu	
Vyplnit nové řešení a odeslat	

**Tabulka 8.4:** Opravné odevzdání úkolu







## Kapitola 9

### Testování

Pro účely testování jsem po dohodě se svým vedoucím využil nástroj pro automatické testování webových aplikací Selenium IDE [19]. Samotné testy se dají vytvářet tak, že se v podstatě zaznamenají provedené akce na stránce a v budoucnu se již provedou automaticky. Samozřejmě je možné přidávat i kroky pro kontrolu, zda se na stránce nachází nějaký element nebo zda je v určitém poli správná hodnota.

Ve své práci jsem vytvořil 15 různých akceptačních testů, které testují nejdůležitější případy užití. Jednoduchou ukázkou lze vidět na obrázku 9.1.

Testy se nejprve zkusí přihlásit jako administrátor a vytvořit několik entit, jako například třídy, učitele a studenty. Nově vytvořeným uživatelům nastaví nové heslo a následně se odhlásí.

Dále se pokusí přihlásit jako učitel a vytvořit několik nových úkolů. Poté, co tyto úkoly jako student vyplní a odešle, tak některé schválí a některé vrátí k přepracování. Po opětovném odevzdání vrácených úkolů i tyto schválí.

Po úspěšném projití všech těchto testů je ověřena většina případů užití.

**Selenium IDE - onlineedu**

Project: onlineedu

Tests +

Search tests...

http://localhost:8780

	Command	Target	Value
1	open	/admin/students	
2	set window size	957x919	
3	click	css=#form:3A dt-students:3A bt n-add-student > .ui-button-text	
4	click	id=form:firstname	
5	type	id=form:firstname	Jane
6	type	id=form:surname	Roe
7	click	id=form:gender_label	
8	click	id=form:gender_2	
9	click	id=form:phone	
10	type	id=form:phone	111111111
11	type	id=form:email	jane@example.com
12	type	id=form:street	Technická 1902/2
13	type	id=form:zip	166 27
14	type	id=form:city	Praha 6 - Dejvice-Praha 6
15	type	id=form:birthday_input	11.11.2005

Command: open

Target: /admin/students

Value:

Description:

Log Reference

Running '001 create classroom'

1. open on /home OK 17:12:02
2. setWindowSize on 1070x889 OK 17:12:02
3. click on css=p-lg-12:nth-child(3) .login-box-icon OK 17:12:03
4. click on name=j\_username OK 17:12:05
5. type on name=j\_username with value admin OK 17:12:06
6. type on name=j\_password with value admin OK 17:12:06

Obrázek 9.1: Ukázka testů v Selenium IDE

## Kapitola 10

### Závěr

Ve své práci jsem se soustředil zejména na práci s úkoly a jejich stavy, což je častý problém současných systémů. Nesnažil jsem se o kompletní řešení celé problematiky. Spíše jsem chtěl ukázat možný způsob implementace speciálně pro danou oblast.

Hlavním požadavkem byla jednoduchost. Na obrázku 7.2 na stránce 44 je vidět příklad toho, jak jsem si s tímto bodem poradil. Maximální přehlednosti zde přispívá zejména rozvržení uživatelského rozhraní. V horní části lze jednoduše přepínat mezi třídami a předměty a pro každou tuto kombinaci lze pak v levém menu volit různé stránky. Uživatel také v každé chvíli přesně ví, kde se nachází díky barevnému zvýraznění v navigačních menu.

Kromě samotné navigace nabízí obě menu i notifikace. U každé třídy a předmětu vidí uživatel počty záležitostí, které by měl vyřešit (např. počty úkolů k opravení). Díky tomu by se pak nemělo stávat, že někdo na něco zapomene. Jednoduše se učitel či žák přihlásí a ihned ví, kam kliknout.

Neméně důležité bylo také potřeba rozmyslet, jak bude systém s úkoly, zejména jejich stavy, vlastně pracovat. Výstižně mé řešení popisuje stavový diagram úkolu 5.5 na stránce 38, kde lze vidět kompletní životní cyklus úkolu. Ve stručnosti je důležité zejména to, že když se úkol nachází v pro uživatele nezajímavém stavu (pro žáka např. stav odesláno), tak tento úkol nevidí v hlavním seznamu úkolů, kde by mělo být opravdu jen to, co je potřeba právě teď udělat.

Dalším velkým problémem, který je potřeba řešit, je registrace uživatelů a vytváření tříd. Já jsem se rozhodl pro to, že tyto entity bude vždy vytvářet nadřizená osoba, která je v daném ohledu většinou vzdělanější. Například

žákům vytvoří účty učitel a těm je zase včetně tříd a předmětů vytvoří administrátor. Z mého pohledu je takový přístup zejména pro žáky mnohem jednodušší a přímočařejší, než aby si museli vytvářet účet sami a následně se přihlašovat do různých předmětů.

Nad rámec zadání jsem také implementoval různé herní prvky včetně třídního chatu (vizte kapitolu 6, Gamifikace). Hlavní pointou je získávání bodů za dobrovolné úkoly, za které si žák následně může vybrat svého avatara. Toho pak používá u svých zpráv v třídním chatu a také v třídním žebříčku.

