

Slovenská technická univerzita

Fakulta informatiky a informačných technológií

Ilkovičova 3, 842 16 Bratislava 4

Webový editor Texu - Webtex

Tímový projekt

Dokumentácia k inžinierskemu dielu

Tím č. 15 – code-X

Bc. Daniel Erban

Bc. Igor Hula

Bc. Kazimír Jaroszewicz

Bc. Radoslav Kontúr

Bc. Radovan Kuka

Vedúci tímového projektu: Ing. Tomáš Kramár

Akademický rok: 2011/2012

Obsah

ÚVOD.....	V
Účel a rozsah dokumentu.....	v
1 PRVÝ ŠPRINT - ALEXANDRA.....	1
2 DRUHÝ ŠPRINT - BOŽENA.....	3
2.1 Autentifikácia používateľov	3
2.1.1 Implementácia	3
2.1.2 Testovanie.....	5
2.2 Registrácia používateľov	5
2.2.1 Implementácia	5
2.2.2 Testovanie.....	6
2.3 Vytváranie projektov.....	6
2.3.1 Implementácia	7
2.3.2 Testovanie.....	9
2.4 Dashboard.....	10
2.4.1 Implementácia	10
2.4.2 Testovanie.....	11
2.5 Autorizácia	12
2.5.1 Implementácia	12
2.5.2 Testovanie.....	13
2.6 Úprava súborov.....	13
2.7 Verziovanie súborov.....	13
2.8 Dizajn a user experience	14

Úvod

Účel a rozsah dokumentu

Témou nášho tímového projektu je Webový editor pre TeX. Na webe je dostupných len málo editorov podobného zamerania. Problémom mnohých je to, že nazhŕňajú všetko čo by bežný používateľ očakával. Preto sme existujúce riešenia podrobili z analýze z ktorej pre nás vyplynuli užitočné informácie, akých chýb sa vyvarovať a akú hodnotu pridať.

Cieľom nášho tímového projektu je vytvorenie interaktívneho webového editora, ktorý by umožňoval úpravu zdrojových kódov v TeXu (LaTeX, ConTeXt...), ich kompiláciu a prezeranie výsledného PDF súboru. Jadro funkcionality editora by malo spočívať v podpore kolaborácie. Editor by mal podporovať verziovanie dokumentu, sledovanie zmien, zadávanie poznámok a tým umožniť plnohodnotnú kolaboratívnu tvorbu dokumentov. Webový editor by mal umožňovať tvorbu dokumentu aj mimo prostredia webu a poskytovať jednoduchú obojsmernú synchronizáciu s lokálnymi súbormi. Súčasťou projektu je aj vytvorenie API, aby sa dal editor jednoducho zaintegrovat' s niektorým z existujúcich desktopových editorov a rozšíriť ich o možnosť prezerania zmien a poznámok.

Riešenie projektu si vyžaduje širokú škálu zručností a technológií: HTML5&CSS3 (rozhranie); JavaScript, CoffeeScript, jQuery, Backbone.js (pre klientsku časť), Ruby, Rails, Node.js a PostgreSQL pre serverovú časť.

Projekt riešime agilnou vývojovou metodikou Scrum s dôrazom na testami riadený vývoj, kde je vývoj rozdelený na dvojtýždňové úseky, tzv. šprinty. V rámci každého šprintu robíme analýzu, návrh, implementáciu a testovanie zvolených funkcionalít, ktoré sú opísané tzv. príbehmi používateľa. Je zvykom šprinty pre ľahšiu identifikáciu pomenúvať podľa nejakého kľúča, my sme si ženské mená zoradené abecedne.

Predkladaný dokument, ktorý predstavuje dokumentáciu k inžinierskemu dielu, odráža toto rozdelenie a zachytáva prvé dva šprinty projektu.

