



SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
Fakulta informatiky a informačných technológií

# **Znalosti a zručnosti študentov**

## **Tímový projekt**

### **Dokumentácia k inžinierskemu dielu**

#### **Tím č.16 – WeKnowIT**

Bc. Jakub Šalmík

Bc. Patrik Polakovič

Bc. Adam Kobyda

Bc. Michal Chylik

Bc. Peter Ivanec

Bc. Anton Szórád

---

Vedúci tímového projektu: Ing. Michal Holub

Akademický rok: 2011/2012

# Obsah

---

|  |    |
|--|----|
| Úvod .....   | 6  |
| Účel a rozsah dokumentu .....                      | 6  |
| 1. Prvý šprint – Trójska vojna .....               | 7  |
| 1.1. US1 Návrh šablóny GUI .....                   | 7  |
| 1.1.1. Analýza .....                               | 7  |
| 1.1.2. Návrh .....                                 | 8  |
| 1.1.3. Implementácia .....                         | 9  |
| 1.1.4. Testovanie .....                            | 12 |
| 1.2. US4 Editácia profilu .....                    | 12 |
| 1.2.1. Analýza .....                               | 12 |
| 1.2.2. Návrh .....                                 | 13 |
| 1.2.3. Implementácia .....                         | 15 |
| 1.2.4. Testovanie .....                            | 16 |
| 1.3. US 6 – Zadanie znalostí .....                 | 18 |
| 1.3.1. Analýza .....                               | 18 |
| 1.3.2. Návrh .....                                 | 19 |
| 1.3.3. Implementácia .....                         | 20 |
| 1.3.4. Akceptačné testy .....                      | 20 |
| 1.4. US 11 - Vytvorenie prvotnej bázy pojmov ..... | 21 |
| 1.4.1. Analýza .....                               | 21 |
| 1.4.2. Návrh .....                                 | 21 |
| 1.4.3. Implementácia .....                         | 21 |
| 2. Druhý šprint – Prvá púnska vojna .....          | 23 |
| 2.1. US2 Prihlasovanie používateľa .....           | 23 |
| 2.1.1. Analýza .....                               | 23 |
| 2.1.2. Návrh .....                                 | 24 |
| 2.1.3. Implementácia .....                         | 25 |
| 2.1.4. Testovanie .....                            | 27 |
| 2.2. US3 Prezeranie profilu .....                  | 29 |
| 2.2.1. Analýza .....                               | 29 |
| 2.2.2. Návrh .....                                 | 29 |
| 2.2.3. Implementácia .....                         | 31 |
| 2.2.4. Testovanie .....                            | 32 |

|        |   |    |
|--------|---|----|
| 2.3.   | US7 Zobrazenie vlastných znalostí .....             | 32 |
| 2.3.1. | Analýza .....                                       | 32 |
| 2.3.2. | Návrh .....   | 33 |
| 2.3.3. | Implementácia .....                                 | 34 |
| 2.3.4. | Akceptačné testy .....                              | 35 |
| 2.4.   | US12 Kategorizácia znalostí .....                   | 35 |
| 2.4.1. | Analýza .....                                       | 35 |
| 2.4.2. | Návrh .....   | 35 |
| 2.4.3. | Implementácia .....                                 | 36 |
| 3.     | Tretí šprint – Druhá púnska vojna .....             | 38 |
| 3.1.   | US8 Zobrazenie znalostí študenta .....              | 38 |
| 3.1.1. | Analýza .....                                       | 38 |
| 3.1.2. | Návrh .....   | 39 |
| 3.1.3. | Implementácia .....                                 | 39 |
| 3.1.4. | Testovanie .....                                    | 42 |
| 3.2.   | US5 Zobrazenie kľúčových slov v systéme .....       | 43 |
| 3.2.1. | Analýza .....                                       | 43 |
| 3.2.2. | Návrh .....   | 44 |
| 3.2.3. | Implementácia .....                                 | 45 |
| 3.2.4. | Testovanie .....                                    | 47 |
| 4.     | Štvrtý šprint – Storočná vojna .....                | 48 |
| 4.1.   | US9 Vyhľadanie študenta .....                       | 48 |
| 4.1.1. | Analýza .....                                       | 48 |
| 4.1.2. | Návrh .....   | 49 |
| 4.1.3. | Implementácia .....                                 | 50 |
| 4.1.4. | Akceptačné testy .....                              | 51 |
| 4.2.   | US14 Pridanie novej znalosti .....                  | 51 |
| 4.2.1. | Analýza .....                                       | 51 |
| 4.2.2. | Návrh .....   | 52 |
| 4.2.3. | Implementácia .....                                 | 53 |
| 4.2.4. | Testovanie .....                                    | 55 |
| 4.3.   | US18 Nastavenie ratingu pri zadávaní znalosti ..... | 56 |
| 4.3.1. | Analýza .....                                       | 56 |
| 4.3.2. | Návrh .....   | 57 |
| 4.3.3. | Implementácia .....                                 | 57 |

|           |   |    |
|-----------|---|----|
| 4.3.4.    | Akceptačné testy .....                                | 59 |
| Príloha A | .....   | 60 |
|           | Dokumentácia k inžinierskemu dielu po 2.šprinte ..... | 60 |
| Úvod      | .....   | 64 |
|           | Účel a rozsah dokumentu.....                          | 64 |
| 1.        | Prvý šprint – Trójska vojna .....                     | 65 |
| 1.1.      | US1 Návrh šablóny GUI .....                           | 65 |
| 1.1.1.    | Analýza .....   | 65 |
| 1.1.2.    | Návrh .....   | 66 |
| 1.1.3.    | Implementácia.....                                    | 67 |
| 1.1.4.    | Testovanie .....                                      | 69 |
| 1.2.      | US4 Editácia profilu .....                            | 69 |
| 1.2.1.    | Analýza .....   | 69 |
| 1.2.2.    | Návrh .....   | 70 |
| 1.2.3.    | Implementácia.....                                    | 71 |
| 1.2.4.    | Testovanie .....                                      | 72 |
| 1.3.      | US 6 – Zadanie znalostí.....                          | 72 |
| 1.3.1.    | Analýza .....   | 72 |
| 1.3.2.    | Návrh .....   | 73 |
| 1.3.3.    | Implementácia.....                                    | 74 |
| 1.3.4.    | Akceptačné testy .....                                | 74 |
| 1.4.      | US 11 - Vytvorenie prvotnej bázy pojmov.....          | 75 |
| 1.4.1.    | Analýza .....   | 75 |
| 1.4.2.    | Návrh .....   | 75 |
| 1.4.3.    | Implementácia.....                                    | 75 |
| 2.        | Druhý šprint – Prvá púnska vojna.....                 | 76 |
| 2.1.      | US2 Prihlasovanie používateľa .....                   | 76 |
| 2.1.1.    | Analýza .....   | 76 |
| 2.1.2.    | Návrh .....   | 77 |
| 2.1.3.    | Implementácia.....                                    | 78 |
| 2.1.4.    | Testovanie .....                                      | 80 |
| 2.2.      | US3 Prezeranie profilu.....                           | 81 |
| 2.2.1.    | Analýza .....   | 82 |
| 2.2.2.    | Návrh .....   | 82 |
| 2.2.3.    | Implementácia.....                                    | 83 |

|        |   |    |
|--------|---|----|
| 2.2.4. | Testovanie .....                        | 83 |
| 2.3.   | US7 Zobrazenie vlastných znalostí ..... | 84 |
| 2.3.1. | Analýza .....                           | 84 |
| 2.3.2. | Návrh .....                             | 84 |
| 2.3.3. | Implementácia .....                     | 85 |
| 2.3.4. | Akceptačné testy .....                  | 86 |
| 2.4.   | US12 Kategorizácia znalostí.....        | 86 |
| 2.4.1. | Analýza .....                           | 86 |
| 2.4.2. | Návrh .....                             | 86 |
| 2.4.3. | Implementácia .....                     | 87 |

# Úvod

---

## Účel a rozsah dokumentu

Každý študent disponuje určitými znalosťami a zručnosťami. V súčasnosti neexistuje vo vysokoškolskom prostredí systém, ktorý by tieto údaje dokázal zbierať a vyhodnocovať. Pritom sú to údaje, ktoré majú vysokú cenu jednak pre školu, pri vyberaní vhodných študentov na stáže, projekty, bakalárske či diplomové práce, ako aj pre firmy, ktoré sa väčšinou o študentov začínajú zaujímať už v čase ich štúdia na vysokej škole. V neposlednom rade ide o užitočné informácie aj pre študentov samotných, ktorí by si za predpokladu, že poznajú zručnosti a znalosti ostatných spolužiakov, mohli ľahšie vyberať spolupracovníkov do rôznych tímových projektov.

Cieľom nášho tímového projektu je vytvorenie systému na zhromažďovanie, uchovávanie a prezentáciu znalostí a zručností študentov.

Ako implementačné prostredie sme si zvolili v súčasnosti najpoužívanejší jazyk pre tvorbu webových aplikácií, jazyk PHP. Vybrali sme si aplikačný rámec Yii, ktorý sa nám po analýze dostupných možností v tejto oblasti, javil najvhodnejší. Voľba databázového systému bola podmienená požiadavkami zákazníka, preto je zvoleným databázovým systémom PostgreSQL.

Projekt je riešený agilnou vývojovou metodikou Scrum. V nej je vývoj rozdelený na dvojtýždňové úseky, tzv. šprinty. V rámci každého šprintu robíme analýzu, návrh, implementáciu a testovanie zvolených používateľských príbehov. Pre pomenovanie jednotlivých šprintov sme vybrali zoznam najväčších vojen v histórii ľudstva.

Predkladaný dokument, ktorý predstavuje dokumentáciu k inžinierskemu dielu, zachytáva prvé dva šprinty vývoja systému.

# 1. Prvý šprint – Trójska vojna

---

V rámci prvého šprintu sme riešili nasledovné používateľské príbehy.

- US1 – Návrh šablóny GUI
- US4 – Editácia profilu
- US6 – Zadanie znalostí
- US11 – Vytvorenie prvotnej bázy pojmov

## 1.1. US1 Návrh šablóny GUI

---

Grafický návrh používateľského rozhrania a jeho následná implementácia.

### 1.1.1. Analýza

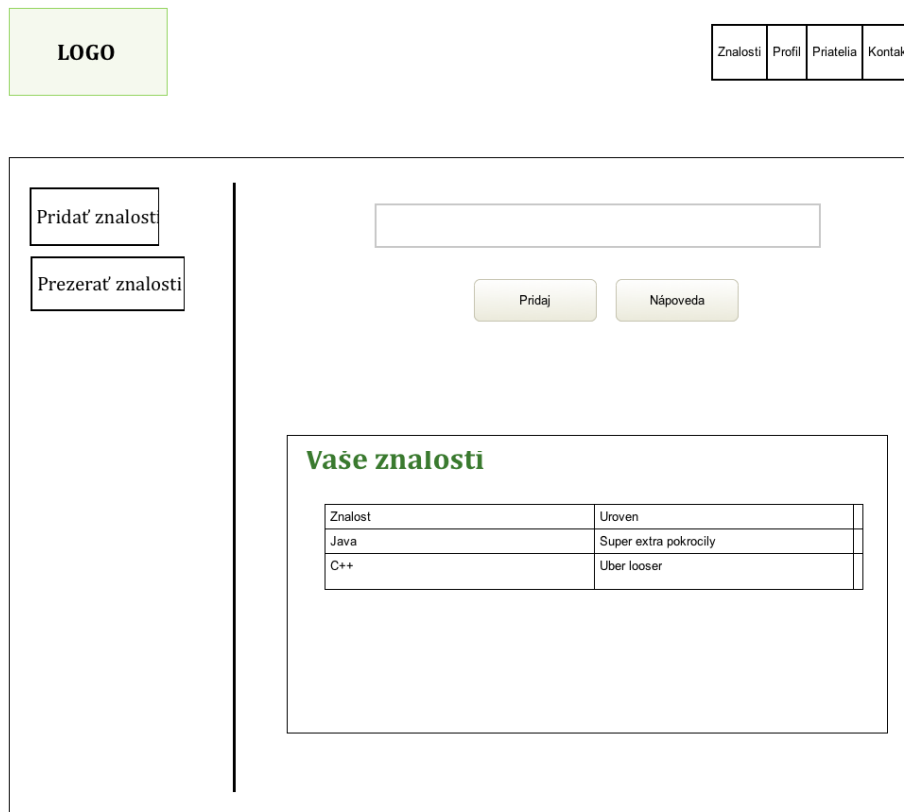
Prvotnou požiadavkou celého systému bola prehľadnosť a použiteľnosť rozhrania. Musíme zabezpečiť používateľskú spokojnosť a zabezpečiť lepšiu orientáciu v systéme.

V systéme sa budú nachádzať klasické prvky webových stránok ako sú:

- Formulárové polia
- Tlačidlá pre odosielanie údajov
- Tabuľky

Všetky prvky nemajú byť len graficky príťažlivé, ale hlavne funkčné a účelné.

