

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NÁSTROJ PRO EFEKTIVNÍ ŘÍZENÍ ČINNOSTI NA PROJEKTU

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTINA STODOLOVÁ

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NÁSTROJ PRO EFEKTIVNÍ ŘÍZENÍ ČINNOSTI NA PROJEKTU

TOOL FOR EFFECTIVE PROJECT TASK ORGANIZATION

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTINA STODOLOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. VÍTĚZSLAV BERAN, Ph.D.

BRNO 2015

Abstrakt

Tato bakalářská práce se zabývá správou diplomových prací a organizací činnosti při jejich tvorbě. Cílem bylo navrhnout a vytvořit systém pro studenty a vedoucí. Výsledkem je webová aplikace implementovaná v jazyce Ruby s použitím frameworků Rails a Bootstrap. Aplikace vedoucím umožňuje mít přehled o aktivitě studentů prostřednictvím týdenních reportů. Studentům nabízí možnost spolupráce mezi sebou a sdílení informací a motivace.

Abstract

This bachelor's thesis deals with management of bachelor's and master's theses. The main goal was to design and develop system for students and supervisors. The result is a web application implemented in Ruby using Rails and Bootstrap frameworks. The application provides supervisors with overview of students' activity by weekly reports. It allows students to cooperate and share information and motivation.

Klíčová slova

bakalářská práce, diplomová práce, student, vedoucí, správa, web, Ruby, Ruby on Rails, Model-View-Controller, Bootstrap

Keywords

bachelor's thesis, master's thesis, student, supervisor, management, web, Ruby, Ruby on Rails, Model-View-Controller, Bootstrap

Citace

Martina Stodolová: Nástroj pro efektivní řízení činnosti na projektu, bakalářská práce, Brno, FIT VUT v Brně, 2015

Nástroj pro efektivní řízení činnosti na projektu

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Ing. Vítězslava Berana, Ph.D. V práci jsem uvedla veškeré literární prameny a publikace, ze kterých jsem čerpala.

.....
Martina Stodolová
20. května 2015

Poděkování

Ráda bych poděkovala panu Ing. Vítězslavu Beranovi, Ph.D. za konzultace a vedení mé práce.

© Martina Stodolová, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	2
2	Problematika webových aplikací	3
2.1	Webové technologie	3
2.2	Architektura Model-View-Controller	4
2.3	Ruby on Rails	5
2.4	Uživatelské testování	6
3	Systém pro správu diplomových prací	7
3.1	Existující wiki řešení	7
3.2	Specifikace potřeb studentů a vedoucích	8
3.3	Reprezentace dat v datovém modelu	9
3.4	Návrh uživatelského rozhraní	11
4	Aplikace Student & Supervisor	15
4.1	Struktura Ruby on Rails aplikace	15
4.2	Řešení implementačních problémů	17
4.3	Grafické ztvárnění aplikace	20
4.4	Testování systému	21
4.5	Výsledky testování	22
5	Závěr	23
A	Diagram případů užití	25
B	Obsah přiloženého CD	27

Kapitola 1

Úvod

Jako studentky posledního ročníku vysoké školy se mě přímo dotýká problematika psaní diplomové práce. Někdy je těžké se donutit k činnosti na projektu. Některé studenty trápí nedostatek motivace. Mnoho vedoucích diplomových prací by uvítalo lepší možnosti komunikace se studenty. Podobné problémy se snaží řešit tato bakalářská práce.

Cílem mé práce bylo vytvořit aplikaci, která vychází z již existujícího řešení. Jedná se o systém pro správu diplomových prací, který je určen pro studenty a jejich školitele. Systém pomáhá s organizací práce na projektech a také zefektivňuje komunikaci mezi studentem a vedoucím.

Systém jsem navrhla a implementovala jako webovou aplikaci, aby se k ní dalo přistupovat odkudkoliv. Mám s jejich realizací již nějaké zkušenosti a chtěla jsem si rozšířit znalosti v tomto odvětví. Nástroj byl podroben testování uživateli z řad studentů.

Hlavní část technické zprávy je rozčleněna na tři kapitoly. Ve druhé kapitole naleznete přehled technologií a technik pro tvorbu webu. O návrhu aplikace pojednává kapitola 3. Je zde popsáno i původní řešení, ze kterého jsem vycházela. Dále je zde uvedena specifikace požadavků a pár řádků o struktuře dat, která je potřeba ukládat. Čtvrtá kapitola zahrnuje popis zprovoznění aplikace, některé implementační zajímavosti a výsledky testování.

Kapitola 2

Problematika webových aplikací

Rozmach webových aplikací jde ruku v ruce s rozšířením internetového připojení. Spousta lidí dnes vlastní tzv. chytrý telefon, u něhož je v podstatě nutností připojení k Internetu, aby se dal využít celý jeho potenciál. Lidé nosí svůj mobilní telefon stále u sebe, a proto si mohou kdykoliv prohlédnout své oblíbené webové stránky. Proto si myslím, že implementovat program jako webovou aplikaci je sázka na jistotu.

Další možností jsou mobilní aplikace, jejichž výhodou je možnost stáhnout si je přímo do mobilního zařízení. I tyto aplikace zažívají v poslední době nebývalý rozmach. Nevýhodou je, že se programátor musí rozhodnout, pro jakou mobilní platformu ji vytvoří. Je libo Android či iOS? Nebo snad obojí? Stálicemi jsou desktopové aplikace pro stolní počítače a notebooky. Ty však už z podstaty věci nemohou konkurovat na poli aplikací pro mobilní zařízení.

Desktopovou aplikaci je možné spustit na stolním počítači a notebooku. Mobilní aplikaci lze stáhnout pouze do mobilu nebo tabletu. Ale k webové aplikaci lze přistupovat ze všech těchto zařízení. Díky svým předchozím zkušenostem s vývojem webových aplikací a kvůli jejich výhodám jsem se rozhodla jít u svého bakalářského projektu právě touto cestou.

2.1 Webové technologie

Přibývá stále více typů zařízení, proto je nutné stylovat HTML dokument tak, aby byla zobrazená stránka optimalizována pro všechny druhy nejrůznějších zařízení (mobily, notebooky, netbooky, tablety atd.). V této souvislosti se mluví o tzv. responzivním web designu. Jedná se o přístup, kdy by měl design reagovat na uživatelské chování a rozměry obrazovky. Prakticky se jedná o kombinaci flexibilních mřížek a rozvržení, obrázků a použití CSS dotazů na medium (media queries). Ať už se uživatel dívá na stránku v jakémkoli zařízení, měla by se přizpůsobit rozlišení zařízení. To omezuje potřebu mít samostatnou verzi stránek pro mobilní zařízení a desktopové počítače [6]. V současnosti je podporována technika *mobile first*, kdy se stránky nejprve přizpůsobí pro mobilní telefony a pak se postupuje dále směrem k větším zařízením.