1 Prvý šprint - Alexandra

V prvom šprinte sme venovali štúdiu, pre nás nového, programovacieho jazyka, oboznamovaním sa s Rails frameworkom a prípravou prostredia pre aplikáciu a development. Riešili sme následovné úlohy:

- Pridať projekt do Jenkinsa – Jenkins budeme používať na automatické nasadzovanie zmien do aplikácie
- Vytvoriť repozitár a skupinu na Gitbuse – na školskom Git serveri sme vytvorili skupinu Codex s repozitárom pre náš projekt WebTex, ktorý sa nachádza tu: <http://gitbus.fiit.stuba.sk/+codex>
- Rozbehať develop prostredie – rozhodli sme sa pre Ubuntu 11.10 distribúciu, ktorú sme nainštalovali ako virtuálny počítač pod VMware Player.
- Nainštalovať a nakonfigurovať PostgreSQL – je nainštalovaný v rámci develop prostredia aj produkčného servera
- Nainštalovať a nakonfigurovať passenger – nainštalovaný na serveri, slúži na jednoduché nasadzovanie aplikácie
- Vygenerovať základnú kostru Rails aplikácie – vygenerovaná príkazom „rails new webtex“
- Naštudovať ako funguje SYNCTEX – túto úlohu sme si nechali na neskorší šprint, kedy budeme riešiť zobrazovanie vygenerovaných PDF dokumentov. SyncTeX slúži na synchronizáciu medzi zdrojovým TeX dokumentom a vygenerovaným PDF dokumentom.
- Naštudovať ako funguje Git note
- Porovnať Codemirror a ACE – rozhodli sme sa pre CodeMirror na základe funkcií, ktoré poskytuje.
- Rozdistribúovať coding-conventions pre Ruby a Git – rozhodli sme sa ísť podľa štýlu, ktorý nám navrhol vedúci:

Coding conventions

Formátovanie:

- Používať ASCII znaky
- pri zarážkach používať dve medzery
- používať medzery v okolí operátorov
- nedávať medzery po znakoch (a [a pred znakmi) a]
- dávať dve medzery pred modifikátormi *if, unless, while, until, rescue*
- dávať zarážky tak hlboko aká je hĺbka *case*
- priávať prázdny riadok pred návratovou hodnotou metódy a medzi *def-mi*
- používať *Rdoc* a jeho konvencie na dokumentáciu API
- rozdelovať dlhé metódy prázdnymi riadkami do logických odstavcov
- dodržiavať maximálnu dĺžku riadku 80 znakov

- vyhýbať sa prázdny riadkom na konci

Syntax

- používať *def* s úvodzovkami ak sú prítomné argumenty
- nepoužívať *for* iba ak presne vieme prečo
- nepoužívať *then*
- používať *&&* a *||* pre boolové výrazy
- vyhýbať sa viacriadkovým *?:*, používať *if*
- preferujú sa zložené zátvorky pred *do...end*
- vyhýbať sa *return* pokiaľ nie je potrebný

Pomenovávanie

- používať *snake_case* pre metódy, tzn. slová oddelené podtržníkom
- pre názvy modulov a tried používať *CamelCase*, skratky nechať veľkými písmenami
- konštanty písať v *SCREAMING_SNAKE_CASE*
- používať *_* pri nepoužívaných premenných

Komentáre

- komentáre, ktoré sú dlhšie ako jedno slovo, majú rozlíšené veľké a malé písmená a používajú interpunkciu
- vyhýbať sa nadbytočným komentárom

GIT – commit message konvencie:

- všetky správy sú v anglickom jazyku
- správa obsahuje: sumarizáciu (cca 50 znakov), jeden prázdny riadok a telo, ktorého riadok by nemal byť dlhší ako 72 znakov, je v prítomnom čase

2 Druhý šprint - Božena

V druhom šprinte sme riešili nasledovné úlohy:

2.1 Autentifikácia používateľov

Ako používateľ sa chcem prihlásiť, aby som mohol mať privátne projekty, ktoré sú viditeľné len mne.