Na obrázku 1.1 je znázornený wireframe (obrys) aplikácie, kde je znázornené rozloženie celej stránky. Obsahuje dve navigačné menu pre zlepšenie orientácie v systéme. Taktiež znázorňuje formulár a tabuľku (iba obrys nie reálny návrh).



Obr. 1.1 : Wireframe používateľského rozhrania

Keďže sme sa nedržali žiadneho z pôvodných systémov, ktoré boli pred tým realizované, návrh rozhrania sme robili podľa štandardov moderných webových technológií s prihliadnutím na to, čo nám v súčasnosti web poskytuje.

Požiadavky na rozhranie od zákazníka sú taktiež minimálne, keďže zákazník sa s podobným systémom ešte nestretol. To nám dalo voľnú ruku, samozrejme s prihliadnutím na štandardy webových technológií.

### 1.1.2.Návrh

Pri návrhu bolo prioritou zvýrazniť prvky, ktoré sú dôležité. Horné horizontálne menu bude farebne zvýraznené, pre lepšiu orientáciu bude zvýraznená aj aktuálna položka v menu.

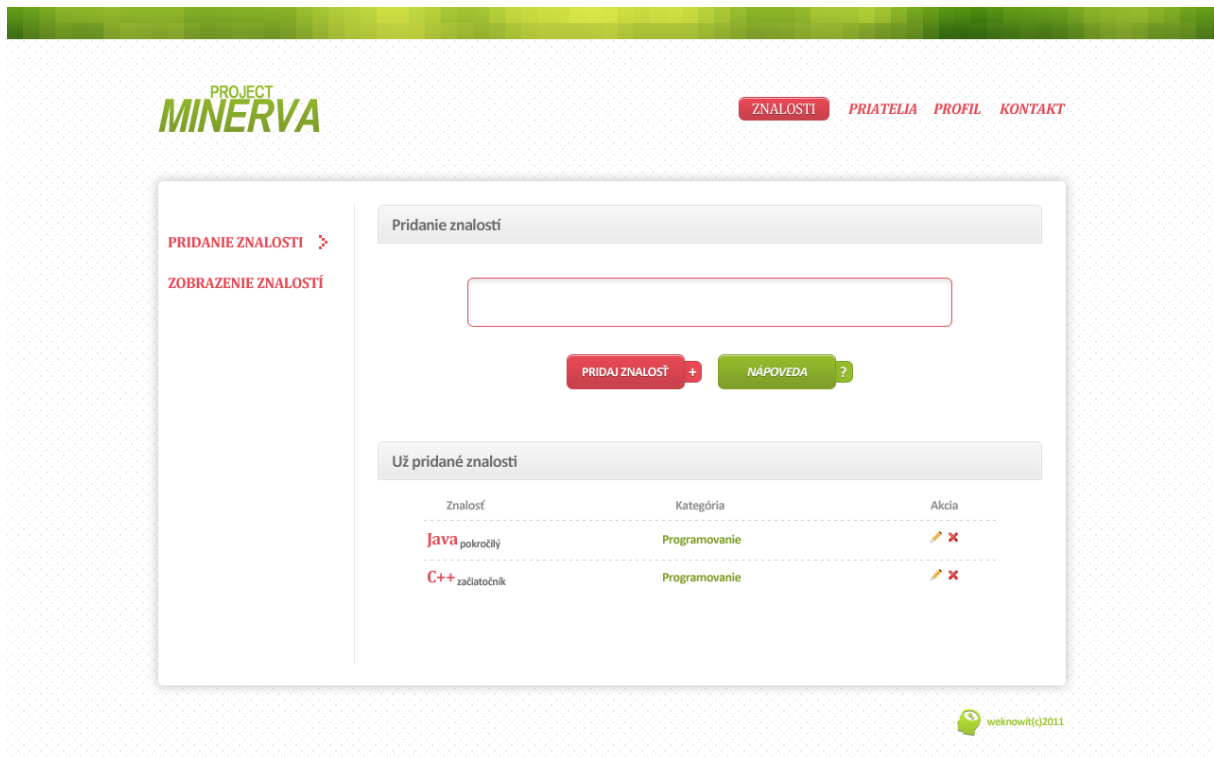
Taktiež navrhujeme aby centrálny panel bol opticky oddelený od bočného vertikálneho menu, ktoré bude mať iba orientačný charakter, čo ale neuberá na jeho dôležitosť v použiteľnosti systému.

Centrálny panel bude združovať všetky funkcie poskytujúce systémom. V závislosti od kombinácie horizontálneho a vertikálneho menu bude zobrazovať obsah, ktorý bude dobre čitateľný a štruktúrovaný.

Celý systém bude v moderných a výrazných farbách, aby sme predišli fádnosti, ale zachovali profesionalitu a čistý vzhľad. Dobrý výber farieb je kľúčový a bude systém charakterizovať.

Na obrázku 1.2 je znázornený grafický návrh obrazovky systému.





Obr. 1.2 : Grafický návrh obrazovky systému

### 1.1.3. Implementácia

Implementácia grafického návrhu sa skladala z viacerých častí.

#### 1. Nastrihanie obrázkov v požadovanej forme.

Všetky obrázky boli nastrihané a uložené vo formáte png, ktorý je dnes štandardom na poli webového dizajnu.

#### 2. Vytvorenie HTML štruktúry

Z predloženého návrhu bolo potrebné vymedziť jednotlivé prvky tak aby sa dali transformovať na elementy zrozumiteľného jazyku HTML, konkrétne verzii 5. Maximum prvkov bolo implementovaných pomocou elementov DIV. Samotné HTML elementy boli implementované tak, aby neobsahovali žiadne vnorené informácie o ich štýle. Všetky vlastnosti sú definované v súboroch css. Výslednú štruktúru kódu je možné vidieť nižšie.

```
<!DOCTYPE html>
<html lang="en">
  <head>

  <title>Projekt Minerva</title>

  <meta charset="utf-8" />
```

```
<meta name="author" content="WeKnowIT">
  <meta name="description" content="Projekt Minerva, zameraný za zber zručností a
znalostí študentov">
  <meta name="keywords" content="minerva, znalost, znalosti, zrucnost, zrucnosti, studenti,
fiit">

<link rel="stylesheet" href="css/reset.css" media="all">
<link rel="stylesheet" href="css/style.css" media="all">
<link rel="stylesheet" href="css/form.css" media="all">

<script src="js/custom-form-elements.js" type="text/javascript"></script>

</head>

<body>

  <div id='top_banner'></div>

  <!-- header -->

  <div id='top_container'>
    <div id='top_logo'>
      <a href='#'><img src='design/logo.png'></a>
    </div>
    <div id='top_menu'>
      <ul>
        <li><a href='#'>ZNALOSTI</a></li>
        <li><a href='#'>PRIATELIA</a></li>
        <li class='active_menu'><a href='#'>PROFIL</a></li>
        <li><a href='#'>KONTAKT</a></li>
      </ul>
    </div>
  </div>

  <!-- body -->

  <div id='body_container'>
    <div id='body_top'></div>
    <div id='body_fix2'>
    <div id='body_fix'>
      <div id='body_left'>
        <ul>
          <li><a href='#'>PRIDANIE ZNALOSTÍ</a></li>
          <li class='active_menu'><a href='#'>ZOBRAZENIE ZNALOSTÍ</a></li>
          <li><a href='#'>VYMAZANIE ZNALOSTÍ</a></li>
        </ul>
      </div>
      <div id='body_right'>
        <div id='content'>

          <!-- obsah -->
```

```
<h1>Pridanie znalostí</h1>

<p id='skill_form'>

  <input type='text' class='input_skill' value='Java EE'><br>
  <input type='button' class='button_add' value='PRIDAJ ZNALOSŤ'>
  <input type='button' class='button_help' value='NÁPOVEDA'>

</p>

<!-- koniec obsahu -->

</div>
</div>
</div>
</div>
<div id='body_bottom'></div>
<div id='footer'>
  <div id='footer_head'></div>
  <div id='footer_text'>weknowit(c)2011</div>
</div>
</div>

</body>
</html>
```

### 3. Vytvorenie CSS súborov

Pre potreby projektu boli vytvorené tri samostatné CSS súbory.

- reset.css – obsahuje vynulovanie všetkých atribútov, pre všetky definované html elementy. Tento postup eliminuje väčšinu problémov spojených s rozdielnym základným nastavením webových prehliadačov, teda rôznym zobrazovaním rovnakých prvkov.
- style.css - obsahuje definície všetkých elementov používaných na jednotlivých stránkach.
- form.css – obsahuje definície štýlov pre formulárové elementy ako checkbox, radio alebo select. Tieto elementy sa za normálnych okolností nedajú ľubovoľne meniť, lebo ich vzhľad je definovaný prehliadačom, preto bol použitý hotový skript dostupný na <http://ryanfait.com/resources/custom-checkboxes-and-radio-buttons/>, ktorý následne umožní elementy formulára upravovať pomocou CSS.

### 4. Optimalizácia pre prehliadače

Po dokončení predchádzajúcich krokov, bolo nutné spraviť optimalizáciu pre najpoužívanejšie prehliadače. Problém so zlým zobrazením sa vyskytol len pri formulároch v internet exploreri, ktoré boli následne vyriešené. Iné chyby sa neobjavili hlavne vďaka opatreniam uvedeným v predchádzajúcich bodoch.

## 1.1.4. Testovanie

Boli navrhnuté 2 testovacie scenáre pre otestovanie kompatibility GUI v internetových prehliadačoch.

|                           |  |  |  |        |
|---------------------------|--|--|--|--------|
| <b>Názov</b>              | Kompatibilita GUI v známych prehliadačoch (okrem Internet Explorer)                                |  | <b>ID Testu</b>                                  | US1-01 |
| <b>Rozhranie</b>          | Všetky   |  | <b>ID UC</b>                                     | -      |
| <b>Účel</b>               | Zaistenie správneho zobrazenia GUI vo všetkých majoritných prehliadačoch (okrem Internet Explorer) |  |  |        |
| <b>Vstupné podmienky</b>  | Žiadne   |  |  |        |
| <b>Výstupné podmienky</b> | Žiadne   |  |  |        |
| <b>Krok</b>               | <b>Akcia</b>   | <b>Očakávaná reakcia</b>                 | <b>Skutočná akcia</b>                            |        |
| 1.                        | Zobrazenie stránky vo všetkých majoritných prehliadačoch.  | Správne zobrazenie všetkých súčastí GUI. | GUI zobrazené správne vo všetkých prehliadačoch. |        |

Tab. 1.1 Akceptačný test US1-01

|                           |  |  |                       |        |
|---------------------------|--|--|-----------------------|--------|
| <b>Názov</b>              | Kompatibilita GUI v prehliadači Internet Explorer                  |  | <b>ID Testu</b>       | US1-02 |
| <b>Rozhranie</b>          | Všetky   |  | <b>ID UC</b>          | -      |
| <b>Účel</b>               | Zaistenie správneho zobrazenia GUI v prehliadači Internet Explorer |  |                       |        |
| <b>Vstupné podmienky</b>  | Žiadne   |  |                       |        |
| <b>Výstupné podmienky</b> | Žiadne   |  |                       |        |
| <b>Krok</b>               | <b>Akcia</b>   | <b>Očakávaná reakcia</b>                 | <b>Skutočná akcia</b> |        |
| 1.                        | Zobrazenie stránky prehliadači Internet Explorer                   | Správne zobrazenie všetkých súčastí GUI. | GUI zobrazené správne |        |

Tab. 1.2 Akceptačný test US1-02

## 1.2. US4 Editácia profilu

Používateľ bude mať možnosť úpravy základných údajov vo svojom profile.

### 1.2.1. Analýza

Keďže pre väčšinu používateľov systému bude registráciou do systému ich prvé prihlásenie, je umožnenie editovania základných údajov profilu nutnosťou. Pri prvej návšteve editácie profile to bude preto skôr vypĺňanie prázdnych položiek profilu, to je však z hľadiska systému presne rovnaká činnosť ako úprava už vyplnených údajov.

Tento jednoduchý profil bude obsahovať naozaj len základné informácie o používateľovi, teda neobsahuje žiadne znalosti. Prihlasovacie údaje, teda login a heslo, nebude možné meniť, keďže sa využívajú prihlasovacie údaje z AIS. Ostatné položky už budú upravovateľné.

Cieľom je ponúknuť jednoduchý a prehľadný formulár, ktorý bude umožňovať úpravu všetkých základných údajov. Samozrejmosťou je, že bude obsahovať prednastavené hodnoty podľa aktuálnych údajov o prihlásenom používateľovi.