Tuto techniku podporuje Twitter Bootstrap¹, což je soubor nástrojů pro vytváření webu. Podporuje dvanáctisloupcové rozvržení a fluidní mřížku [10]. Nabízí mnoho prvků, které je možné snadno implementovat do své stránky. Interaktivní prvky, jako jsou tlačítka, boxy, menu a další kompletně nastavené a graficky zpracované elementy, jsou hlavní předností tohoto frameworku. To se někomu může zdát jako nevýhoda, ale je na každém návrháři,

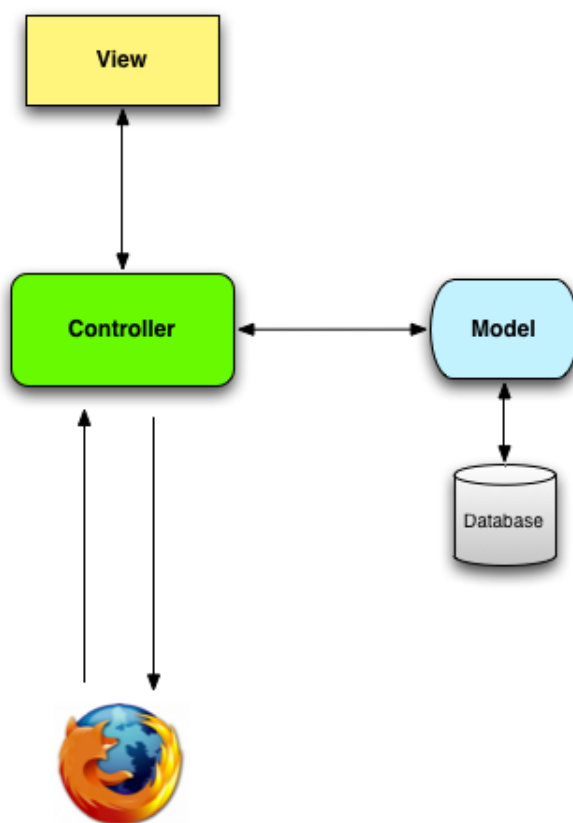
¹<http://getbootstrap.com/>

které části použije do své aplikace. Další podobné responzivní frameworky jsou např. ZURB Foundation² nebo Skeleton³.

UX design (User eXperience Design) vysvětluje všechny aspekty uživatelské zkušenosti se systémem, mj. i to, zda produkt dělá to, co se od něj očekává, tím způsobem, který se očekává [7]. Tato zkušenost zahrnuje i rozhraní, grafiku, design a fyzickou interakci. Už od začátku a v každé fázi procesu návrhu designu se počítá s koncovým uživatelem. UX design je často spojován s použitelností.

2.2 Architektura Model-View-Controller

Rozdělení aplikace na tři základní části podporuje architektura Model-View-Controller (viz obrázek 2.1). Model reprezentuje objekt v aplikaci (např. uživatele) a komunikuje s databází. View (pohled) je zodpovědný za generování uživatelského rozhraní. Zobrazuje data, která mu zprostředkuje model [11]. Požadavek, který je zaslán webovým prohlížečem, je přijat serverem a postoupen controlleru (řadiči), jenž reprezentuje řídicí logiku. Ten reaguje na události, typicky uživatelský vstup, spolupracuje s modelem a rozhoduje, který pohled se má použít [1]. Na této architektuře je založeno i Ruby on Rails.



Obrázek 2.1: Schéma architektury Model-View-Controller (převzato z [1])

²<http://foundation.zurb.com/>

³<http://getskeleton.com/>

2.3 Ruby on Rails

Rails⁴ je webový framework napsaný v jazyce Ruby⁵. Je navržen tak, aby bylo programování webových aplikací co nejjednodušší. Toho dosahuje tím, že spousta věcí už je předpřipravena a nastavena, a proto vývojáři nemusí psát tolik vlastního kódu. Tato filozofie je známa jako *convention over configuration*, tedy konvence má přednost před konfigurací. Druhou filozofií je tzv. DRY (Don't Repeat Yourself), která podporuje myšlenku nepsat jeden a ten samý kus kódu pořád dokola. Tím se stává kód lehce udržitelným s méně chybami [2].

Model-View-Controller

Architektura Model-View-Controller je v Rails realizována tak, že model je reprezentován pomocí vrstvy Active Record, která dodržuje principy ORM (objektově relační mapování), tj. tabulky jsou mapovány na třídy, řádky na objekty a sloupce na atributy objektů [11].

Action View generuje výstup jako HTML stránku, která je definovaná pomocí šablony, což je HTML kód, do kterého jsou vkládány ruby příkazy a data pomocí šablonovacího systému Embedded Ruby. Následující kód vypíše třikrát za sebou uživatelovo jméno.

```
<% 3.times do %>
  <%= @user.name %>
<% end %>
```

Action Controller přebírá uživatelský vstup, interaguje s modelem a předává pohledu informace nutné k zobrazení stránky. Rails podporuje architekturu REST (Representational State Transfer) [3]. Pomocí HTTP požadavků (např. GET nebo POST) a čtyř základních CRUD operací (Create, Read, Update, Delete) implementuje sedm metod (viz tabulka 2.1).

REST požadavek	Rails akce	Popis akce
GET /users	index	Zobrazí všechny uživatele
GET /users/new	new	Zobrazí formulář pro vytvoření uživatele
POST /users	create	Vytvoří uživatele
GET /users/1	show	Zobrazí konkrétního uživatele
GET /users/1/edit	edit	Zobrazí formulář pro úpravu uživatele
PUT /users/1	update	Upraví uživatele
DELETE /users/1	destroy	Smaže uživatele

Tabulka 2.1: Reprezentace REST architektury

Migrace, validace dat

K úpravě databáze se využívá tzv. migrací, které ji zachovávají v konzistentním stavu [4]. Pomocí nich můžeme přidávat či mazat tabulky a sloupce, nebo vkládat data. To vše aniž by musel programátor napsat ručně jediný SQL příkaz. Díky tomu jsou migrace nezávislé na typu databáze. Každá z nich je uložena jako samostatný soubor. Příklad migrace, která

⁴<http://rubyonrails.org/>

⁵<https://www.ruby-lang.org/>

vytvoří tabulku `Products` a v ní dva sloupce `Name` a `Description`, může vypadat takto:

```
class CreateProducts < ActiveRecord::Migration
  def change
    create_table :products do |t|
      t.string :name
      t.text :description
    end
  end
end
```

Ověřování platnosti dat může být uskutečněno na straně klienta pomocí JavaScriptu nebo lze přímo využít základních funkcí Rails frameworku. Validace dat probíhá na úrovni modelu, a to těsně předtím, než jsou data uložena do databáze. Validace lze využít i pouze pro některé akce, např. vytvoření nového uživatele, nebo mohou být použity i bez potřeby ukládat objekt do databáze. Jako příklad je uveden následující kód:

```
class User < ActiveRecord::Base
  validates :name, presence: true # jméno musí být zadáno
  validates :password, length: { in: 6..20 } # heslo musí obsahovat 6 až 20
  znaků
end
```

2.4 Uživatelské testování

Důležitou metodou při testování použitelnosti je uživatelské testování. Tato metoda pozoruje chování uživatelů. Může vývojářům poskytnout informace, pomoci odhalit chyby či eliminovat jiné problémy. Testování může probíhat buď v průběhu vývoje, nebo v době, kdy je aplikace plně funkční [5].

Jedním z možných způsobů testování, které se provádí na začátku vývoje, může být pozorování uživatelů při práci. Následně je s nimi proveden rozhovor. Zjistí se tím, co vlastně lidé dělají, jaké činnosti provádějí nejčastěji a jak jim může vyvíjený produkt pomoci v práci. Tato technika je cílena na konkrétní skupinu potenciálních uživatelů aplikace [9].

A/B testování je vhodné v případech, kdy je na výběr ze dvou či více grafických návrhů a je potřeba zjistit, který z nich je nejlepší. Vytvoří se proto různé verze produktu a každou z nich používá jiná skupina uživatelů. Pomocí různých nástrojů lze měřit různé parametry, např. čas strávený na stránce nebo kolik uživatelů kliklo na odkaz [9].

Další možností testování jsou skupinové diskuze. V tomto případě je v průběhu vývoje pozvána skupina lidí, kteří mají za úkol pracovat s produktem a plnit zadané úkoly. Přitom nahlas vyjadřují své myšlenky. Nakonec se uskuteční diskuze mezi všemi účastníky. Touto technikou lze odhalit některé chyby uživatelského rozhraní [8].

Kapitola 3

System pro správu diplomových prací

Diplomová práce je obvykle pro studenty posledním projektem, který musí v rámci svého studia vytvořit. Kromě státních závěrečných zkoušek je to nutnost k úspěšnému zakončení studia. Není proto divu, že se studenti snaží, aby byli při vytváření práce úspěšní. Úspěch ale nezávisí pouze na vědomostech studenta. Zde je třeba vzít v úvahu i čas strávený nad tvorbou práce, tj. zda jí student věnoval dostatečné množství času a jestli si dokázal práci dobře časově zorganizovat. Proto by se jim mohl hodit nástroj, díky němuž si naplánují hlavní body, které je zapotřebí splnit, v rámci delšího časového úseku, např. školního roku, nebo si budou psát reporty o tom, co už zvládli udělat a co by mohli mít za úkol následující týden.