Vrámcami úlohy sme riešili nasledovné podúlohy:

- Nainštalovať a naštudovať DEVISE
- Nakonfigurovať DEVISE
- Upraviť vygenerovaný HTML do dizajnu
- Cucumber testy

2.1.1 Implementácia

Implementácia má obsahovať formulár, ktorým sa používateľ prihlási do systému. Mal by použiť kombináciu e-mailu a hesla. Pre autentifikáciu sme použili gem Devise. Je to modulárne autentifikačné riešenie pre Rails framework. Po inštalácii samotného gem-u sme vygenerovali model (users.rb), konfiguračný súbor a views pre všetky akcie. Model sme nakonfigurovali na ukladanie šifrovaných hesiel do databázy a validáciu používateľov, registráciu, zapamätanie si používateľov cez cookies, sledovanie prihlásení a obnovenie hesla. Samotný zdrojový kód vyzerá nasledovne:

Config/initializers/devise.rb

```
Devise.setup do |config|
  config.mailer_sender = "please-change-me-at-config-initializers-devise@example.com"
  require 'devise/orm/active_record'
  config.case_insensitive_keys = [ :email ]
  config.strip_whitespace_keys = [ :email ]
  config.stretches = Rails.env.test? ? 1 : 10
  config.use_salt_as_remember_token = true
  config.reset_password_within = 2.hours
  config.sign_out_via = :get
End
```

Models/devise.rb

```
1 class User < ActiveRecord::Base
2   has_many :projects
3
4   # Include default devise modules. Others available are:
5   # :token_authenticatable, :encryptable, :confirmable, :lockable, :timeoutable and :
6   devise :database_authenticatable, :registerable,
```



```

7         :recoverable, :rememberable, :trackable, :validatable
8
9     # Setup accessible (or protected) attributes for your model
10    attr_accessible :email, :password, :password_confirmation, :remember_me
11  end

```

Vygenerované views obsahovali viaceré obrazovky, najdôležitejší pre prihlasovanie bol view pre sign-in:

Views/devise/sessions/new.html.erb

```

1  <h2>Sign in</h2>
2
3  <%= form_for(resource, :as => resource_name, :url => session_path(resource_name)) do
4    <div><%= f.label :email %><br />
5    <%= f.email_field :email %></div>
6
7    <div><%= f.label :password %><br />
8    <%= f.password_field :password %></div>
9
10   <% if devise_mapping.rememberable? -%>
11     <div><%= f.check_box :remember_me %> <%= f.label :remember_me %></div>
12   <% end -%>
13
14   <div><%= f.submit "Sign in" %></div>
15 <% end %>
16
17 <%= render :partial => "devise/shared/links" %>

```

Úprava HTML kódu do dizajnu nebola zatiaľ spravená, kvôli personálnym problémom.

2.1.2 Testovanie

Testovali sme pomocou Cucumber testov, na preverenie autentifikácie boli použité nasledovné scenáre:

Názov scenára	Popis	Akcia
Logging in with non-existing account	Given I am logged in And no user with login tkramar@gmail.com exists When I login as tkramar@gmail.com with password „password“ Then I should not be logged in And I should see that i have entered wrong account information	Po navigovaní na prihlasovaciu stránku a vyplnení údajov pre neexistujúce konto sa zobrazí hláška „Invalid email or password“
Logging in with existing account	Given i am not logged in And a user with pin and tkramar@gmail.com password „secret“ exists When i login as tkramar@gmail.com with password „secret“ Then i should be logged in	Po navigovaní na prihlasovaciu stránku a vyplnení formulára s platnými údajmi sa používateľ prihlási a stránka ho presmeruje na domovskú stránku aplikácie

2.2 Registrácia používateľov

Ako používateľ sa chcem registrovať, aby som sa mohol prihlásiť a mať výhody z toho vyplývajúce

2.2.1 Implementácia

Registrácia používateľa bola implementovaná pomocou rovnakého gem-u ako autentifikácia – Devise. Bol nakonfigurovaný aby prijímal ako meno validný e-mail a heslo s minimálne šiestimi znakmi.

Vygenerovaný bol nasledovný view:

Views/devise/registrations/new.html.erb:

```
<h2>Sign up</h2>
<%= form_for(resource, :as => resource_name, :url =>
registration_path(resource_name)) do |f| %>
  <%= devise_error_messages! %>
  <div><%= f.label :email %><br />
<%= f.email_field :email %></div>
  <div><%= f.label :password %><br />
<%= f.password_field :password %></div>
  <div><%= f.label :password_confirmation %><br />
<%= f.password_field :password_confirmation %></div>
  <div><%= f.submit "Sign up" %></div>
<% end %>
<%= render :partial => "devise/shared/links" %>
```