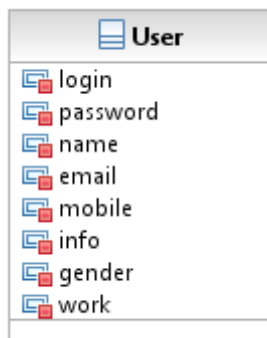
Pred vytvorením prezentačnej vrstvy profilu používateľa je nutné vytvoriť doposiaľ neexistujúcu údajovú vrstvu, teda reprezentáciu profilu používateľa v rámci databázy.

| Prípád použitia             | Editácia Profilu  |
|-----------------------------|---|
| <b>ID</b>                   | 1   |
| <b>Stručný popis</b>        | Používateľ si edituje základné údaje svojho profilu prostredníctvom formulára.  |
| <b>Primárny aktéri</b>      | Prihlásený používateľ   |
| <b>Vedľajší aktéri</b>      | Žiadny  |
| <b>Vstupné podmienky</b>    | Používateľ je prihlásený  |
| <b>Hlavný scenár</b>        | <ol style="list-style-type: none"> <li>1. Prípád použitia začína po zobrazení formuláru na editáciu profilu.</li> <li>2. Používateľ prepíše položky formulára, ktoré chcel editovať.</li> <li>3. Používateľ odošle formulár.</li> <li>4. Systém spracuje formulár a uloží údaje do databázy.</li> </ol> |
| <b>Výstupné podmienky</b>   | Žiadne  |
| <b>Alternatívne scenáre</b> | Žiadne  |

Tab.1.3 Prípád použitia ID 1

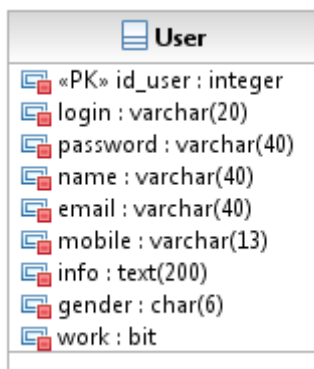
## 1.2.2.Návrh

Prvým krokom návrhu bolo vytvorenie logického a fyzického modelu profilu používateľa. Logický model je znázornený na Obr.1.3.



Obr.1.3 Logický model entity User

Pre fyzický model (Obr.1.4) sa doplnil jednoznačný identifikátor `id_user`, ktorý je zároveň primárnym kľúčom a má vlastnosť `autoincrement`. Stĺpce `gender` a `work` sú dátového typu číselník. `Gender` môže nadobúdať hodnoty `male(muž)` a `female(žena)`. `Work` môže nadobúdať taktiež dve hodnoty, `áno(yes)` a `nie(no)` a vyjadruje údaj o tom či používateľ je momentálne zamestnaný.



Obr.1.4 Fyzický model entity User

Ďalším krokom bol návrh formulára a jeho logiky na základe prípadu použitia ID 1, ktorý sme uviedli vyššie. Návrh formulára je možné vidieť na Obr.1.5. Formulár obsahuje potrebné polia na vyplnenie osobných údajov. Polia meno a osobné číslo sú needitovateľné, ale sprehľadňujú celý formulár. Emailová adresa sa overuje interaktívne pomocou technológie ajax. Logiku overovania korektnosti emailovej adresy sme neimplementovali, je súčasťou frameworku, ktorý používame pri vývoji.

disabled input, neda sa menit obsah

ajaxové overenie emailu

Priezvisko a meno: Mak Jozef, Ing.

Osobné číslo: 48079

Email: jozef.mak@stu. (with error icon)

Telefón: +421 911 222 890

Pohlavie:  Muž  Žena

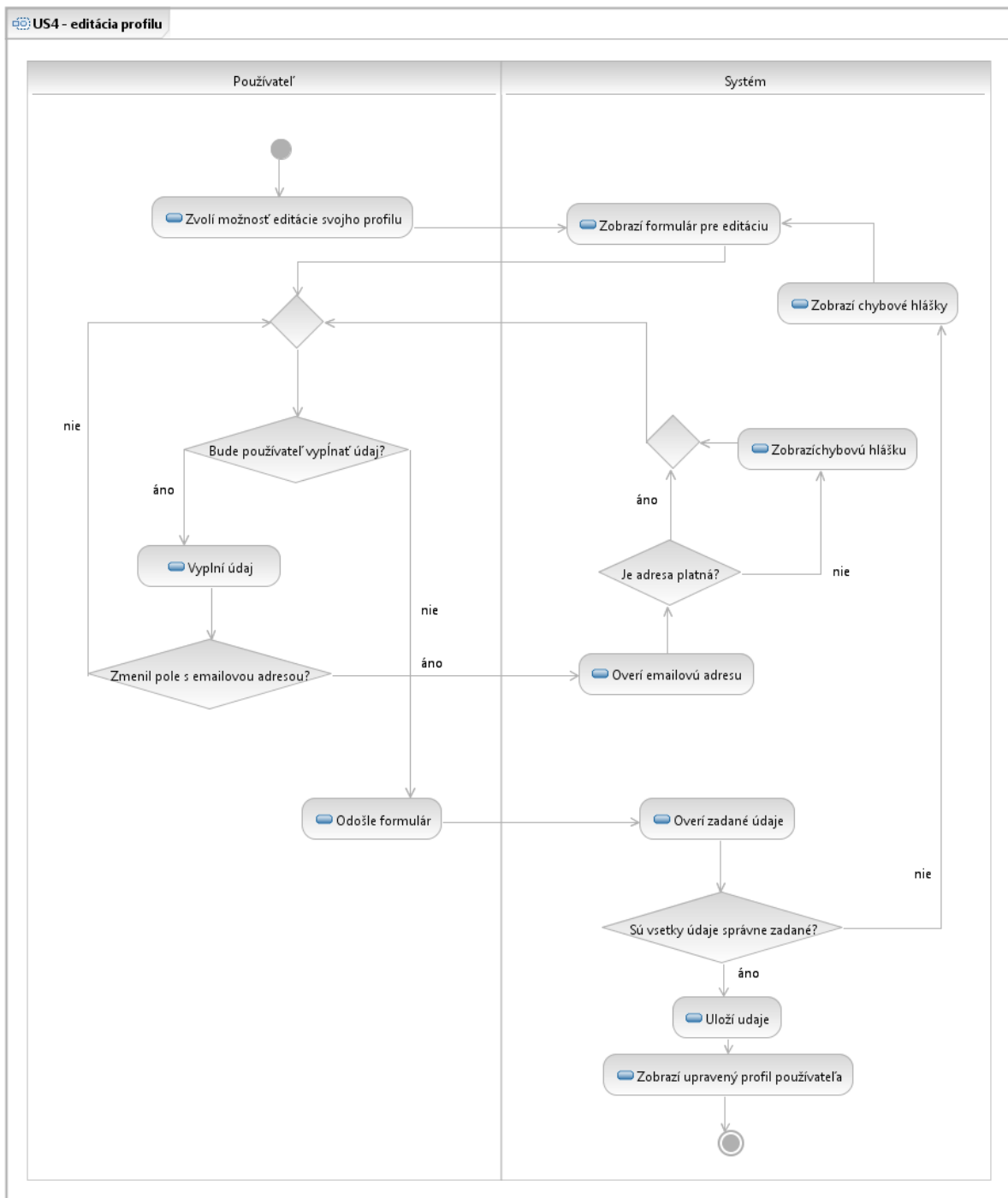
Práca:  Ano  Nie

O mne: Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries.

**Uložiť**

Obr.1.5 Návrh formulára pre editáciu profilu

Obr.1.6 znázorňuje diagram aktivít pre daný prípad použitia. Diagram zahŕňa všetky prípady, ktoré môžu pri editácii profilu nastať. Prípad použitia sa odohráva medzi používateľom a systémom.



Obr.1.6 Diagram aktivít

### 1.2.3. Implementácia

Pri implementácii boli využité výhody frameworku Yii, ktorý využívame pre náš projekt. Po vytvorení tabuľky User v databáze bol využitý CRUD generátor, ktorý automaticky vygeneroval potrebný formulár, jeho logiku a ukladanie spracovaných údajov do databázy. Takto bol implementovaný MVC vzor. Pochopiteľne, bolo potrebné mierne upraviť ako View (čiže samotný formulár), tak súbor UserController (teda jeho logiku a ukladanie údajov do databázy).

Úpravy logiky sa týkali najmä odstránenia možnosti úpravy používateľského mena a hesla. Na prezentačnej vrstve (view) bolo potrebné urobiť zmeny súvisiace s implementáciou dizajnu. Zmena HTML kódu a vytvorenie štýlov pre použité elementy. Výsledok je možné vidieť na Obr.1.7. V UserController bolo umožnené ajaxové overovanie formulára ako aj na prezentačnej vrstve. Nakoniec sa definovali pravidlá pre overovanie v modeli User a metóde Rules().

**Úprava profilu**

|  |  |
|--|--|
| Meno a priezvisko  | <input style="width: 90%;" type="text" value="Michal Chylik"/>   |
| Osobné číslo   | <input style="width: 90%;" type="text" value="Mike22"/>  |
| Email  | <input style="width: 90%;" type="text" value="michal.chylik@stuba.sk"/>                                |
| Mobil  | <input style="width: 90%;" type="text" value="0911111111"/>  |
| Pohlavie   | <input checked="" type="radio"/> Muž <input type="radio"/> Žena  |
| Práca  | <input checked="" type="radio"/> Áno <input type="radio"/> Nie   |
| O mne  | <div style="border: 1px solid #ccc; padding: 5px; min-height: 50px;">         Random text       </div> |
| <div style="background-color: #e74c3c; color: white; padding: 5px 15px; display: inline-block; border-radius: 5px;">           ULOŽ PROFIL  </div> |  |

Obr.1.7 Formulár pre editáciu profilu

### 1.2.4. Testovanie

|                           |  |                                       |                                       |
|---------------------------|--|---------------------------------------|---------------------------------------|
| <b>Názov</b>              | Chybný email                                     | ID Testu                              | US4-01                                |
| <b>Rozhranie</b>          | Obrazovka editácia profilu                       | ID UC                                 | ID 1                                  |
| <b>Účel</b>               | Výpis hlášky po Ajaxovom overení chybného emailu |                                       |                                       |
| <b>Vstupné podmienky</b>  | Prihlásený používateľ, zadaný neplatný email     |                                       |                                       |
| <b>Výstupné podmienky</b> | Zobrazenie chybovej hlášky                       |                                       |                                       |
| <b>Krok</b>               | <b>Akcia</b>                                     | <b>Očakávaná akcia</b>                | <b>Skutočná akcia</b>                 |
| 1.                        | Vloženie neplatného emailu do poľa „email“       | Žiadna                                | Žiadna                                |
| 2.                        | Kliknutie kurzorom na miesto mimo poľa „email“   | Overenie a zobrazenie chybovej hlášky | Zobrazenie chybovej hlášky po overení |

Tab. 1.4 Editácia profilu Test US4-01



|                           |  |                           |  |
|---------------------------|--|---------------------------|--|
| <b>Názov</b>              | Platný email                                       | ID Testu                  | US4-02   |
| <b>Rozhranie</b>          | Obrazovka editácia profilu                         | ID UC                     | ID 1   |
| <b>Účel</b>               | Overenie zadania platnej emailovej adresy cez Ajax |                           |  |
| <b>Vstupné podmienky</b>  | Prihlásený používateľ, zadaný platný email         |                           |  |
| <b>Výstupné podmienky</b> | Nezobrazenie chybovej hlášky                       |                           |  |
| <b>Krok</b>               | <b>Akcia</b>                                       | <b>Očakávaná akcia</b>    | <b>Skutočná akcia</b>                              |
| 1.                        | Vloženie platného emailu do poľa „email“           | Žiadna                    | Žiadna   |
| 2.                        | Kliknutie kurzorom na miesto mimo poľa „email“     | Overenie emailovej adresy | Adresa bola overená, systém nevypísal žiadnu chybu |

Tab. 1.5 Editácia profilu Test US4-02

|                           |   |                           |  |
|---------------------------|---|---------------------------|--|
| <b>Názov</b>              | Nezadaný email  | ID Testu                  | US4-03   |
| <b>Rozhranie</b>          | Obrazovka editácia profilu  | ID UC                     | ID 1   |
| <b>Účel</b>               | Overiť reakciu na prázdne pole „email“                                      |                           |  |
| <b>Vstupné podmienky</b>  | Prihlásený používateľ, nezadaný email/vyprázdnenie poľa s emailovou adresou |                           |  |
| <b>Výstupné podmienky</b> | Nezobrazenie chybovej hlášky  |                           |  |
| <b>Krok</b>               | <b>Akcia</b>  | <b>Očakávaná akcia</b>    | <b>Skutočná akcia</b>                              |
| 1.                        | Zadanie platnej emailovej adresy  | Žiadna                    | Žiadna   |
| 2.                        | Kliknutie kurzorom na miesto mimo poľa „email“                              | Overenie emailovej adresy | Adresa bola overená, systém nevypísal žiadnu chybu |
| 3.                        | Vyprázdnenie poľa email   | Žiadna                    | Žiadna   |
| 4.                        | Kliknutie kurzorom na miesto mimo poľa „email“                              | Overenie emailovej adresy | Adresa bola overená, systém nevypísal žiadnu chybu |