Na druhé straně stojí vedoucí diplomové práce. Ten chce mít přehled o aktivitě svých studentů. Chce si prohlédnout, jak na tom studenti jsou, příp. reagovat na jejich reporty. Psát hromadné e-maily všem studentům. Toto jsou dvě strany pohledu na problematiku řešení diplomových prací.

3.1 Existující wiki řešení

Základem pro tuto bakalářskou práci byla wiki stránka pana Ing. Vítězslava Berana, Ph.D. který chtěl mít přehled o aktivitě svých studentů. Proto si vytvořil stránku s tabulkou, která by reprezentovala každotýdenní postup studentů vyjádřený pomocí smajlíků. Čísla označují pokrok v rámci daného měsíce (viz obrázek 3.2). Druhou hlavní částí je stránka s reporty. Student si každý týden vyplní, co zvládl udělat za daný týden, a naplánuje si práci na týden následující. Detail takového reportu je vidět na obrázku 3.1.

2014-10-05
<ul style="list-style-type: none">• Co se v tomto týdnu podařilo udělat?<ol style="list-style-type: none">1. začala jsem číst knížku o webdesignu2. byla jsem na konzultaci, kde jsme si dohodli základní funkcionalitu3. doplnila jsem si plán práce na celý školní rok• Co udělám v následujícím týdnu?<ol style="list-style-type: none">1. budu číst tutoriály nebo nějakou knížku2. studium existujících řešení• Jak jsem se za minulý týden posunula k plánovanému výsledku na minulý měsíc?<ul style="list-style-type: none">• z [1/10] na [3/10]

Obrázek 3.1: Detail reportu

Název a autor	22.-28.9.	29.9.-5.10.	6.-12.10.	13.-19.10.	20.-26.10.	27.10.-2.11.	3.-9.11.	10.-16.11.	17.-23.11.	24.-30.11.	1.-7.12.	8.-14.12.	15.-21.12.
WEB a GUI													
Martina Stodolová - Webový portál pro BP/DP projekty	[4/10]	[10/10]	[1/10]	[3/10]	[5/10]	[5/10]	[1/10]	[3/10]	[3/10]	[3/10]	[1/10]	[4/10]	[6/10]
Balcar Juraj - Webový portál pro správu konzultačních schůzek	[5/10]	[10/10]	[0/10]	[1/10]	[3/10]	[10/10]	[3/10]	[3/10]	[6/10]	[6/10]	[2/10]	[3/10]	[4/10]
Štěpán Černoch - Webový portál pro rodinná multimediální data	[5/10]	[6/10]	[1/10]	[4/10]	[6/10]	[9/10]	[2/10]	[2/10]	[4/10]	[5/10]	[2/10]	[3/10]	[4/10]
David Špilka - Webový portál pro privátní půjčovnu rodinného vybavení	[6/10]	[10/10]	[2/10]	[3/10]	[6/10]	[9/10]	[1/10]	[2/10]	[4/10]	[10/10]	[2/10]	[3/10]	[4/10]

Obrázek 3.2: Tabulka na hlavní straně

Pokud mají studenti navštěvovat takovou stránku každý týden, mělo by být její ovládání příjemné a nemělo by člověka příliš zdržovat od důležitých věcí. Na obrázku 3.3 je možné vidět, že editace takové wiki stránky není zrovna jednoduchá a snadno pochopitelná. Proto si myslím, že autor wiki stránky i studenti ocení, pokud se původní řešení trochu inovuje, pozmění a přidají se nové možnosti.

Editing Beran-BP DP Projekty 2014/2015 (section)

```

'''[[Martina Stodolová - Webový portál pro BP/DP projekty]]'''
|| http://www.fit.vutbr.cz/~beranv/images/btn_ok.gif <br /> [4/10]
|| style="border-right: 3px solid black; background: #c0f56e;" | http://www.fit.vutbr.cz/~beranv/images/btn_good.gif <br /> [10/10]
|| http://www.fit.vutbr.cz/~beranv/images/btn_holiday.gif <br /> [1/10]
|| http://www.fit.vutbr.cz/~beranv/images/btn_ok.gif <br /> [3/10]
|| http://www.fit.vutbr.cz/~beranv/images/btn_ok.gif <br /> [5/10]
|| style="border-right: 3px solid black;" | http://www.fit.vutbr.cz/~beranv/images/btn_bad.gif <br /> [5/10]
|| http://www.fit.vutbr.cz/~beranv/images/btn_bad.gif <br /> [1/10]
|| http://www.fit.vutbr.cz/~beranv/images/btn_ok.gif <br /> [3/10]
|| http://www.fit.vutbr.cz/~beranv/images/btn_holiday.gif <br /> [3/10]
|| style="border-right: 3px solid black;" | http://www.fit.vutbr.cz/~beranv/images/btn_holiday.gif <br /> [3/10]
|| http://www.fit.vutbr.cz/~beranv/images/btn_bad.gif <br /> [1/10]
|| http://www.fit.vutbr.cz/~beranv/images/btn_ok.gif <br /> [4/10]
|| http://www.fit.vutbr.cz/~beranv/images/btn_ok.gif <br /> [6/10]

```

Obrázek 3.3: Editace tabulky na hlavní straně

3.2 Specifikace potřeb studentů a vedoucích

System rozlišuje tři typy uživatelů. *Student* je řešitelem projektu a může upravovat pouze údaje týkající se jeho osoby nebo projektu. *Vedoucí* může spravovat projektové skupiny a v rámci nich i studenty, jejich projekty, dále pak termíny a jiné. *Administrátor* má dohled nad celým systémem. Jeho hlavní funkcí je správa vedoucích. Na následujících řádcích bude uvedeno využití aplikace z pohledu studenta a vedoucího.

Student

Někteří studenti ocení, když si mohou **naplánovat práci** na týden dopředu. Ujasní si myšlenky a připraví si plán na nadcházející týden. A nejlépe když si to budou moci všechno

sepsat na jedno místo. Na konci každého týdne by si student sepsal **report**, co se mu za uplynulý týden podařilo zvládnout. Něco jako pracovní výkaz. Čím více věcí, tím lépe. Potom by měl možnost porovnat to, co měl v úmyslu udělat, s tím, co udělal doopravdy. To by ho mohlo upozornit na to, že nesplnil vše, co si naplánoval, nebo motivovat jako týdenní posun.

Nástroj pro organizaci práce na diplomových projektech by měl také podporovat vzájemnou **spolupráci studentů**. Studenta dokáže motivovat pokrok ostatních kolegů. Když uvidí, že se jim daří, může ho to přimět k tomu, aby se více věnoval své práci. Za vzájemnou spolupráci lze považovat i **sdílení informací**. Často může student při studování různých zdrojů narazit na nějaký materiál (např. knihu, webovou stránku nebo dokument), který by si rád někde uložil či poznamenal pro pozdější využití. Systém by mu měl dovolit si všechny tyto materiály shromažďovat přehledně na jednom místě nebo sdílet dokumenty s ostatními studenty.

Vyskytnou se situace, kdy student potřebuje poslat e-mail spolužákům (vedeným pod stejným vedoucím), např. aby je požádal o radu, sdílel nějaký nápad nebo aby mu pomohl s testováním jeho programu. V tu chvíli se hodí možnost poslat **hromadný e-mail**, aniž by se musel zdlouhavě zaobírat hledáním e-mailových adres svých kolegů.