Vytváral sa tu aj model pre user-a, ktorý vyzerá nasledovne:

User	
projects	Project
current_sign_in_ip	string
current_sign_in_at	datetime
sign_in_count	integer
encrypted_password	string
email	string
reset_password_token	string
last_sign_in_at	datetime
remember_created_at	datetime
last_sign_in_ip	string
reset_password_sent_at	datetime

2.2.2 Testovanie

Testovali sme pomocou Cucumber testov, na preverenie registrácie boli použité nasledovné scenáre:

Názov scenára	Popis	Akcia
Registering	Given no user with login tkramar@gmail.com exists When I register as tkramar@gmail.com and choose my password to be secret And I login as tkramar@gmail.com with password "secret" Then I should be logged in	Po navigovaní na Sign-up stránku a vyplnení formulára údajmi sa zobrazí informácia o úspešnej registrácii a automaticky sa používateľ prihlási
Registering with already taken account name	Given a user with login tkramar@gmail.com already exists When I register as tkramar@gmail.com and choose my password to be "secret" Then I should not be registered And I should see that the username tkramar@gmail.com is already taken	Po registrovaní s údajmi používateľa, ktorý už existuje v systéme sa zobrazí

2.3 Vytváranie projektov

Ako používateľ si chcem vytvoriť projekt, aby som mohol logicky zhlukovať dokumenty, ktoré budem písať.

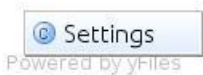
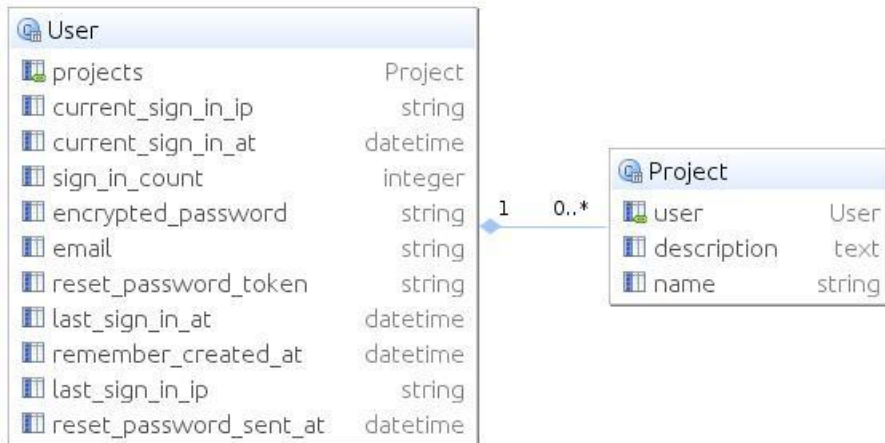
Vytváranie projektov v sebe zahŕňa tieto podúlohy:

- Navrhnuť a vytvoriť tabuľky pre vytváranie projektov
- HTML a štýlovanie
- Inštalácia GRIT gemu
- Ruby interakcia s Gritom
- Cucumber testy

Táto feature je úzko prepojená s Dashboardom, kde sa zobrazujú samotné projekty.

2.3.1 Implementácia

Bola vytvorená tabuľka pre projekty (viď obrázok), v ktorej sa ukladajú potrebné dáta.



Ďalej bol nainštalovaný gem Grit, ktorý dáva objektovo-orientovaný prístup do Git repozitárov pomocou Ruby. Pri vytváraní projektu sa prostredníctvom tohto gemu vytvorí na súborovom systéme repozitár s cestou: `git_path/ID_USER/ID_Project`, pričom `git_path` je nastavený v `config.yml` súbore a `id_user` je ID aktuálne prihláseného používateľa. Celú funkčnosť má na starosti controller `projects_controller.rb`:

```
1 class ProjectsController < ApplicationController
2   include ApplicationHelper
3
4   before_filter :find_project, :only => [:show, :edit, :update, :destroy]
5
6   def show
7     authorize! :read, @project
8   end
9
10  def new
11    @project = Project.new
12  end
```

```
13
14 def create
15   @project = Project.create(params[:project].merge(:owner => current_user))
16
17   if @project.save
18     flash[:notice] = t('project.created', :name => @project.name)
19     redirect_to dashboard_user_path
20   else
21     render :action => :new
22   end
23 end
24
25 def edit
26   authorize! :edit, @project
27 end
28
29 def update
30   if @project.update_attributes(params[:project])
31     flash[:notice] = t('project.updated', :name => @project.name)
32     redirect_to dashboard_user_path
33   else
34     render :action => :edit
35   end
36   authorize! :edit, @project
37 end
```

```

38
39 def destroy
40   if @project.destroy
41     flash[:notice] = t('project.deleted', :name => @project.name)
42     redirect_to dashboard_user_path
43   else
44     flash[:alert] = t('project.not_deleted', :name => @project.name)
45     redirect_to dashboard_user_path
46   end
47   authorize! :edit, @project
48 end
49
50 private
51
52 def find_project
53   @project = Project.find(params[:id])
54 end
55 end