Tab. 1.6 Editácia profilu Test US4-03

|                           |  |                                |  |
|---------------------------|--|--------------------------------|--|
| <b>Názov</b>              | Odoslanie formulára  | ID Testu                       | US4-05                                 |
| <b>Rozhranie</b>          | Obrazovka editácia profilu   | ID UC                          | ID 1                                   |
| <b>Účel</b>               | Overenie správneho zápisu zadaných údajov                            |                                |  |
| <b>Vstupné podmienky</b>  | Prihlásený používateľ, zadanie údajov do všetkých položiek formulára |                                |  |
| <b>Výstupné podmienky</b> | Zobrazenie profilu používateľa s upravenými údajmi                   |                                |  |
| <b>Krok</b>               | <b>Akcia</b>   | <b>Očakávaná akcia</b>         | <b>Skutočná akcia</b>                  |
| 1.                        | Zadanie všetkých údajov do formulára                                 | Žiadna                         | Žiadna                                 |
| 2.                        | Kliknutie na tlačidlo „Ulož profil“                                  | Zobrazenie profilu používateľa | Zobrazenie profilu s upravenými údajmi |

|  |  |                     |  |
|--|--|---------------------|--|
|  |  | s upravenými údajmi |  |
|--|--|---------------------|--|

Tab. 1.7 Editácia profilu Test US5-05

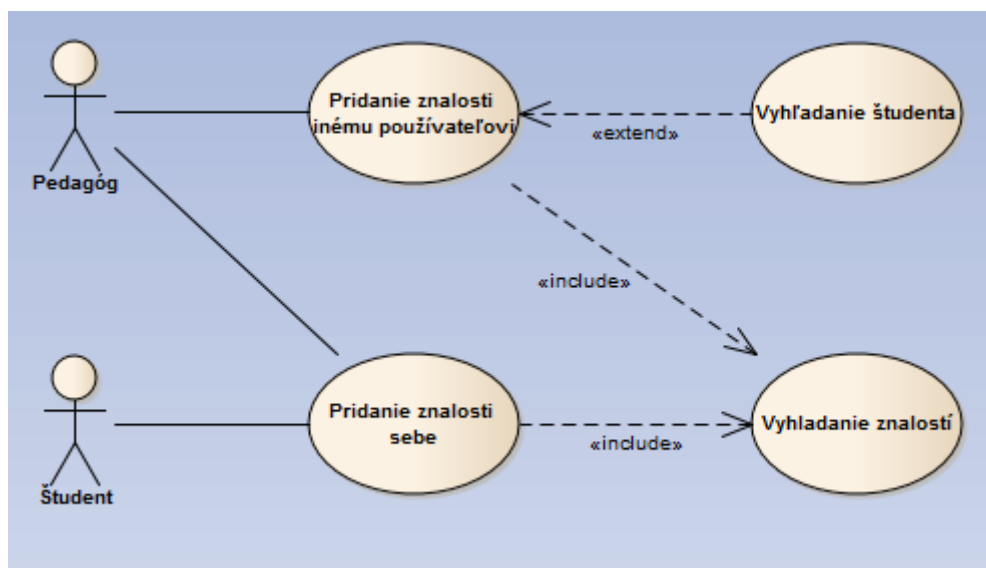
## 1.3. US 6 – Zadanie znalostí

Používateľ má umožnené pridávať do svojho profilu znalosti, alebo znalosti niekomu inému podľa oprávnení.

### 1.3.1. Analýza

Tento systém je zameraný na správu a zobrazovanie znalostí používateľov. Preto je nepochybne dôležitým krokom v tomto procese priradenie znalostí používateľovi.

Ide o jednoduchý formulár, ktorý umožní prehľadne pridávať znalosti sebe samému (obr. 1.8), alebo používateľovi, ktorému máme oprávnenie znalosti pridať. Na to aby sme mohli pridávať znalosti sebe alebo inému používateľovi, je nutné vyhľadať znalosti daného používateľa aby bolo zrejmé, ktoré znalosti už má pridané. Na to aby sme pridávali znalosti iným používateľom, ich nepotrebuje vyhľadať, môžeme sa dostať na obrazovku pridávania aj z profilu danej osoby.



Obr. 1.8 UseCase diagram US6

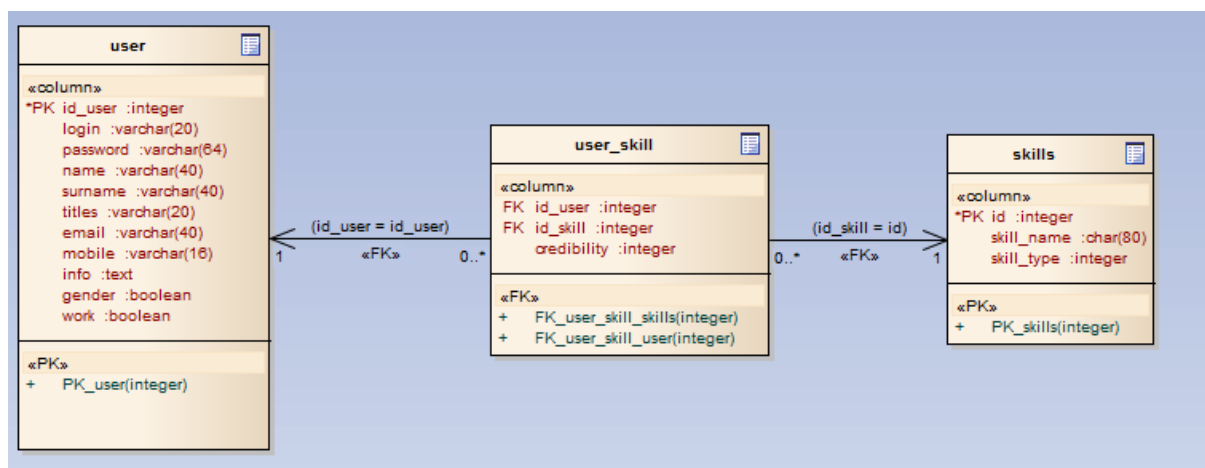
| Prípád použitia | Pridanie znalostí                            |
|-----------------|--|
| ID              | 2  |
| Stručný popis   | Používateľ si pridá znalosti pomocou výberu. |

|                             |   |
|-----------------------------|---|
| <b>Primárny aktéri</b>      | Prihlásený používateľ   |
| <b>Vedľajší aktéri</b>      | Žiadny  |
| <b>Vstupné podmienky</b>    | Používateľ je prihlásený  |
| <b>Hlavný scenár</b>        | <ol style="list-style-type: none"> <li>1. Prípád použitia začína po zobrazení formuláru na pridanie znalostí.</li> <li>2. Používateľ zvolí a pridá požadované znalosti do profilu.</li> <li>3. Používateľ potvrdí zmeny.</li> <li>4. Systém formulár spracuje a priradí používateľovi v databáze príslušné znalosti.</li> </ol> |
| <b>Výstupné podmienky</b>   | Žiadne  |
| <b>Alternatívne scenáre</b> | Žiadne  |

Tab. 1.8 Prípád použitia ID2

### 1.3.2.Návrh

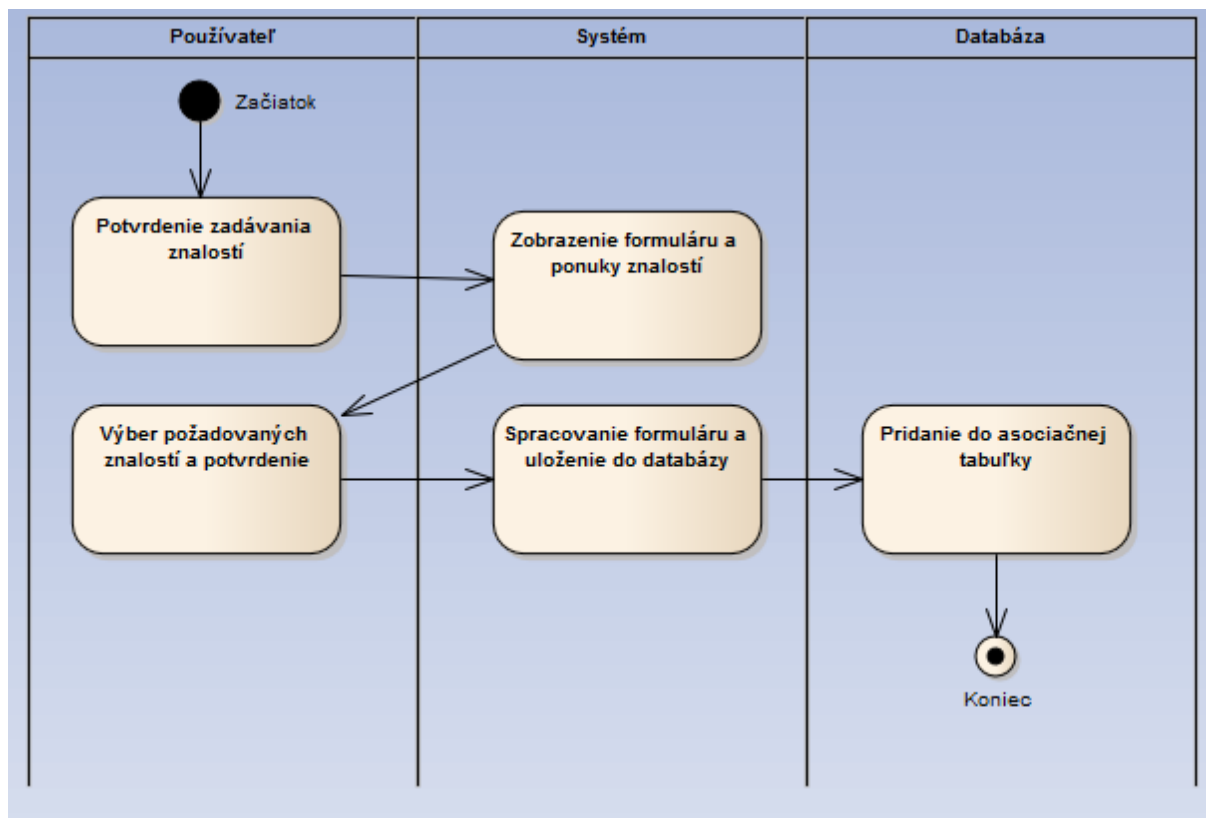
Funkcionalita je navrhnutá v databáze pomocou asociačnej tabuľky, ktorá obsahuje cudzie kľúče, a to ID používateľa a priradené znalosti z tabuľky znalostí (Obr.1.9).



Obr. 1.9 Fyzický model entity user knowledge

Aktivita diagram na obrázku 1.10 popisuje v slede aktivity, ktoré sa vykonávajú pri pridávaní znalostí. Ide o vyžiadanie formuláru na pridávanie znalostí a jeho zobrazenie. Následne používateľ vyberie množinu požadovaných znalostí ktorú chce pridať a odošle tieto zmeny. O ďalšie akcie už riadi

system. Príslušné znalosti pridá do asociačnej tabuľky v databáze ku zvolenému používateľovi:



Obr. 1.10 ActivityDiagram ZadanieZnalosti

### 1.3.3. Implementácia

Implementácia návrhu prebiehala pomocou frameworku Yii, kde sa po vytvorení tabuľky pre ukladanie vygeneroval CRUD generátorom formulár pre zadanie požadovaných znalostí do databázy. Tento formulár bol ešte upravený pre správne zobrazovanie a spôsob ukladania údajov do databázy.

### 1.3.4. Akceptačné testy

Pre testovanie bol navrhnutý scenár použitia pre zadanie znalostí používateľa.

|                           |                                     |                          |                       |
|---------------------------|-------------------------------------|--------------------------|-----------------------|
| <b>Názov</b>              | Pridanie znalostí                   | <b>ID Testu</b>          | US6-01                |
| <b>Rozhranie</b>          | Obrazovka formuláru                 | <b>ID UC</b>             | ID 2                  |
| <b>Účel</b>               | Pridanie nových znalostí do profilu |                          |                       |
| <b>Vstupné podmienky</b>  | Prihlásený používateľ               |                          |                       |
| <b>Výstupné podmienky</b> | Žiadne                              |                          |                       |
| <b>Krok</b>               | <b>Akcia</b>                        | <b>Očakávaná reakcia</b> | <b>Skutočná akcia</b> |

|    |   |  |  |
|----|---|--|--|
| 1. | Požiadavka na zobrazenie obrazovky s formulárom pridávania znalostí | Zobrazenie obrazovky                       | Zobrazenie obrazovky                       |
| 2. | Vybratie požadovaných znalostí a potvrdenie                         | Priradenie znalostí a uloženie do databázy | Priradenie znalostí a uloženie do databázy |

Tab. 1.9 Akceptačný test ZadanieZnalosti

## 1.4. US 11 - Vytvorenie prvotnej bázy pojmov

Ako používateľ chcem mať k dispozícii pojmy opisujúce znalosti, aby som s nimi mohol pracovať.