Vedoucí

Vedoucí chce mít přehled o svých studentech. Rád by viděl jejich pokrok a udělal si obrázek o jejich aktivitě. Je vhodné rozdělit studenty do **projektových skupin** podle tématu diplomové práce či akademického roku. Dále by vedoucí jistě ocenili možnost třdit jednotlivé práce pomocí **štítků**. Příkladem takového štítku může být „GUI“ pro práce zabývající se grafickým uživatelským rozhraním nebo „robo“ u prací s tématem robotiky.

Důvodem pro používání takovéto aplikace by pro vedoucího měla být možnost snadné vizualizace aktivity studentů na projektu. Tento **přehled** by měl být hlavní částí systému referující o tom, jak se studentům dařilo v jednotlivých týdnech, resp. měsících.

Dalším cílem systému by mělo být zprostředkování **shrnutí reportů**. Vedoucí chce vidět, co měl student na daný týden naplánováno a co skutečně udělal, a také na to chce nějak reagovat. Což by bylo možné díky zadávání poznámek či **zpětné vazby** k reportům.

Během semestru či akademického roku se mohou vyskytnout události, na které chce vedoucí své studenty upozornit. Může se jednat o prezentaci semestrálního projektu nebo odevzdání diplomové práce. Studenty je třeba o těchto **termínech** informovat, a to např. tím, že se jim zobrazí na hlavní straně.

V neposlední řadě by měl systém umožňovat propojení s fakultním informačním systémem. Ne vždy je pohodlné přidávat studenty ručně po jednom. Proto by měla aplikace zahrnovat nějaký jednoduchý způsob, jak **importovat** data získaná z **informačního systému**.

Výše uvedená specifikace je vyjádřena pomocí diagramu případů užití, který je uveden v příloze A.

3.3 Reprezentace dat v datovém modelu

Uživateli aplikace jsou studenti a vedoucí a potřebujeme o nich uchovávat jejich *jméno* a *příjmení*, dále pak *e-mail* a *heslo*.

Systém se zaměřuje na diplomové práce, je tedy nutné o nich ukládat informace. Zde je využita entita *projekt*. Projekt je řešen *studentem* a veden *vedoucím*. O projektu potřebu-

jeme znát jeho *název*, *zadání* a *typ*, tj. jestli se jedná o bakalářskou nebo diplomovou práci. Každý student může být řešitelem jednoho či více projektů.

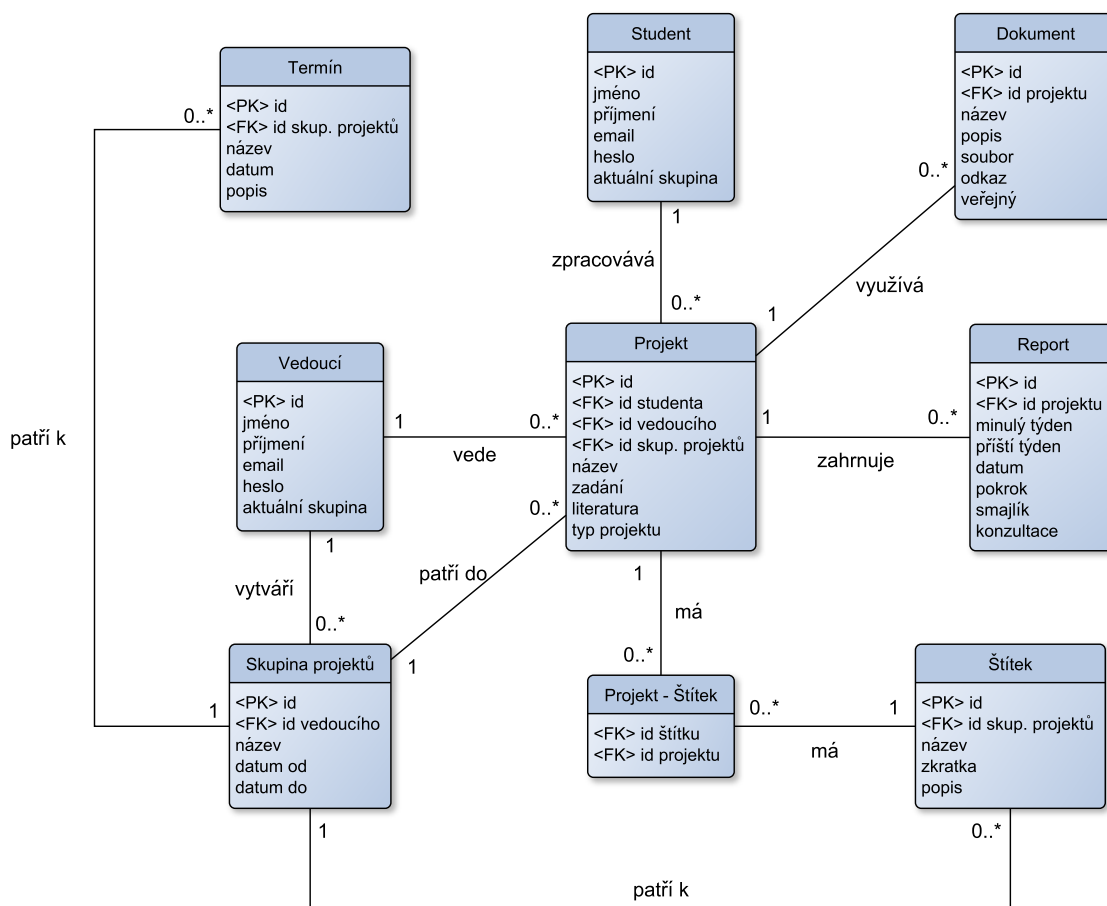
Entita *projekt* je vazbou M:N propojena s entitou *tag* (štítkem), jelikož projekt může být označen více štítky a zároveň jeden štítek může být použit u více projektů. O štítku se uchovávají informace o jeho *názvu*, *zkratce* a *popisu*.

Důležitým bodem aplikace jsou *skupiny projektů*. Každý projekt patří právě do jedné skupiny. Ke každé skupině může být připojen *štítek* nebo *termín*, který má atributy *název*, *datum* a *popis*.

Při psaní *reportů* si chce student zaznamenat shrnutí za *minulý týden* a plán na *další týden*. Dále pak zadá *pokrok* a *smajlík*. Report je vázán na konkrétní *projekt* a určitý *týden*.

Studenti si přidávají *dokumenty*, které se vážou k projektu. Důležitý je jejich *název* a *popis*, dále pak *odkaz* na ně, pokud se jedná o webové stránky, příp. celý dokument. Nechybí ani atribut určující, zda je zdroj *veřejný*.

Tato data se dají shrnout do ER diagramu (entity relationship diagram), který slouží pro modelaci dat a vyjádření vztahu mezi nimi (viz obrázek 3.4).



Obrázek 3.4: Reprezentace dat a vztahů mezi nimi pomocí ER diagramu

3.4 Návrh uživatelského rozhraní

Hlavní strana vychází z původního řešení (viz obrázek 3.5). Snažila jsem se zachovat její jednoduchost a přehlednost. Zachovala jsem původní smajlíky, ale vyjádření pokroku v rámci měsíce jsem změnila z [2/10] na 20 %, což je dle mého názoru více vypovídající. Zkoušela jsem prohodit osu se studenty a osu s týdny, ale nakonec se ukázalo, že je výhodnější mít týdny jako sloupce, aby se mezi nimi dalo pohodlně posouvat.

	4.5. - 10.5.	11.5. - 17.5.	18.5. - 24.5.	25.5. - 31.5.
Jméno Příjmení Název BP/DP štitky	○ 80 %	○ 80 %	○ 80 %	
Jméno Příjmení Název BP/DP štitky		○ 80 %		
Jméno Příjmení Název BP/DP štitky	○ 80 %			○ 80 %
Jméno Příjmení Název BP/DP štitky		○ 80 %		
Jméno Příjmení Název BP/DP štitky				

Obrázek 3.5: Hlavní strana

Původně jsem zamýšlela mít hlavní stranu přístupnou i pro nepřihlášené uživatele. Nakonec jsem od tohoto záměru musela upustit, jelikož je aplikace zamýšlena pro více vedoucích, a nebylo by tedy jasné, kteří studenti se mají na stránce zobrazit. Vše se totiž zobrazuje v rámci projektové skupiny. Je důležité zmínit, že student může být zařazen do více projektových skupin zároveň, a to i u jiného vedoucího. Student i vedoucí se může mezi těmito skupinami přepínat.