Zadáním mé bakalářské práce bylo implementovat systém pro zjednodušení úkonů ze strany učitelů a žáků na základních školách. Zejména jsem se měl soustředit na práci s domácími úkoly, a to z uživatelského hlediska v maximální jednoduchosti. Tohoto směru jsem se držel a rozhodně je tedy možné pro tuto oblast začít výsledný systém reálně používat. Co se ovšem týče dalších oblastí, určitě zde vidím spoustu dalších možností k pokračování například formou diplomové práce. Kromě nápadů jako videohovory, rozvrhy, zapisování absence a dalších tradičních funkcí si konkrétně dovedu představit například pokročilejší a dynamičtější formu již zmíněné gamifikace skrze předpřipravené výukové hry, za které by žáci mohli získávat body. Všechna tato rozšíření není do budoucna problém do mé aplikace doimplementovat. Dokonce nic nebrání tomu přidat REST rozhraní a nový modul vyvíjet naprosto nezávisle v jiné technologii.





## Přílohy

# Příloha A

## Literatura

- [1] Masarykova univerzita. Jak pracovat s Google Classroom? <https://kisk.phil.muni.cz/kiskonline/jak-pracovat-s-google-classroom>. (Citováno 27. 4. 2021).
- [2] Bakaláři. <https://www.bakalari.cz/images/index/wtk.png>. (Citováno 26. 4. 2021).
- [3] Škola OnLine. <https://www.skolaonline.cz/Aktuality/Clanky/tabid/436/articleType/ArticleView/articleId/2994/Novy-vzhled-aplikace.aspx>. (Citováno 25. 4. 2021).
- [4] Škola v pyžamu. <https://www.skolavpyzamu.cz/>. (Citováno 28. 4. 2021).
- [5] Online cvičení. [https://www.onlinecviceni.cz/exc/list\\_sel\\_topics.php](https://www.onlinecviceni.cz/exc/list_sel_topics.php). (Citováno 26. 4. 2021).
- [6] Umíme to. <https://www.umimeto.org/>. (Citováno 27. 4. 2021).
- [7] The Jakarta EE 8 – Distributed Multitiered Applications. <https://eclipse-ee4j.github.io/jakartaee-tutorial/overview004.html#BNABJ>. (Citováno 01. 4. 2021).
- [8] Antonio Goncalves. *Beginning Java EE 7*. Apress, 2013.
- [9] PNGEGG. <https://www.pngegg.com/en/png-ezgn1>. (Citováno 11. 3. 2021).
- [10] The Jakarta EE 8. <https://jakarta.ee/about/>. (Citováno 01. 4. 2021).
- [11] The Jakarta EE – JSF. <https://jakarta.ee/specifications/faces/3.0/jakarta-faces-3.0.html>. (Citováno 19. 4. 2021).

- [12] Kasandra Darwin. Jakarta WebSocket. <https://projects.eclipse.org/projects/ee4j.websocket>. (Citováno 25. 2. 2021).
- [13] WebSockets. <https://developer.mozilla.org/en-US/docs/Glossary/WebSockets>. (Citováno 28. 4. 2021).
- [14] PostgreSQL Global Development Group. PostgreSQL. <https://www.postgresql.org/>. (Citováno 21. 4. 2021).
- [15] Payara. <https://www.payara.fish/>. (Citováno 21. 4. 2021).
- [16] Docker. <https://www.docker.com/>. (Citováno 21. 4. 2021).
- [17] NGINX Reverse Proxy. <https://www.nginx.com/resources/glossary/reverse-proxy-server/>. (Citováno 01. 4. 2021).
- [18] Merunka Vojtěch. *Datové modelování*. Alfa Publishing, 2006.
- [19] Selenium IDE. <https://www.selenium.dev/selenium-ide/>. (Citováno 26. 4. 2021).





## Příloha B

### Instalace

Zdrojové soubory lze stáhnout z GitHub repozitáře na adrese <https://github.com/koty10/online-edu>.

Dále bude potřeba JDK 11 (Java Development Kit), který lze stáhnout z <https://www.oracle.com/java/technologies/javase-jdk11-downloads.html>.

Pro zprovoznění aplikace vřele doporučuji nainstalovat kontejnerovou technologii Docker [16] (spolu s Docker Compose). Pak stačí v adresáři `/docker/` pouze spustit tyto skripty.

- `01-build-src-local-docker.sh`
- `02-build-dockers.sh`
- `03-run.sh`

Aplikace, je poté přístupná na adrese <http://localhost:8780/home>.

Pokud jde jen o prohlédnutí běžící aplikace, tak je samozřejmě možné si ji otevřít přímo na „produkčním“ prostředí na adrese <https://online-edu.aubrecht.net/home>.





## Příloha C

### Seznam zkratk

Následující seznam blíže popisuje zkratky použité v práci.

- AJAX Asynchronous JavaScript and XML – časté označení pro asynchronní zpracování webových požadavků bez nutnosti znovunačtení stránky
- API Application Programming Interface – definované rozhraní pro komunikaci s nějakou částí systému
- BE Backend – vrstva pracující s daty a logikou aplikace
- FE Frontend – prezentační vrstva pracující zejména se vzhledem aplikace
- FP Funkční požadavek
- JDBC Java Database Connectivity – rozhraní pro práci s databází v jazyku Java
- JPA Java Persistence API – standard umožňující ORM, existuje několik implementací
- JSF JavaServer Faces
- NP Nefunkční požadavek
- ORM Objektově relační mapování – přístup umožňující práci s tabulkami uložené v relační databázi podobně jako s objekty
- OS Operační systém
- RAM Operační paměť
- REST Representational State Transfer
- TCP Transmission Control Protocol – protokol transportní vrstvy TCP/IP