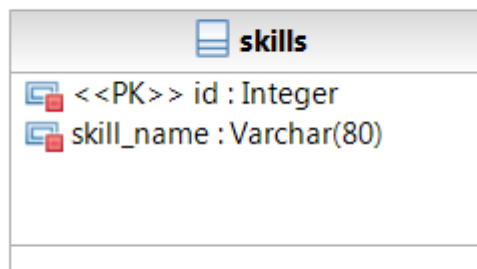
### 1.4.1. Analýza

Vytvorenie prvotnej bázy znalostí a zručností je základný kameň nášho projektu. Bez nej by sa nič iné nedalo spraviť. Z tejto prvotnej bázy pojmov sa bude vychádzať pri hierarchizácii zručností a znalostí študentov. Je dôležité oddeliť tieto znalosti od grúp, čo bude pole znalostí. Podľa logického modelu budú tri rôzne druhy grúp: práca, predmet, projekt.

### 1.4.2. Návrh

Ako databázu použijeme PostgreSQL.

Fyzický model entity vidíme na Obr. 1.11.



Obr. 1.11 Fyzický model entity skills

### 1.4.3. Implementácia

Ako GUI sa použil pgAdmin III. Tabuľka s prvotnou bázou pojmov sa vytvorila nasledovne:

```
-- Table: skills
CREATE TABLE skills
```

```
(  
  id serial,  
  skill_name character(80),  
  CONSTRAINT skills_id_key UNIQUE (id )  
)  
WITH (  
  OIDS=FALSE  
)  
);  
ALTER TABLE skills  
  OWNER TO postgres;
```

## 2. Druhý šprint – Prvá púnska vojna

---

V rámci druhého šprintu sme riešili nasledovné používateľské príbehy.

- US2 – Prihlasovanie používateľa
- US3 – Prezeranie profilu
- US7 – Zobrazenie vlastných znalostí
- US12 – Kategorizácia znalostí

### 2.1. US2 Prihlasovanie používateľa

---

Ako používateľ sa chcem prihlásiť do systému, aby som v ňom mohol pracovať.

#### 2.1.1. Analýza

Nakoľko väčšina systémov na škole používa pre prihlásenie do systému meno a heslo z AIS, tak sme sa aj my rozhodli pre tento spôsob prihlásenia. Tiež chceme aby systém bol jednoduchý pre používanie a teda prihlásenie bude používať javascript pre kontrolu vyplnených polí.

| Prípadoúžitia               | PrihlásenieDoSystému   |
|-----------------------------|--|
| <b>ID</b>                   | 3  |
| <b>Stručný popis</b>        | Používateľ sa pomocou prihlasovacieho formulára prihlasuje do systému. Používa pritom prihlasovacie meno a heslo do AIS.   |
| <b>Primárny aktéri</b>      | Neprihlásený používateľ  |
| <b>Vedľajší aktéri</b>      | Žiadny   |
| <b>Vstupné podmienky</b>    | Používateľ je neprihlásený   |
| <b>Hlavný scenár</b>        | <ol style="list-style-type: none"> <li>1. Prípadoúžitia začína pri zobrazení formuláru na prihlásenie.</li> <li>2. Používateľ zadá svoje prihlasovacie meno a heslo do AIS</li> <li>3. Používateľ odošle formulár.</li> <li>4. Systém spracuje formulár a prihlási používateľa.</li> </ol> |
| <b>Výstupné podmienky</b>   | Prihlásený používateľ.   |
| <b>Alternatívne scenáre</b> | <ol style="list-style-type: none"> <li>2a. Používateľ zadá zlé meno alebo heslo.</li> <li>2b. Systém po spracovaní formulára vyhlási chybu.</li> </ol>   |

## 2c. Systém umožní používateľovi opäť sa prihlásiť.

## Tab.2.1 Prípad použitia ID 3

## 2.1.2. Návrh

Na obrázku č. 2.1 je zobrazená obrazovka pre prihlásenie používateľa. Prihlasuje sa pomocou údajov z AIS, a preto je tu vynechaná možnosť zaregistrovať sa. Možnosť „zapamätať prihlásenie“ sa podľa uváženia všetkých bezpečnostných rizík zanechá alebo odstráni.



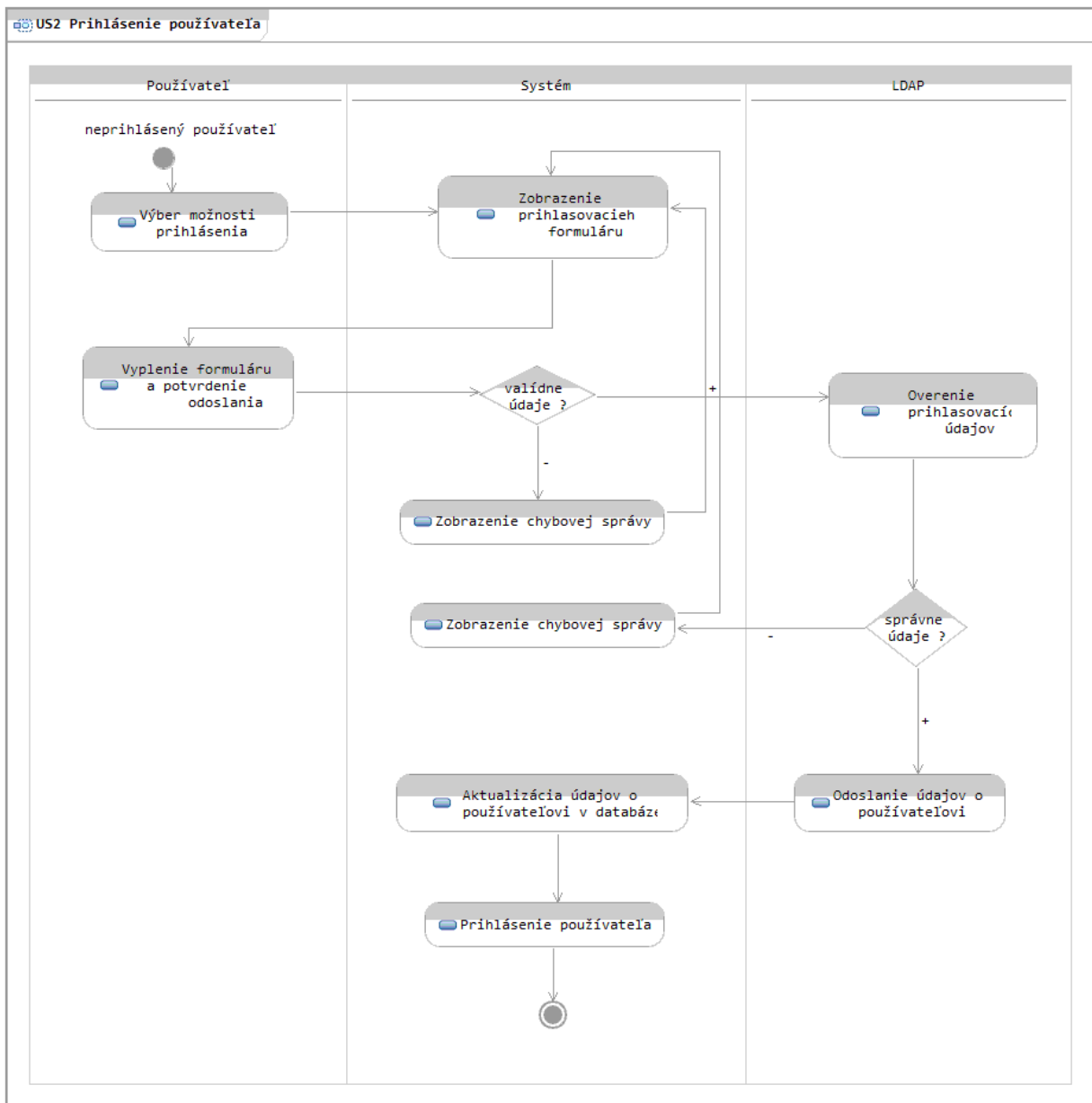
The image shows a login form with the following elements:

- A header bar with the text "Login".
- A label "Login" next to a text input field.
- A label "Heslo" next to a password input field.
- A checkbox labeled "zapamätaj prihlásenie" below the password field.
- A red button labeled "Prihlásiť" with a pencil icon to its right.

Obr. 2.1 Prihlásenie používateľa

Obrázok č. 2.2. nám znázorňuje diagram aktivít prípadu použitia, ktorý sa realizuje pri prihlásení používateľa. Znázorňuje aj alternatívne toky, ktoré sa vykonajú pri chybe. Neprihlásený používateľ má zobrazený formulár pre prihlásenie (Obr. 2.1). Pre prihlásenie použije meno a heslo ako do akademického informačného systému (AIS). Zadané údaje následne potvrdí odoslaním, kde sa na strane systému validujú, teda či boli vyplnené polia pre meno a heslo. Validné údaje sú odoslané pomocou protokolu LDAP, ktorý autentifikuje používateľa vzhľadom k AIS systému a navráti pre nás potrebné údaje ako meno a priezvisko, typ zamestnanca, md5 hash hesla, celé meno aj s titulmi, ktoré na našej strane vkladáme do databázy (ak sa používateľ prihlasuje prvý krát).





Obr. 2.2 : Diagram aktivít pre prihlásenie používateľa.

### 2.1.3.Implementácia

Najdôležitejšou súčasťou prihlásenia používateľa je overovanie údajov cez LDAP (Lightweight Directory Access Protocol). V nasledujúcom kóde je ukázané ako je toto overovanie realizované.

```
private function logonCheckLDAP()
{
    $basedn='ou=People,dc=stuba,dc=sk';
```

```
$ds = @ldap_connect('ldap://ldap.stuba.sk', 636);
//nastavovanie ldap
@ldap_set_option($ds, LDAP_OPT_PROTOCOL_VERSION, 3);
//tzv bind cize overenie mena hesla
$ldapBindResult = @ldap_bind($ds, 'uid='.$this->username.',ou=People,dc=stuba,dc=sk',
$this->password);
if (!$ldapBindResult)
{
    @ldap_close($ds);
    return false;
}
//filtrujeme co chceme dostat
$ldapFilter = array("uid", "userPassword", "employeetype", "uisid", "cn", "sn", "givenname");
//hladame
$ldapSearchResult = @ldap_search($ds, $basedn, 'uid='.$this->username, $ldapFilter);
if ($ldapSearchResult)
{
    $this->result = @ldap_get_entries($ds, $ldapSearchResult);
    @ldap_close($ds);
    return true;
}
else
{
    @ldap_close($ds);
    return false;
}
@ldap_close($ds);
return false;
}
```

Ak sú údaje dobré a overenie sa podarí tak do premennej **result** sa uloží výsledok funkcie `@ldap_search`, ktorá nám vráti meno, id, osobné číslo a typ zamestnanca priamo z AIS.

Ak sa používateľ prihlasuje do systému prvý krát (napr. študent prvého ročníka), systém to overí a automaticky v databáze uloží nového používateľa. V nasledujúcom kóde je možné vidieť ako to systém vykoná.

```
$connection=Yii::app()->db;

$command1 = $connection->createCommand("SELECT id_user FROM \"user\" WHERE login = :login
LIMIT 1");
$command1->bindParam(":login", $this->result[0]["uid"][0], PDO::PARAM_STR);

$user = $command1->query();

if($user->getRowCount() == 0)
{
    // ulozim noveho pouzivatelya
    $command2 = $connection->createCommand("INSERT INTO \"user\" (login, name)
VALUES (:login, :name)");
```

```

        $command2->bindParam(":login",    $this->result[0]["uid"][0], PDO::PARAM_STR);
        $command2->bindParam(":name",    $this->result[0]["cn"][0],
PDO::PARAM_STR);
$command2->execute();

        $user = $command1->query();

        $row = $user->read();
        $this->unique = array(

                "rid"    => $row['id_user'],

                "id"      => "sem AIS id",

                "perm" => "sem prava"

        );

        return true;
    }

```

Obe funkcie sa volajú z triedy UserIdentity, ktorá je komponentom, ktorá v sebe nesie informácie o používateľovi.