Prioritou pro studenty je rychlé zadávání reportu, proto jsem se tuto činnost snažila co nejvíce zjednodušit (viz obrázek 3.6). Přidávání týdenního výkazu je koncipováno jako jeden formulář. Student do dvou textových polí shrne odvedenou práci a plán na další týden. Zadávání pokroku za daný týden jsem mohla vyřešit jako rozbalovací seznam. Rozhodla jsem se však pro čtyři obrázky, z nichž se dá jeden vybrat kliknutím. K udávání měsíčního pokroku by se mohlo využít textového pole pro zápis čísla, ale myslím si, že posuvník od 0 % do 100 % lépe vystihuje daný posun.

Přehled reportů zobrazuje tabulka (viz obrázek 3.7), kde každý řádek odpovídá jednomu studentovi. Sloupce vychází z toho, co student zadal do svých reportů, tedy co měl udělat, co skutečně udělal a co udělá příští týden. Poslední sloupec je vyhrazen jako textové pole pro vedoucího, kam může zadat zpětnou vazbu k reportům, různé poznámky a podobně. Toto odstraňuje nutnost posílání reportů vedoucímu prostřednictvím e-mailu. Vedoucí má vše přehledně zobrazeno v systému svoji zpětnou vazbu může kdykoliv uložit do systému či poslat jako zprávu studentům.

Logo
 Martina S.

Přidat report

Co jsem udělal tento týden

Co udělám příští týden

Pokrok za tento týden

Pokrok v rámci měsíce

Týden

Uložit

Obrázek 3.6: Návrh formuláře pro přidání reportu

Logo
 Martina S.

Přehled reportů

	Měl jsem udělat 4.5. - 10.5.	Udělal jsem 4.5. - 10.5.	Udělám 11.5. - 17.5.	Zpětná vazba vedoucího
Jméno Příjmení	1) text 2) text 3) text 4) text	1) text 2) text 3) text 4) text	1) text 2) text 3) text 4) text	1) text 2) text 3) text 4) text
Jméno Příjmení	1) text 2) text 3) text 4) text	1) text 2) text 3) text 4) text	1) text 2) text 3) text 4) text	1) text 2) text 3) text 4) text
Jméno Příjmení	1) text 2) text 3) text 4) text	1) text 2) text 3) text 4) text	1) text 2) text 3) text 4) text	1) text 2) text 3) text 4) text

Odeslat zpětnou vazbu

Obrázek 3.7: Tabulka přehledu reportů

Pokud vedoucí požaduje po studentech odevzdání určitých dokumentů, může ke každé skupině přidat položky reprezentující povinné dokumenty. Může se jednat např. o technickou zprávu nebo prezentaci k obhajobě. Studenti je ke svým projektům připojí buď jako odkaz, nebo ho přímo nahrají do systému. Vedoucí tak má tyto dokumenty přehledně na jednom místě, kdykoliv si je chce prohlédnout.

Studenti mohou do systému ukládat webové odkazy, ale i přímo nahrávat různé dokumenty. Pokud se chce student o některý dokument podělit i s ostatními kolegy, může ho označit jako veřejný, což ho zpřístupní ostatním. Pokud se jinému studentovi líbí nějaký dokument, může ho sdílet, čímž se mu zobrazí i v jeho materiálech.

Jestliže by vedoucí musel zadávat v každé skupině podobné nebo úplně stejné štítky, termíny či povinné dokumenty, zbytečně by ho to zdržovalo. Proto se mu bude hodit možnost zkopírovat dané položky z jiné projektové skupiny (viz obrázek 3.8). Potom je jen upraví nebo některé smaže či přidá.

The screenshot shows a web interface for copying items. At the top left is a 'Logo' placeholder. At the top right is a user profile for 'Martina S.'. The main content area is titled 'Kopírovat' and contains three unchecked checkboxes: 'termíny', 'povinné dokumenty', and 'štítky'. Below these is a dropdown menu labeled 'ze skupiny' with a checkmark icon. At the bottom is a 'Kopírovat' button.

Obrázek 3.8: Kopírování položek z jiné skupiny

Nastane-li situace, kdy potřebuje vedoucí poslat všem studentům stejný e-mail, např. aby všechny seznal na společnou konzultaci, může tak učinit stejně lehce jako studenti, kdy využije možnost aplikace poslat hromadné e-maily. Aby bylo možné odhalit případné spamování uživatelů, je zde možnost zobrazit si přehledný seznam všech e-mailů odeslaných systémem. Toto slouží zároveň i jako přehled pro vedoucího, aby věděl, kdo a jaké zprávy posílá.

Studenti by se mohli do systému registrovat sami nebo by je tam mohl ručně přidávat vedoucí. To by ale nebylo zrovna pohodlné. Proto systém podporuje import dat získaných z fakultního informačního systému. Vedoucí si vyexportuje data o studentech a jejich diplomových pracích. Tato data lze pak nahrát do aplikace. V synchronizačním procesu pak mohou být k pracím přidávány štítky.

V původním návrhu importu jsem počítala s tím, že se v každém kroku přidá jeden student a umožní se editace všech jeho údajů i informací o projektu (viz obrázek 3.9). Po delším zvážení a po poradě se školitelem jsem se však rozhodla tento návrh nezrealizovat, protože by takové přidávání studentů nebylo pro vedoucího vůbec efektivní. Proto jsem dospěla k názoru, že bude lepší importovat všechny studenty v jednom kroku s minimální

možností úpravy dat (viz obrázek 4.2). Vedoucí si pouze zvolí, které studenty chce přidat do aplikace, případně k jejich projektům ještě přiřadí štítky, a pomocí tlačítka spustí import.

The screenshot shows a web interface for importing a student. At the top left is a placeholder for a logo, and at the top right is a user profile icon labeled 'Martina S.'. The main heading is 'Import studenta #3 z 20'. The form contains the following fields:

- Jméno** (Name)
- Příjmení** (Surname)
- E-mail**
- Projekt** (Project) section with three sub-fields:
 - Název** (Name)
 - Zadání** (Assignment)
 - Literatura** (Literature)

At the bottom of the form is a button labeled 'Uložit studenta a přejít na dalšího' (Save student and go to the next one).

Obrázek 3.9: Původní návrh importu studentů z informačního systému

Kapitola 4

Aplikace Student & Supervisor

Tato kapitola popisuje realizaci výsledné aplikace Student & Supervisor. Je zde popsána základní struktura Rails aplikace, dále jsou zde uvedeny některé použité balíčky a zajímavosti z implementace. Poslední dvě části jsou věnovány provedenímu testování a jeho výsledkům.

4.1 Struktura Ruby on Rails aplikace

Výsledná Ruby on Rails aplikace má tuto adresářovou strukturu:

```
app/
  |-- assets/
  |-- controllers/
  |-- helpers/
  |-- mailers/
  |-- models/
  +-- views/
bin/
config/
db/
lib/
log/
public/
vendor/
```

Složka `app` obsahuje jádro celé aplikace. Architektura Model-View-Controller se do adresářové struktury promítla třemi složkami nazvanými `controllers`, `models` a `views`. Jejich obsah budu demonstrovat na entitě `student`. Soubor `student.rb` reprezentuje model, který je definovaný třídou `Student`. V tomto souboru jsou uvedeny asociace s jinými modely, jejichž ukázkou může být:

```
has_many :projects, dependent: :destroy
```

Tento kus kódu vyjadřuje vztah studenta a projektu, kdy jeden student může mít přiřazeno víc studentů. Pokud bude student smazán, budou smazány i k němu přiřazené projekty. Soubor `students_controller.rb` zahrnuje definici sedmi základních metod zmíněných v ka-

pitole 2.3 a navíc metodu `change_password` pro změnu hesla. Většina metod uvedených v controlleru má svůj vlastní pohled jako samostatný soubor ve složce `views`. Příkladem může být soubor `index.html.erb`, který slouží k zobrazení všech studentů.