```

Pomocou metód create, show, edit a destroy sa vytvorí nový projekt, zobrazia existujúce projekty pre aktuálne prihláseného používateľa, zmení sa názov projektu a zmaže sa celý projekt.

2.3.2 Testovanie

Pri testovaní sa predpokladá, že používateľ je prihlásený

Názov scenára	Popis	Akcia
Creating project	When I go to my dashboard When I follow link to create new project And I create a new project called "iit.src" Then I should have a project "iit.src" among my projects	Po navigovaní do dashboard, kliknem na tlačítko "new project", vyplním príslušné údaje a kliknem na "Create project". V databáze sa vytvorí záznam a na dashboard sa zobrazí mnou vytvorený projekt

Editing project	Given I have a project "iit.src" And I edit the project When I change its name to "iit.src 2012" and its description to "iit 2009" Then I should be on my dashboard And I should see that I have a project "iit.src 2012"	Na dashboarde mám vytvorený projekt "iit.src". po kliknutí na link "edit" môžem meniť názov projektu a popis. Keď ho zmením danými údajmi a kliknem na tlačítko "Update Project", v dashboarde vidím, že projekt je zmenený.
Delete project	Given I have a project "iit.src" And I am on the dashboard When I delete the project Then I should have no project "iit.src"	V dashboarde mám projekt "iit.src". Po kliknutí na tlačítko "delete" sa daný projekt vymaže a už nie je zobrazený v dashboarde.

2.4 Dashboard

Ako používateľ chcem vidieť celú pracovnú plochu na jednom mieste – dashboard, aby som mal rýchly prehľad o mojich projektoch a súboroch.

2.4.1 Implementácia

Dashboard je východiskový bod po prihlásení používateľa. Sú v ňom prehľadne zobrazené všetky projekty a ich súbory, na ktorých používateľ pracuje.

Implementovaný je vo view dashboard.html.erb, využíva funkcionality vytvárania projektov z predošlej úlohy.

Views/users/dashboard.html.erb:

```

1 <h3>Dashboard</h3>
2
3 <table title="dashboard" border="1" cellpadding="3" cellspacing="0" id="dashboard">
4   <tr>
5     <th>Name</th>
6     <th>Owner</th>
7     <th>Last modified</th>
8     <th></th>
9     <th></th>
10  </tr>
11  <% @projects.each do |project| %>

```

```

12 <tr>
13 <td><%=link_to project.name, project_path(project)%></td>
14 <td><%= project.owner.email %></td>
15 <td><%= time_ago_in_words(project.updated_at)%> ago</td>
16 <td><%= link_to 'Edit', edit_project_path(project) %></td>
17 <td><%= link_to 'Delete', project, :confirm => 'Are you sure you want to delete
=> :delete %></td>
18 </tr>
19 <% end %>
20 </tr>
21 </table>
22
23 <br>
24 <%= link_to 'New Project', new_project_path %>