## 2.1.4. Testovanie

Nasledovné tri testovacie scenáre sme navrhli pre otestovanie prihlásenia používateľa.

|                           |   |   |   |
|---------------------------|---|---|---|
| <b>Názov</b>              | Úspešné prihlásenie do systému                        | <b>ID Testu</b>                                   | US2-01  |
| <b>Rozhranie</b>          | Obrazovka prihlásenia                                 | <b>ID UC</b>                                      | ID 3  |
| <b>Účel</b>               | Prihlásenie používateľa s prihlasovacími údajmi z AIS |   |   |
| <b>Vstupné podmienky</b>  | Správne prihlasovacie údaje                           |   |   |
| <b>Výstupné podmienky</b> | Žiadne  |   |   |
| <b>Krok</b>               | <b>Akcia</b>  | <b>Očakávaná reakcia</b>                          | <b>Skutočná akcia</b>                                     |
| 1.                        | Zadanie prihlasovacieho mena do príslušnej kolónky.   | Zobrazenie prihlasovacieho mena v kolónke "Login" | Prihlasovacie meno zobrazené vo formulárovom poli "Login" |

|    |  |  |   |
|----|--|--|---|
| 2. | Zadanie prihlasovacieho hesla do príslušnej kolónky. | Zobrazenie hesla v tvare hviezdíčiek v kolónke "Heslo" | Vyhviezdičkované heslo zobrazené vo formulárovom poli "Heslo" |
| 3. | Kliknutie na tlačidlo "Prihlásiť"                    | Prihlásenie používateľa do systému.                    | Prihlásený používateľ a zobrazenie úvodnej obrazovky.         |

Tab. 2.2 Prihlasovanie Test US2-01

|                           |   |   |   |        |
|---------------------------|---|---|---|--------|
| <b>Názov</b>              | Nevyplnenie poľa "Heslo"                                    |   | <b>ID Testu</b>   | US2-02 |
| <b>Rozhranie</b>          | Obrazovka prihlásenia                                       |   | <b>ID UC</b>  | ID 3   |
| <b>Účel</b>               | Zobrazenie chybovej hlášky: "Pole heslo nemôže byť prázdne" |   |   |        |
| <b>Vstupné podmienky</b>  | Žiadne  |   |   |        |
| <b>Výstupné podmienky</b> | Žiadne  |   |   |        |
| <b>Krok</b>               | <b>Akcia</b>  | <b>Očakávaná reakcia</b>                          | <b>Skutočná akcia</b>   |        |
| 1.                        | Zadanie prihlasovacieho mena do príslušnej kolónky.         | Zobrazenie prihlasovacieho mena v kolónke "Login" | Prihlasovacie meno zobrazené vo formulárovom poli "Login"                               |        |
| 2.                        | Kliknutie na tlačidlo "Prihlásiť"                           | Zobrazenie chybovej hlášky.                       | Zobrazená chybová hláška: "Pole heslo nemôže byť prázdne" pri formulárovom poli "Heslo" |        |

Tab. 2.3 Prihlasovanie Test2

|                           |  |  |  |        |
|---------------------------|--|--|--|--------|
| <b>Názov</b>              | Neúspešné prihlásenie do systému.                            |  | <b>ID Testu</b>  | US2-03 |
| <b>Rozhranie</b>          | Obrazovka prihlásenia  |  | <b>ID UC</b>   | ID 3   |
| <b>Účel</b>               | Zobrazenie chybovej hlášky: "Zadané zlé prihlasovacie údaje" |  |  |        |
| <b>Vstupné podmienky</b>  | Nesprávne prihlasovacie údaje                                |  |  |        |
| <b>Výstupné podmienky</b> | Žiadne   |  |  |        |
| <b>Krok</b>               | <b>Akcia</b>   | <b>Očakávaná reakcia</b>                               | <b>Skutočná akcia</b>  |        |
| 1.                        | Zadanie prihlasovacieho mena do príslušnej kolónky.          | Zobrazenie prihlasovacieho mena v kolónke "Login"      | Prihlasovacie meno zobrazené vo formulárovom poli "Login"                            |        |
| 2.                        | Zadanie prihlasovacieho hesla do príslušnej kolónky.         | Zobrazenie hesla v tvare hviezdíčiek v kolónke "Heslo" | Vyhviezdičkované heslo zobrazené vo formulárovom poli "Heslo"                        |        |
| 3.                        | Kliknutie na tlačidlo "Prihlásiť"                            | Vypísanie chybovej hlášky.                             | Zobrazená chybová hláška: "Zadané zlé prihlasovacie údaje" nad tlačidlom "Prihlásiť" |        |

Tab. 2.4 Prihlasovanie Test3

## 2.2. US3 Prezeranie profilu

Ako prihlásený používateľ systému chcem mať možnosť zobrazíť si vlastný profil, teda moje osobné údaje.

### 2.2.1. Analýza

Každý prihlásený používateľ musí mať možnosť prezrieť si vlastný profil. Ako profil rozumieme len údaje o osobe používateľa ako AIS meno, číslo, informácie o sebe a iné. Teda žiadne znalosti nebudú obsiahnuté v profile.

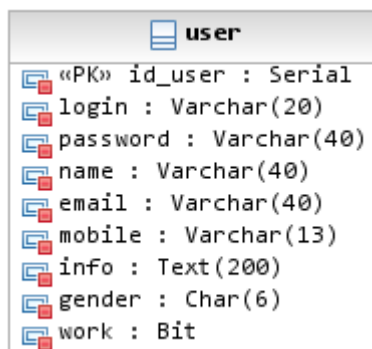
Samotný prípad použitia bude definovaný nasledovnými opisom:

| Prípad použitia             | Prezeranie profilu  |
|-----------------------------|---|
| <b>ID</b>                   | 4   |
| <b>Stručný popis</b>        | Používateľ si zobrazí svoje osobné údaje  |
| <b>Primárny aktéri</b>      | Prihlásený používateľ   |
| <b>Vedľajší aktéri</b>      | Žiadny  |
| <b>Vstupné podmienky</b>    | Používateľ je prihlásený  |
| <b>Hlavný scenár</b>        | <ol style="list-style-type: none"> <li>1. Používateľ vyberie možnosť zobrazenia vlastného profilu</li> <li>2. Systém získa osobné údaje používateľa</li> <li>3. Systém zobrazí osobné údaje používateľa</li> <li>4. Koniec</li> </ol> |
| <b>Výstupné podmienky</b>   | Žiadne  |
| <b>Alternatívne scenáre</b> | Žiadne  |

Tab.2.5 Prípad použitia ID 4

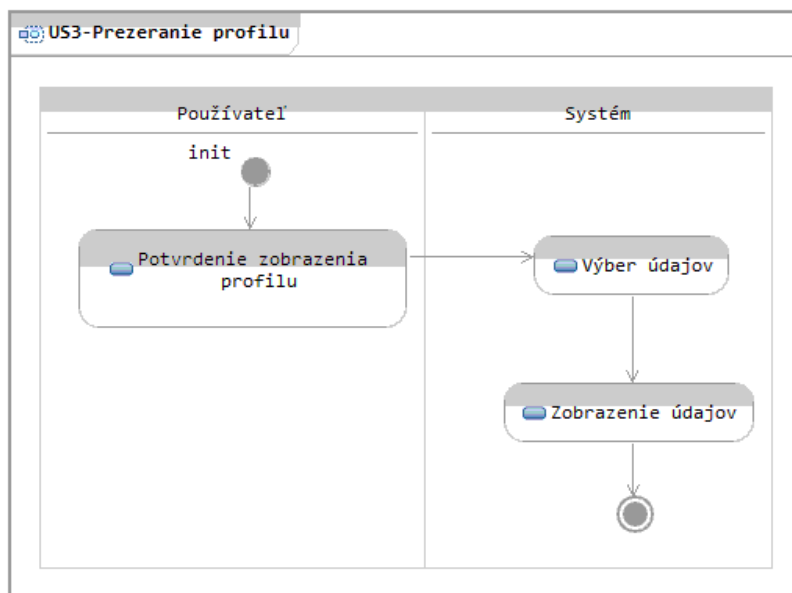
### 2.2.2. Návrh

Samotné údaje sú uložené v databáze ako entita User(Obr. 2.3), ktorej model je výsledkom US4 Editácia profilu.



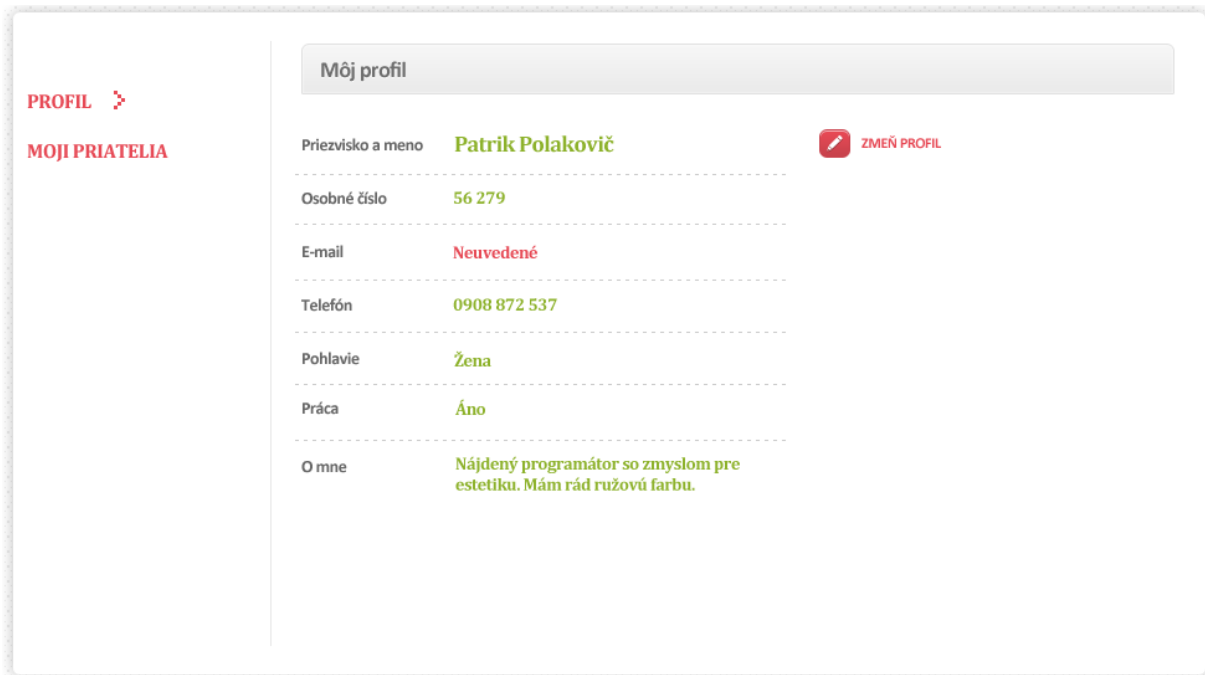
Obr. 2.3 Fyzický model entity User

Na Obr.2.4 vidíme diagram aktivít, ktorý znázorňuje prezeranie profilu používateľom.



Obr. 2.4 Activity diagram Prezeranie profilu

Ďalším krokom je návrh obrazovky (Obr. 2.5), v do ktorej sa vypíšu údaje o používateľovi.



Obr. 2.5 Návrh obrazovky Prezeranie profilu

### 2.2.3. Implementácia

Implementácia samotnej logiky bola jednoduchá, nakoľko sme pracovali s už existujúcimi modelmi pre databázovú entitu User. My sme len vybrali dáta z databázy a posunuli ich príslušnému zobrazeniu, v zmysle MVC. Funkcionalita sa nachádza v *SiteController.php*:

```
public function actionShowProfil()
{
    $conn = Yii::app()->db;

    $query = $conn->createCommand()
        ->select('login, name, email, mobile, info, gender, work')
        ->from('user')
        ->where('login="'.Yii::app()->user->getId().'")
        ->limit(1)
        ->query();

    if($query->getRowCount() != 1)
    {
        throw new CHttpException(404,'The specified content does not exist.');
```

## 2.2.4. Testovanie

---

Pre testovanie bol navrhnutý jediný scenár a to zobrazenie existujúcich údajov o používateľovi:

|                           |   |                          |                          |
|---------------------------|---|--------------------------|--------------------------|
| <b>Názov</b>              | Zobrazenie profilu                                | ID Testu                 | US3-01                   |
| <b>Rozhranie</b>          | Obrazovka prihláseného používateľa                | ID UC                    | ID 4                     |
| <b>Účel</b>               | zobrazenie vlastného profilu                      |                          |                          |
| <b>Vstupné podmienky</b>  | prihlásený používateľ                             |                          |                          |
| <b>Výstupné podmienky</b> | Žiadne  |                          |                          |
| <b>Krok</b>               | <b>Akcia</b>                                      | <b>Očakávaná akcia</b>   | <b>Skutočná akcia</b>    |
| 1.                        | V menu kliknem na tlačidlo pre zobrazenie profilu | Zobrazenie môjho profilu | Zobrazenie môjho profilu |

Tab. 2.6 Prezeranie profilu Test US3-01

## 2.3. US7 Zobrazenie vlastných znalostí

---

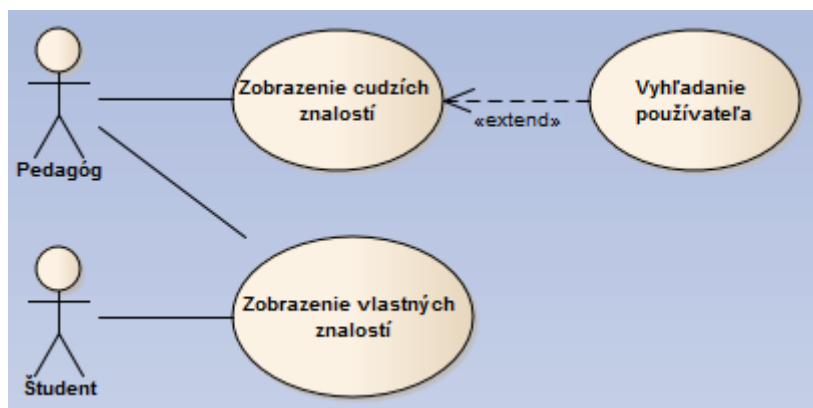
Používateľ má možnosť prezerat' všetky svoje pridané znalosti.