Soubor s mými kaskádovými styly a skripty JavaScriptu jsem uložila do složky `assets`. Rovněž jsou zde umístěny obrázky použité v aplikaci, např. obrázky smajlíků. Dále je zde adresář s tzv. `mailery`, které definují formát e-mailů zasílaných aplikací. Složka `helpers` pak obsahuje soubory s pomocnými metodami, které lze volat z jakékoliv hlavní části aplikace.

Do složky `config` jsem kromě jiného umístila i soubor `database.yml`, jehož ukázka je uvedena níže. Do adresáře `db` patří všechny soubory s migracemi, jejichž využití bylo vysvětleno v kapitole 2.3. Je zde též soubor `seeds.rb`, jehož prostřednictvím lze vkládat prvotní data do databáze již při její inicializaci. Toho jsem využila, abych vytvořila administrátora celé aplikace.

Při nahrávání souborů do aplikace knihovna *Paperclip* (viz následující sekce) ukládá tyto soubory do složky `public`. Na toto místo jsem uložila i soubory s fonty použitými aplikací. Poslední je složka `vendor`, která slouží pro uchování souborů knihoven třetích stran, jako jsou např. pluginy JavaScriptu nebo kaskádové styly frameworku Bootstrap.

RubyGems

RubyGems je software pro správu balíčků, kterým se říká gemy. Tyto balíčky se dají snadno použít v Ruby aplikacích. Gemy mohou být použity k rozšíření nebo pozměnění Ruby on Rails aplikace. Využívají se jako prostředek pro znovupoužití často se vyskytujícího kódu. V Rails aplikacích se použité gemy uvádějí v souboru `Gemfile`. Zde uvedu některé gemy, které jsem použila ve své aplikaci.

Ke zprovoznění autorizace uživatelů jsem použila knihovnu *CanCanCan*¹, která vymezuje, ke kterým částem aplikace smí konkrétní uživatel přistupovat. Práva uživatelů jsou definována v jednom souboru s třídou `Ability`.

Pro vylepšení některých URL adres jsem využila balíček *FriendlyId*², který umí vytvořit URL čitelnou pro člověka. Programátor si může sám zvolit, jestli preferuje vytvořit URL třeba z jména a příjmení uživatele. Např. adresu

```
http://example.com/users/238
```

dokáže vylepšit na

```
http://example.com/users/martina-stodolova
```

Jako poslední uvedu gem *Paperclip*³, který usnadňuje nahrávání souborů do aplikace. Balíček poskytuje validaci nahrávaných souborů na základě jejich velikosti nebo typu. Využila jsem ho při připojování souborů k dokumentům.

Připojení k databázi

Datový model z kapitoly se stal základem relační databáze. Jelikož je aplikace umístěna na platformě *Heroku*⁴, rozhodla jsem se využít její *PostgreSQL* databázi k ukládání dat.

¹<https://github.com/CanCanCommunity/cancancan>

²https://github.com/norman/friendly_id

³<https://github.com/thoughtbot/paperclip>

⁴<https://www.heroku.com/>

Propojení Ruby on Rails aplikace s databází je snadné. Stačí pouze nakonfigurovat soubor `database.yml`. Ukázka takového souboru:

```
production:
  adapter: postgresql
  host: localhost
  database: rails_aplikace
  pool: 5
  timeout: 5000
  username: uzivatel
  password: heslo
  port: 5432
```

4.2 Řešení implementačních problémů

Do této části jsem vybrala tři problémy, se kterými jsem se potýkala při implementaci aplikace, a popsala svůj způsob jejich řešení.

Import dat z informačního systému

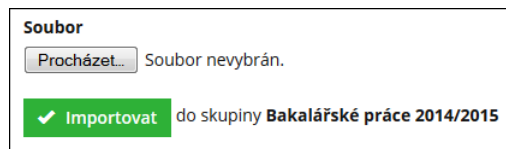
Kromě manuálního přidávání studentů po jednom je v aplikaci možnost importovat data z fakultního informačního systému. Po zadání URL adresy

```
https://wis.fit.vutbr.cz/FIT/db/vyuka/szz/get-bciprj.php?login=LOGIN&year=YEAR
```

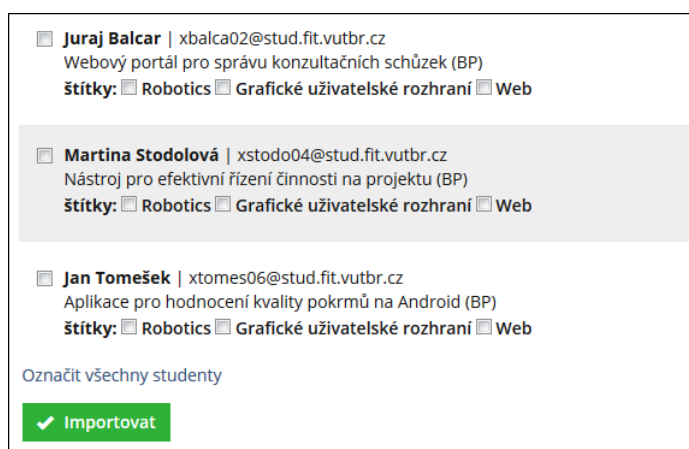
kde LOGIN zastupuje login vedoucího a YEAR znamená rok, si může vedoucí vyexportovat informace o studentech a jejich projektech do JSON souboru, který má následující formát:

```
{
  "data": [{
    "id": 17665,
    "year": 2014,
    "type": "BP",
    "student": "Stodolová Martina",
    "login": "xstodo04",
    "title": "Nástroj pro efektivní řízení činnosti na projektu",
    "text": "<ol>\r\n<li>Studujte postupy návrhu uživatelských rozhraní a techniky a nástroje pro tvorbu moderních uživatelských rozhraní pro web. \r\n</li></ol>",
    "literature": "<ul>\r\n<li>B. Shneiderman, C. Plaisant. <i>Designing the user interface: strategies for effective human-computer interaction</i>, Addison-Wesley Computing, ISBN-10: 0-321-53735-1, 2009 \r\n</li></ul>",
    "created_time": "2014-0921T18:20:17",
    "updated_time": "2014-1030T20:18:07",
  }]
}
```

Poté ho pomocí jednoduchého formuláře nahraje do aplikace (viz obrázek 4.1). V dalším kroku (zobrazeném na obrázku 4.2) si vybere, které studenty chce importovat, přiřadí k nim odpovídající štítky a jediným stiskem tlačítka přidá všechny vybrané studenty do systému.



Obrázek 4.1: Pole pro nahrání JSON souboru pro import

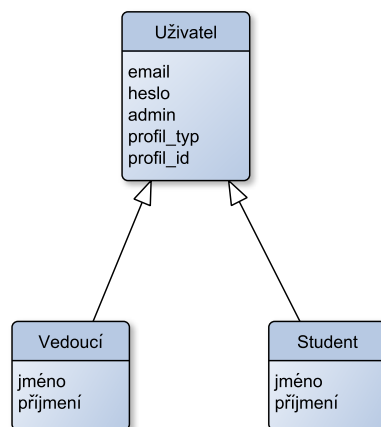


Obrázek 4.2: Vybírání uživatelů při importu

Role – vedoucí, student

Bylo nutné vyřešit, jak implementovat uživatelské role. Jedním z řešení by bylo přidat k modelu `User` atribut `role`, který by určoval, zda se jedná o vedoucího či studenta. To by však neumožňovalo ukládat si o vedoucím jiné informace než o studentovi a naopak.