```

2.4.2 Testovanie

Názov scenára	Popis	Akcia
Seeing dashboard after login	Given I am logged out When I log in Then I should be on my dashboard page	Po prihlásení sa zobrazí dashboard
Seeing list of projects and their files in dashboard	Given I have the following projects and files: Project: tpcup Files: main.tex, overview.jpg Project: iitsrc Files: iitsrc.tex, bibliography.bib When I go to my dashboard Then I should see that I have project tpcup And I should see that project tpcup has files main.tex, overview.jpg Then I should see that I have project iit.src And I should see that project	V dasbhoarde mám zoznam mojich projektov spolu so súbormi, ktoré k nim patria

	iitsrc has files iitsrc.tex, bibliography.bib	
Seeing last change for a project and each file	Given I am logged in Given I have project iitsrc And the project has file main.tex which was last modified 1 days ago And the project has file bibliography.bib which was last modified 3 days ago When I go to my dashboard Then I should see that the project iitsrc was last modified 1 day ago And I should see that the file main.tex was last modified 1 day ago And I should see that the file bibliography.bib was last modified 3 days ago	V dashboarde je zoznam projektov so súbormi a údajmi o poslednej zmene súboru/projektu

2.5 Autorizácia

Ako používateľ by som mal byť jediný, ktorý vidí moje súbory, aby som obmedzil prístup k projektom, ktoré vytváram.

Vrámci tejto úlohy sme riešili nasledovné podúlohy:

- Nainštalovať a naštudovať CanCan
- Definovať CanCan ability
- Zavolať CanCan pred zobrazením projektu
- Cucumber testy

2.5.1 Implementácia

Pri autorizácii sa využíva gem CanCan, ktorý obmedzuje prístup používateľov k prostriedkom. Všetky prístupy sú riešené cez jednu Ability triedu. Táto trieda je nastavená tak, aby zobrazovala používateľom iba tie projekty, ktorého sú Owner:

```

1 class Ability
2   include CanCan::Ability
3
4   def initialize(user)
5     user ||= User.new
6     can [:read, :edit], Project do |project|
7       project.owner == user

```

8	end
9	end
10	end

2.5.2 Testovanie

Testovanie sme vykonávali pomocou Cucumber testov.

Názov scenára	Popis	Akcia
Only author can see his projects	Given I am logged in as tkramar@gmail.com And I have a project "iit.src" And I go to my dashboard Then I should be able to see the project "iit.src" When I relogin as mbielik@gmail.com And I go to my dashboard Then I should not see the project "iit.src"	Pri prihlásení sa dashboard zobrazuje iba projekty aktuálne prihláseného používateľa
Projects are visible to their authors	Given I am logged in as tkramar@gmail.com And I have a project "iit.src" And there is another user and she has a project "papers" When I go to the "iit.src" project page Then I should be able to see the project When I go to the "papers" project page Then I should not be able to see the project	Pri manuálnom zadaní projektu do linku, na ktorý nemá aktuálny používateľ právo, sa zobrazí error 403 – forbidden.

2.6 Úprava súborov

Ako používateľ, chcem mať možnosť upravovať súbory, aby som mohol vytvárať dokumenty na webe, bez nutnosti inštalovať TEX na mojom počítači.

V čase písania dokumentácie ešte táto úloha nebola vyriešená z dôvodu personálnych problémov v time, presun do ďalšieho šprintu.

2.7 Verziovanie súborov

Ako používateľ chcem vidieť verzie dokumentu, aby som videl ako sa dokument vyvíjal v čase.

Vrámci úlohy sme riešili následovné podúlohy:

- Zobrazenie zoznamu revízií
- Zobrazenie konkrétnej revízie
- Cucumber testy

Verziovanie dokumentov projektu je jeho dôležitá súčasť v kontexte vývoja dokumentu čase. Jednotlivé verzie dokumentu sa vytvoria v prípade, že používateľ, ktorý je na tomto dokumente oprávnený pracovať, upraví tento dokument a uloží ho. Úlohou verziovania je pre daný projekt zobraziť všetky k nemu prislúchajúce verzie a takisto zobraziť aj konkrétnu verziu dokumentu.

V čase písania dokumentácie ešte táto úloha nebola úplne vyriešená z dôvodu personálnych problémov v time, presun do ďalšieho šprintu.

2.8 Dizajn a user experience

Ako prevádzkovateľ editora chcem, aby editor dobre vyzeral a pohodlne sa používal, aby ľudia za neho boli ochotní platiť.

V čase písania dokumentácie ešte táto úloha nebola vyriešená z dôvodu personálnych problémov v time, presun do ďalšieho šprintu.