### 2.3.1. Analýza

Každý používateľ by mal mať možnosť prezerania informácií o sebe minimálne kvôli kontrole informácií. Preto je potrebné mu poskytnúť toto zobrazenie.

Zobrazí sa jednoduchý formulár, ktorý prehľadne zobrazí všetky používateľove pridané znalosti. Ako je znázornené na obrázku číslo 2.6, pedagóg má možnosť zobrazenia znalostí iných študentov. Je použitý vzťah *extend*, pretože pedagóg si môže zobrazit' znalosti používateľa aj iným spôsobom ako pomocou hľadania, napríklad sa na obrazovku zobrazenia znalostí môže dostať z obrazovky zobrazenia profilu príslušného študenta.





Obr. 2.6 UseCase diagram US7

| Prípád použitia      | Pridanie znalostí  |
|----------------------|--|
| ID                   | 5  |
| Stručný popis        | Používateľ si zobrazí vlastné alebo cudzie znalosti  |
| Primárny aktéri      | Prihlásený používateľ  |
| Vedľajší aktéri      | Žiadny   |
| Vstupné podmienky    | Používateľ je prihlásený   |
| Hlavný scenár        | <ol style="list-style-type: none"> <li>1. Prípád použitia začína po zvolení zobrazenia znalostí,</li> <li>2. Systém zobrazí používateľovi jeho znalosti.</li> </ol>  |
| Výstupné podmienky   | Žiadne   |
| Alternatívne scenáre | <ol style="list-style-type: none"> <li>1. Oprávnený používateľ si vyhľadá iného používateľa,</li> <li>2. Systém zobrazí znalosti vyhľadaného používateľa.</li> </ol> |

Tab. 2.7 UseCase scenár ID5

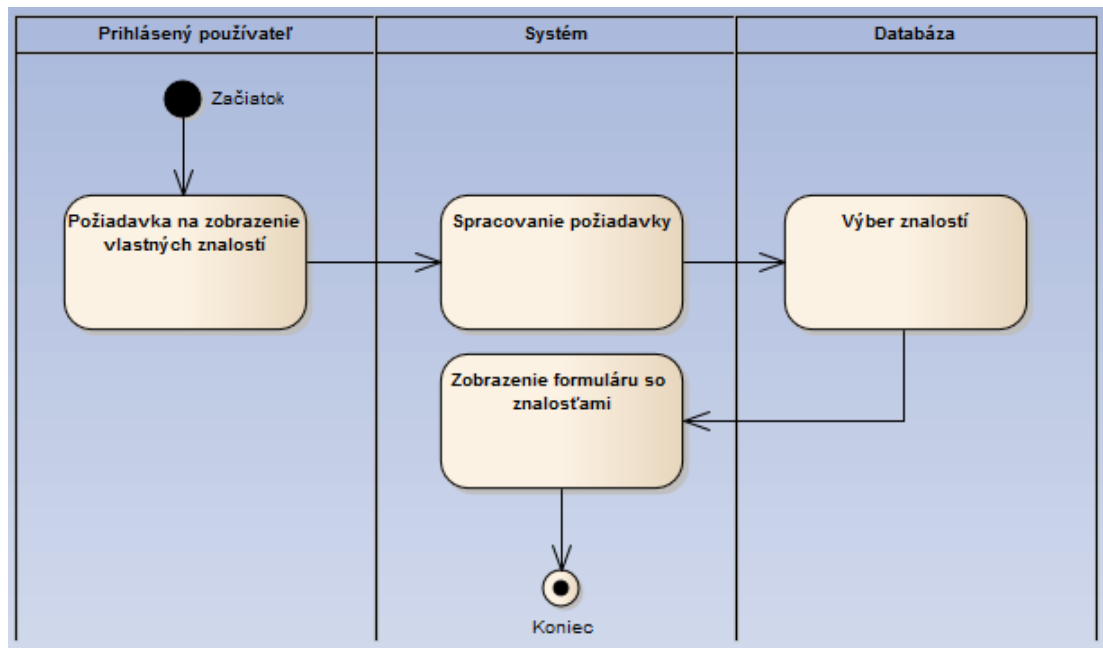
### 2.3.2.Návrh

Funkcionalita berie informácie z databázy pomocou asociačných tabuliek používateľa so znalosťou.

Cieľom je vypísať tie znalosti z tabuliek, ktoré sú priradené jednému konkrétnemu používateľovi, v tomto prípade prihlásenému používateľovi.

Diagram aktivít na obrázku 2.7 znázorňuje spôsob zobrazenia. Používateľ interaguje veľmi krátko, iba si zvolí zobraziť znalosti. Ostatné už riadi systém, ktorý spracuje potrebné údaje a vykoná načítanie

príslušných znalostí z databázy. Načítané znalosti spracuje do formuláru a zobrazí späť používateľovi.



Obr. 2.7. Activity diagram Zobrazenie vlastných znalostí

### 2.3.3. Implementácia

Implementácia návrhu pomocou frameworku Yii bola realizovaná pomocou vyhľadávania v databáze znalostí používateľa, a následným výpisom týchto znalostí

```

$ssql = "
SELECT
A.id_skill,
        B.name_skill
FROM
        user_skill as A
INNER JOIN
        skill as B
ON
        A.id_skill = B.id_skill
WHERE
        A.id_user = " . $id_user . "
";
$skills = $connection->createCommand($ssql)->query();
$ret = array(); $c = 0;
foreach($skills as $row){
$ret[$c]['id'] = $row['id_skill'];
    $ret[$c++]['name'] = $row['name_skill'];
}
  
```

### 2.3.4. Akceptačné testy

Pre testovanie bol navrhnutý jediný scenár použitia pre zobrazenie znalostí používateľa.

|                           |  |                                    |                                    |
|---------------------------|--|------------------------------------|------------------------------------|
| <b>Názov</b>              | Zobrazenie znalostí  | <b>ID Testu</b>                    | US7-01                             |
| <b>Rozhranie</b>          | Obrazovka formuláru  | <b>ID UC</b>                       | ID 5                               |
| <b>Účel</b>               | Zobrazenie vlastných znalostí používateľa                    |                                    |                                    |
| <b>Vstupné podmienky</b>  | Prihlásený používateľ  |                                    |                                    |
| <b>Výstupné podmienky</b> | Žiadne   |                                    |                                    |
| <b>Krok</b>               | <b>Akcia</b>   | <b>Očakávaná reakcia</b>           | <b>Skutočná akcia</b>              |
| 1.                        | Požiadavka na zobrazenie obrazovky so znalosťami používateľa | Zobrazenie obrazovky so znalosťami | Zobrazenie obrazovky so znalosťami |

Tab. 2.8 Zobrazenie vlastných znalostí test US7-01

## 2.4. US12 Kategorizácia znalostí

Ako používateľ chcem mať k dispozícii kategorizované pojmy v určitej hierarchii opisujúce znalosti, aby som s nimi mohol pracovať.

### 2.4.1. Analýza

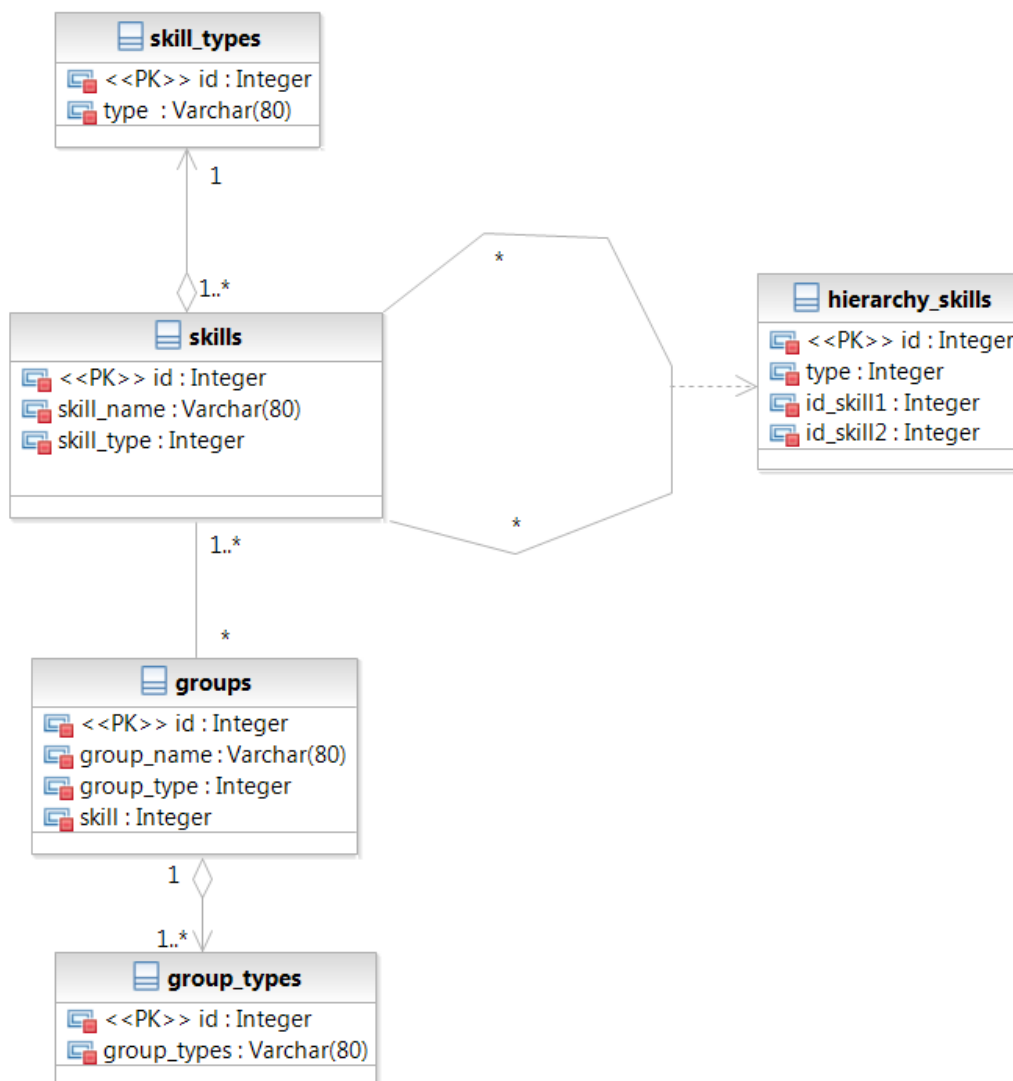
Keďže už máme k dispozícii prvotnú bázu pojmov, je nutné pre tieto pojmy spraviť hierarchiu. Táto hierarchia je dôležitá, aby sa vedelo, ktorá znalosť študenta je nadriadená ktorej. Takže ak máme pre pojem ".NET" priradený pojem "C#" znamená to, že pojem "C#" je podriadený. Zase si tieto znalosti netreba mýliť s grupami, ktoré podľa návrhu logického modelu predstavujú pole znalostí. V tomto prípade by grupa mohla byť "Programovanie". Grupy sa budú riešiť v nasledujúcich šprintoch.

### 2.4.2. Návrh

V databáze PostgreSQL pribudla tabuľka `hierarchy_skills`. Ako je vidieť z modelu, znalosť môže mať viac typov. Rozhodli sme sa tak preto, lebo sme našli niekoľko takých znalostí, napr.: PHPUnit môže byť typu `framework`, ale aj typu `test`. Navrhli sme prepájaciu tabuľku `hierarchy_skills`, ktorá nám poskytuje hierarchizáciu znalostí. Niektoré znalosti totiž môžu patriť pod iné znalosti ako napríklad `JavaBeans`, ktorá je súčasťou znalosti `J2EE`. Prvá znalosť predstavuje nadradenú znalosť (`id_skill1`) a druhá podradenú (`id_skill2`). Dôležitý je aj typ prepojenia, ktorý môže byť napríklad `isA(je)`, `precondition(predpoklad - na to aby som vedel C++ existuje predpoklad, že viem C)` alebo `isPartOf(je súčasťou, resp. patrí pod)`. Viacej znalostí môže byť prepojených na viacero znalostí.