Proto jsem se rozhodla použít polymorfickou vazbu, kdy jeden model může patřit více modelům, což je znázorněno na obrázku 4.3. Zde je model `User` propojen s modely `Supervisor` a `Student`. O tom, zda je uživatel vedoucí nebo student, rozhoduje atribut `profile_type`, který může nabývat hodnot "Supervisor" nebo "Student", a atribut `profile_id`, který reprezentuje cizí klíč do příslušné tabulky. Nyní lze ke studentovi, resp. k vedoucímu přidávat různé atributy dle potřeby.



Obrázek 4.3: Polymorfická vazba

Posouvání přehledu v rámci týdnů

Jedním z problémů u grafického rozhraní bylo vyřešit posouvání mezi týdny v tabulce na hlavní straně (viz obrázek 4.4). Řešení nabídla knihovna *Slick*⁵, která implementuje tzv. carousel, který umožňuje zobrazit více obrázků a přepínat mezi nimi posouváním do stran. Tuto myšlenku jsem aplikovala tak, že jsem jednu velkou tabulku rozdělila na spoustu menších, jednu pro každý týden. Všechny tyto tabulky jsem obalila jedním elementem a na ten jsem použila Slick. Díky němu se z těžce řešitelného problému stal jednoduchý a ve své aplikaci jsem ukázala, že takový nástroj nemusí být využíván pouze k prezentaci obrázků.



Obrázek 4.4: Ukázka hlavní strany aplikace

⁵<https://github.com/kenwheeler/slick/>

4.3 Grafické ztvárnění aplikace

Pro ilustraci grafického uživatelského rozhraní přikládám další dvě ukázky z aplikace.

Pokrok za týden

Pokrok v rámci měsíce 25 %

Týden 13. 4. 2015 - 19. 4. 2015

Co jsem udělal tento týden *

Co udělám příští týden *

Proběhla konzultace s vedoucím

uložit zpět

Obrázek 4.5: Formulář pro přidání reportu

	Měl jsem udělat 12. 1. - 18. 1.	Udělal jsem 12. 1. - 18. 1.	Udělám další týden 19. 1. - 25. 1.	Zpětná vazba vedoucího
Juraj Balcar <input checked="" type="checkbox"/> Webový portál pro správu konzultačních schůzek Pr Pi Vi Te				
Martina Stodolová <input checked="" type="checkbox"/> Nástroj pro efektivní řízení činnosti na projektu web, GUI Pr Pi Vi Te	<ul style="list-style-type: none">• spojit reporty a smajlíky do jednoho• zdokonalit reporty• vzít v úvahu návrhy, které padly na konzultaci• napsat několik stránek do TZ• pokračovat v implementaci	<ul style="list-style-type: none">• začít s psaním TZ• začít s možností posílání emailů• spojit reporty a smajlíky do jednoho	<ul style="list-style-type: none">• trochu designově vylepšit• napsat několik stránek do TZ• pokračovat v implementaci• přehodnotit časový plán• půjdu na konzultaci	Jen tak dál.
Jan Tomešek <input checked="" type="checkbox"/> Aplikace pro hodnocení kvality pokrmů na Android Pr Pi Vi Te				

Zaslat zpětnou vazbu studentům Uložit zpětnou vazbu

Obrázek 4.6: Vizualizace přehledu reportů po týdnech

4.4 Testování systému

V průběhu řešení této bakalářské práce a poté v samotném závěru jsem provedla uživatelské testování. V první fázi testování mě zajímalo, jak moc jsou uživatelé spokojeni se stávajícím řešením na wiki stránkách. K tomu bylo nutné vyplnit dotazník, jenž zahrnoval tyto otázky:

- Jak často navštěvujete wiki za účelem správy projektu?
- Kolik času strávíte na wiki při jedné návštěvě?
- Myslíte si, že má smysl spravovat projekt pomocí aplikace?
- Čím je pro Vás přínosná správa projektu?
- Co Vás naopak odrazuje od správy projektu?
- Co se Vám líbí na současné verzi systému?
- Co byste změnil(a) na současné verzi systému?
- Je pro Vás obtěžující psát pravidelné reporty o své činnosti?
- Co by Vás přimělo k tomu, navštěvovat systém častěji?
- Co by měl podle Vás takový systém obsahovat?
- Myslíte si, že z takového systému těží pouze vedoucí?

Výsledky prvního testování mají ukázat hlavně to, které části systému by měly být v nové aplikaci vyřešeny lépe. Také se staly inspirací pro různá vylepšení a nové prvky nového systému.

Potom následovala druhá fáze testování, kdy bylo potřeba zaregistrovat uživatele do systému a po pár týdnech používání aplikace jim předložit druhý dotazník. Ten se již týkal testované aplikace. Uživatelé zodpověděli tyto otázky:

- Co se Vám líbí na nové verzi systému?
- Co byste změnil(a) na nové verzi systému?
- Je nová verze přehlednější než původní?
- Je jednoduchá na ovládání?
- Jak na Vás graficky působí aplikace?
- Uvítal(a) byste více popisného či vysvětlujícího textu?
- Jak rychle jste se zorientoval(a) v aplikaci?

Tyto dva dotazníky byly vyhodnoceny. Výsledky testování ukázaly, jestli jsou uživatelé spokojeni více s novější verzí a jak se jim s ní pracuje.

4.5 Výsledky testování

Testování se zúčastnilo 11 studentů, kteří byli v dané době řešiteli diplomových prací. V této části jsou shrnuty odpovědi uvedené v dotaznících.

Z dotazníku týkajícího se původní verze systému lze zjistit, že studenti navštěvují wiki stránku za účelem správy projektu nejčastěji jednou týdně, dva lidé pouze jednou měsíčně. Při jedné návštěvě tam stráví průměrně 5-10 minut. Z toho lze soudit, že studentům vyhovuje si jedenkrát týdně vyplnit report a nestrávit u toho zbytečně moc času.

Většina respondentů se shodla na tom, že z takového systému netěží pouze vedoucí, a také si myslí, že je pro ně přínosné si spravovat projekt pomocí aplikace. Jako důvod byl uveden fakt, že mají přehled o své dosavadní práci a donutí je to dále pracovat. Dále pak je to možnost zpětně nahlédnout, jak práce postupovala v závislosti na čase, nebo přehled o tom, jak se práce vyvíjí v porovnání s ostatními kolegy. Na druhou stranu některé respondenty od správy projektu odrazuje to, že vypisováním reportů stráví příliš mnoho času, který by mohli věnovat samotné práci. Více než třetina studentů připadá psaní pravidelných reportů obtěžující.

Podle studentů by měl systém pro správu projektů zahrnovat jednoduché rozhraní, přehledné reporty, výstupy z konzultačních schůzek, viditelné milníky na aktuální měsíc. Někteří z nich by ocenili více interakce se svým vedoucím, jeho komentáře a hodnocení, protože by od něj rádi dostávali nějakou odezvu, která by jim pomohla s řešením projektu a zároveň je motivovala v další práci.

Původní verzi systému považují za jednoduchou a rychlou, ale jinak nevyzdvihli nic konkrétního. Naopak by uvítali lepší grafické uživatelské rozhraní. Nejsou spokojeni s editací wiki stránky, která jim přijde uživatelsky nepřívětivá.

Druhá fáze testování ukázala, že nová verze aplikace se zdá být pro uživatele přehlednější než ta původní. Přijde jim jednoduchá na ovládání a vizuálně na ně působí příjemně. Tři z jedenácti dotázaných by uvítali více popisného či vysvětlujícího textu a v systému se zorientovali během pár minut. Zbývající studenti odpověděli, že se zorientovali hned po přihlášení do aplikace.

Studenti ocenili možnost snadného přidávání reportů a nahrání dokumentů přímo do aplikace. Naopak jim chybí možnost přihlášení studenta na konzultace a možnost zobrazení reportu z těchto schůzek. Přestože se většině studentů zdá aplikace přehledná, pár studentů by uvítalo přehledné menu k navigaci v systému.

Některé z názorů a návrhů studentů byly brány v potaz již v průběhu implementace. Ostatní z nich budou pečlivě zváženy a ty přínosné budou zohledněny v další iteraci vývoje aplikace.

Bohužel se mi nepovedlo provést řádné testování aplikace s jinými vedoucími diplomových prací. Změny v návrhu vycházely z připomínek na pravidelných konzultacích s mým školitelem. Bylo by ale vhodné, aby byly do návrhu začleněny i poznatky jiných vedoucích.

Kapitola 5

Závěr

V závěru bych chtěla zhodnotit svůj nástroj pro efektivní řízení činnosti na projektu. Cílem bakalářské práce bylo navrhnout a implementovat systém, který vedoucím usnadní vedení studentských diplomových prací a zjednoduší komunikaci mezi nimi a studenty. Školitelé mají přehled o odvedené práci každého studenta a mohou jim poskytnout zpětnou vazbu. Aplikace také podporuje spolupráci studentů mezi sebou a může zvyšovat jejich motivaci při práci na projektech.

Návrh systému vycházel z již existujícího řešení, jehož myšlenku jsem se snažila zachovat, ale zároveň jsem se pokusila některé věci vylepšit nebo přidat zcela nové prvky systému. K implementaci jsem použila skriptovací jazyk Ruby a jeho framework Rails, protože už jsem s ním měla předchozí zkušenosti a také mi přišlo vhodné systém realizovat jako webovou aplikaci. Ta je umístěna na webu studentandsupervisor.herokuapp.com

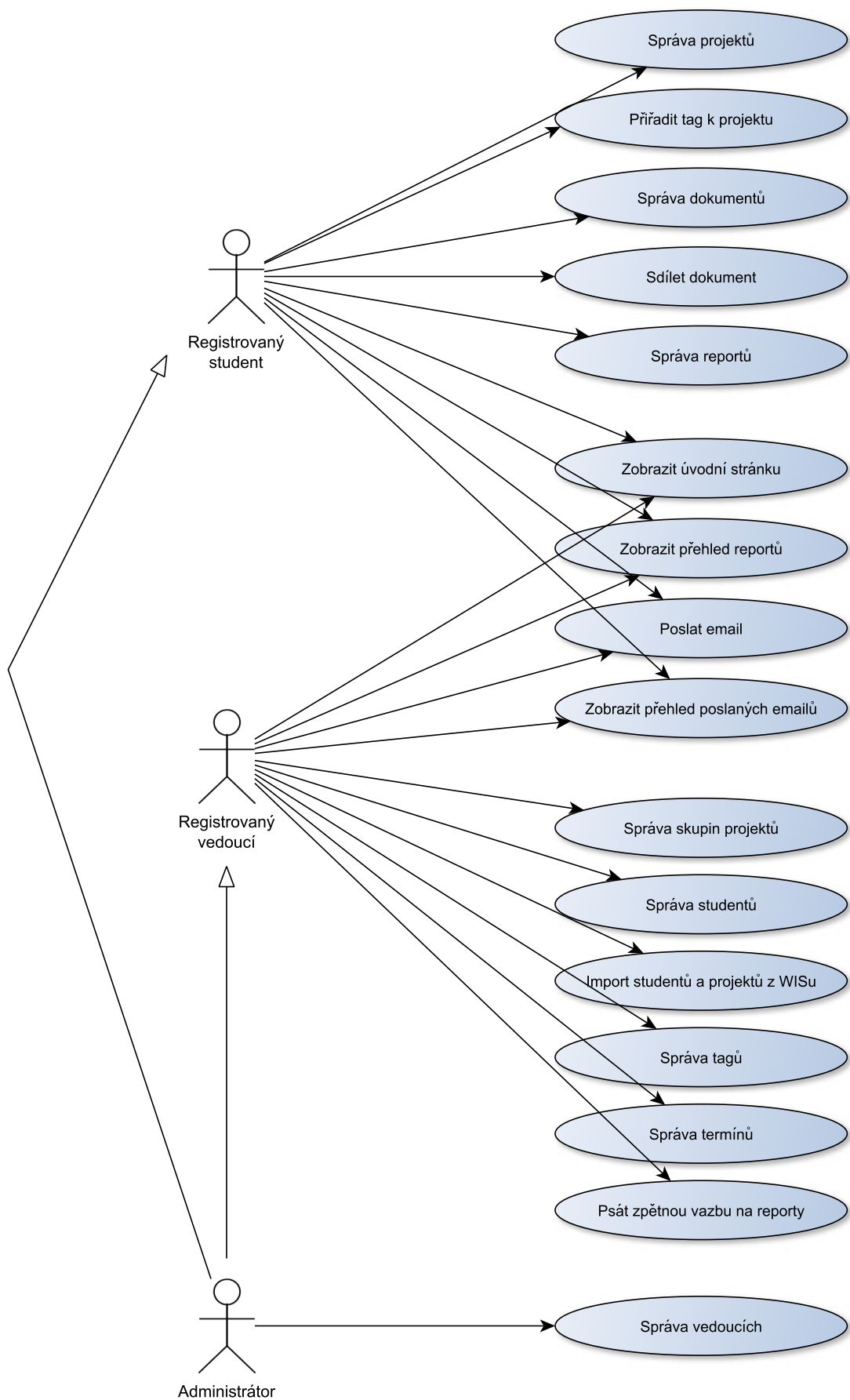
Při programování této aplikace pro mě bylo velmi motivující vědět, že se z ní nestane pouze bakalářský projekt, ale že ji budou využívat reální uživatelé. Na pravidelných konzultacích se školitelem jsme diskutovali o možných řešeních některých problémů, což přispělo k tomu, že se implementovaný systém začal vyvíjet tím správným směrem, což potvrdilo i testování se studenty. Do budoucna bych chtěla systém dále rozšiřovat a začleňovat návrhy na zlepšení i od dalších vedoucích a studentů. Doufám, že se rozšíří povědomí o této aplikaci a začne ji používat více vedoucích.

Literatura

- [1] *Ruby on Rails Tutorial*, [online]. [cit. 2015-05-17].
URL <https://www.railstutorial.org/book/>
- [2] *Getting Started with Rails*, [online]. [cit. 2015-05-17].
URL http://guides.rubyonrails.org/getting_started.html
- [3] *Building Your First Rails Application: Views and Controllers*, [online]. [cit. 2015-05-17].
URL <http://www.sitepoint.com/building-your-first-rails-application-views-and-controllers/>
- [4] *Active Record Migrations*, [online]. [cit. 2015-05-17].
URL http://guides.rubyonrails.org/active_record_migrations.html
- [5] *Uživatelské testování*, [online]. [cit. 2015-05-17].
URL
http://cs.wikipedia.org/wiki/U%C5%BEivatelsk%C3%A9_testov%C3%A1n%C3%AD
- [6] Frain, B.: *Responsive Web Design with HTML5 and CSS3*. Packt Publishing, 2012, ISBN 978-1-84969-318-9.
- [7] Garrett, J. J.: *The Elements of User Experience*. New Riders, 2011, ISBN 978-0-321-68368-7.
- [8] Krug, S.: *Don't Make Me Think!* New Riders, 2006, ISBN 0-321-34475-8.
- [9] Mathis, L.: *Designed for Use*. The Pragmatic Programmers, 2011, ISBN 978-1-93435-675-3.
- [10] Spurlock, J.: *Bootstrap*. O'Reilly Media, 2013, ISBN 978-1-449-34391-0.
- [11] Thomas, D., Hansson, D. H.: *Agile Web Development with Rails*. The Pragmatic Programmers, 2005, ISBN 978-0-9766940-0-7.

Příloha A

Diagram případů užití



Obrázek A.1: Diagram případů užití

Příloha B

Obsah příloženého CD

/technicka_zprava.pdf – text práce ve formátu PDF

/technicka_zprava_latex – zdrojové soubory technické zprávy

/src_aplikace – zdrojové soubory implementovaného systému

/readme.txt – návod na instalaci

/video.mp4 – video

/plakat.pdf – plakát