Skupiny môžu byť rôzneho typu, ktoré sú definované v tabuľke `group_types`. Takýto typ môže byť napríklad práca, určitý predmet alebo projekt.

Skupina môže obsahovať viacero znalostí. Toto zaistí, že keď existuje istá skupina *Java tester*, ktorá je typu práca, tak sa od tejto skupiny vyžadujú znalosti *Java* ale aj *JUnit*.



Obr. 2.8 Fyzický model

### 2.4.3. Implementácia

Ako GUI sa použil pgAdmin III. Prepájacia tabuľka sa vytvorila nasledovne:

```

-- Table: hierarchy_skills
CREATE TABLE hierarchy_skills
(
  id serial,
  id_skill1 integer,
  id_skill2 integer,
  CONSTRAINT skills_id_key UNIQUE (id)
)
  
```

```
)  
WITH (  
  OIDS=FALSE  
);  
ALTER TABLE hierarchy_skills  
  OWNER TO postgres;
```

## 3. Tretí šprint – Druhá púnska vojna

V rámci tretieho šprintu sme riešili nasledovné používateľské príbehy.

- US5 – Zobrazenie kľúčových slov v systéme
- US8 – Zobrazenie znalostí študenta

### 3.1. US8 Zobrazenie znalostí študenta

Ako pedagóg chcem mať možnosť zobrazenia znalostí vybraného používateľa.

#### 3.1.1. Analýza

Systém je primárne vytváraný pre potreby vyhľadávania znalostí študentov, aby títo študenti mohli byť buď kontaktovaných príslušným pedagógom alebo zaradený do určitej pracovnej skupiny na základe samotných znalostí študenta.

| Prípád použitia             | Prezeranie profilu  |
|-----------------------------|---|
| <b>ID</b>                   | 6   |
| <b>Stručný popis</b>        | Používateľ môže vyhľadať študenta a zobraziť si jeho znalosti   |
| <b>Primárny aktéri</b>      | Prihlásený používateľ   |
| <b>Vedľajší aktéri</b>      | Žiadny  |
| <b>Vstupné podmienky</b>    | Používateľ je prihlásený  |
| <b>Hlavný scenár</b>        | <ol style="list-style-type: none"> <li>1. Používateľ vyberie možnosť zobrazenia vyhľadávania študentov</li> <li>2. Systém zobrazí formulár pre vyhľadávanie študentov</li> <li>3. Používateľ zadá meno používateľa, ktorého znalosti chce zobraziť a potvrdí vyhľadanie</li> <li>4. Systém zobrazí znalosti vybraného používateľa</li> <li>5. Koniec</li> </ol> |
| <b>Výstupné podmienky</b>   | Žiadne  |
| <b>Alternatívne scenáre</b> | <p>Scenár sa aktivuje v bode 3.</p> <ol style="list-style-type: none"> <li>4a. Kritérium vyhľadania spĺňa viacero študentov</li> <li>5a. Systém zobrazí zoznam študentov, ktorý spĺňajú kritérium vyhľadania</li> <li>6a. Používateľ vyberie konkrétneho študenta</li> </ol>  |

zo zobrazeného zoznamu  
7a. Tok pokračuje v bode 4. hlavného scenára

Tab. 3.1 UseCase scenár ID6

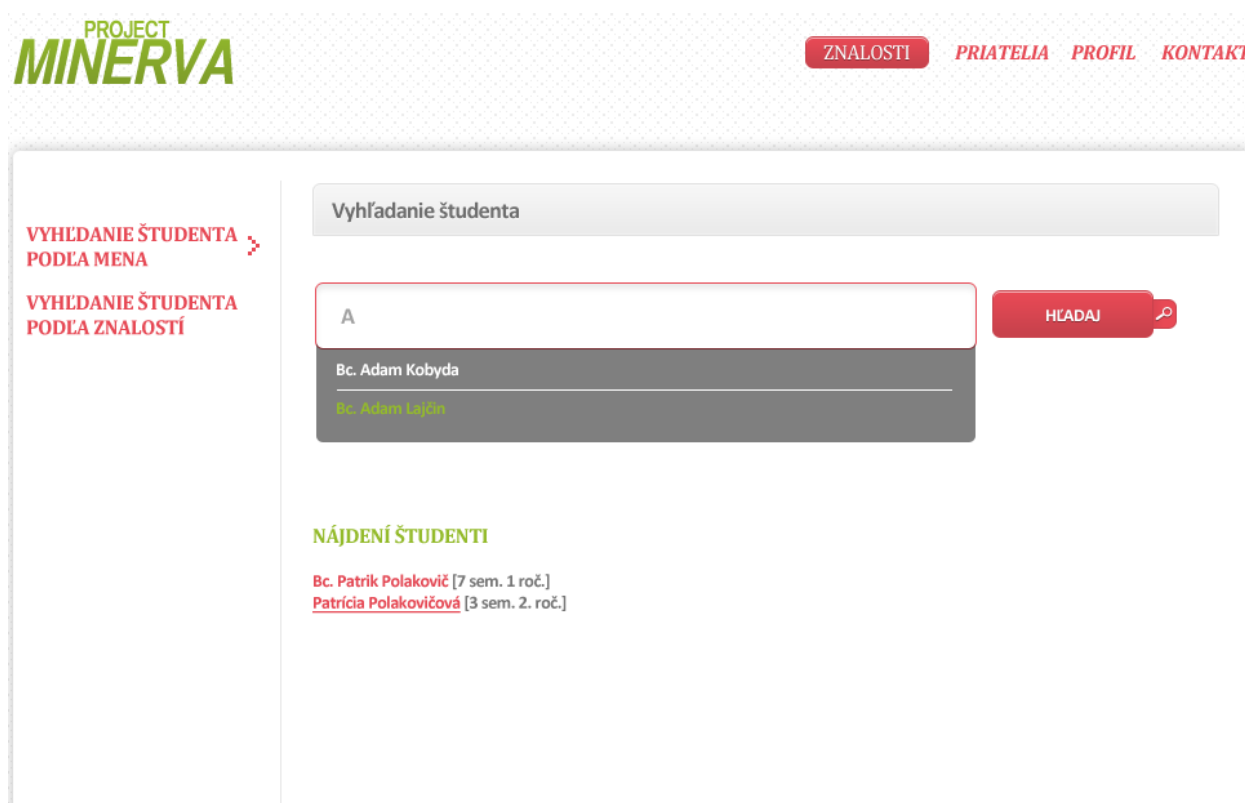
### 3.1.2.Návrh

Pre vytvorenie tohto prípadu použitia bolo treba pracovať na všetkých vrstvách MVC modelu ponúkaného Yii.

Pre návrh formuláru sme si vytvorili kritérium, aby bolo vyhľadávanie totožné s vyhľadávaním v systéme AIS. Teda pri zadávaní mena alebo priezviska sa automaticky zobrazí nápoved' s menami študentov spĺňajúci kritérium. Tento efekt sa dosahuje pomocou prvku autocomplete, ktorý sme aj použili.

Pre zobrazenie znalostí študenta je možné použiť existujúci formulár z prípadu použitia Zobrazenie vlastných znalostí.

Samotný grafický návrh vidieť na Obr. 3.1.



Obr.3.1 Návrh formulára pre vyhľadanie

### 3.1.3.Implementácia

Najzložitejšou časťou bola implementácia samotného autocomplete prvku. Tu bolo treba vytvoriť vyhľadanie podľa mena alebo podľa priezviska, čo sme dosiahli na úrovni databázy, ktorá poskytuje takúto funkcionality. Navyše bolo treba uvažovať vyhľadávanie *case-insensitive*, s signorovaním

diakritiky. Mená študentov, ktorý spĺňali dané kritérium vyhľadania odosielame klientovi pomocou ajax technológie, aby sme zabránili pomalému načítaniu stránky pri odosielaní všetkých mien naraz. Samotný kód bol implementovaný v *SiteController.php*, a k nemu patrí i príslušné zobrazenie *searchUsers.php*.

Najdôležitejšia časť kódu pre tento prípad použitia:

```
<?php
public function actionSearchUsers()
{
    $model=new PersonAutocomplete;

    // if it is ajax validation request
    if(isset($_POST['ajax']) && $_POST['ajax']==='search-form')
    {
        echo CActiveForm::validate($model);
        Yii::app()->end();
    }
    if(isset($_POST['PersonAutocomplete']))
    {
        $model->attributes=$_POST['PersonAutocomplete'];
        if($model->validate())
        {
            $res = $model->getPerson();
            // bolo najdenych viacero ludi
                if(is_array($res))
                {
                    $this->render('searchUser', array('model'=>$model, 'dataProvider' =>
$res));
                    Yii::app()->end();
                }
            // bol najdeny len 1 typu CActiveRecord
            else
            {
                $this->redirect('index.php?r=skill/showUserSkills&id='.$res-
>attributes['id_user']);
                Yii::app()->end();
            }
        }
    }

    $this->render('searchUser', array('model'=>$model, 'person' => Null));
}

public function actionAjaxPersonRequest()
{
    if(isset($_GET['term']))
    {
        $usrs = User::model()->findAll('(translate(name, \' Ľščťžýáíéúäö\',
\'łſctzyaieuao\') ilike
        translate(\'.$_GET[\'term\'].\'%\', \' Ľščťžýáíéúäö\', \'łſctzyaieuao\')) OR
        (translate(surname, \' Ľščťžýáíéúäö\', \'łſctzyaieuao\') ilike
        translate(\'.$_GET[\'term\'].\'%\', \' Ľščťžýáíéúäö\', \'łſctzyaieuao\'))
        ORDER BY surname LIMIT 50');

        $source = array();
        foreach($usrs as $row)
        {
```



```

    $source[] = array(
        'id' => $row->attributes['id_user'],
        'value' => $row->attributes['surname'].' '.$row->attributes['name'].'.',
        '$row->attributes['titles']
    );
}
echo CJSON::encode($source);
}
Yii::app()->end();
}
?>

```

Pre úplnú konformnosť s vyhľadávaním v systéme AIS boli implementované dve obrazovky. Resp. Obr.3.2 je hlavná obrazovka a obsahuje prvok vyhľadávania a jej podobrazovka (Obr.3.3), ktorá zobrazuje mená študentov spĺňajúcich vyhľadávacie kritérium.



Obr.3.2 Implementovaný formulár pre vyhľadávanie



### Obr.3.3 Implementované zobrazenie zoznamu všetkých študentov, spĺňajúcich kritérium vyhľadania

#### 3.1.4. Testovanie

Pre testovanie sme navrhli tri scenáre. Jeden pri ktorom sa za vyberie konkrétny používateľ ponúknutý autocomplete prvkom a druhý, kedy používateľ zadá čiastočné meno používateľa a zvolí tlačidlo. Vtedy by systém mal zobrazíť ponuku s viacerými študentmi. Tretí scenár by mal skončiť zobrazením chybovej hlášky, keď dám vyhľadať neexistujúceho študenta.

|                           |  |   |   |
|---------------------------|--|---|---|
| <b>Názov</b>              | Zobrazenie profilu   | ID Testu  | US12-01   |
| <b>Rozhranie</b>          | Obrazovka prihláseného používateľa                               | ID UC   | ID 6  |
| <b>Účel</b>               | zobrazenie znalosti konkrétneho študenta                         |   |   |
| <b>Vstupné podmienky</b>  | prihlásený používateľ  |   |   |
| <b>Výstupné podmienky</b> | Žiadne   |   |   |
| <b>Krok</b>               | <b>Akcia</b>   | <b>Očakávaná akcia</b>  | <b>Skutočná akcia</b>   |
| 1.                        | V menu kliknem na tlačidlo zobrazenia formuláru pre vyhľadávanie | Zobrazenie formuláru pre vyhľadávanie   | Zobrazenie formuláru pre vyhľadávanie   |
| 2.                        | Zadám priezvisko študenta, ktorý existuje v systéme              | Autocomplete mi ponúkne všetkých študentov, ktorých priezvisko začína na reťazec, ktorý som zadal | Autocomplete mi ponúkne všetkých študentov, ktorých priezvisko začína na reťazec, ktorý som zadal |
| 3.                        | Z ponuky autocomplete kliknem na meno konkrétneho študenta       | Vybrané meno sa zobrazí v edit okne autocomplete prvku  | Vybrané meno sa zobrazí v edit okne autocomplete prvku  |
| 4.                        | Kliknem na tlačidlo vyhľadania                                   | Systém zobrazí znalosti študenta  | Systém zobrazí znalosti študenta  |

Tab. 3.2 Zobrazenie znalostí študenta Test US12-01

|                           |  |   |   |
|---------------------------|--|---|---|
| <b>Názov</b>              | Zobrazenie profilu   | ID Testu  | US12-02   |
| <b>Rozhranie</b>          | Obrazovka prihláseného používateľa   | ID UC   | ID 6  |
| <b>Účel</b>               | výber zo zobrazenia viacerých používateľov, ktorý spĺňajú kritérium vyhľadávania |   |   |
| <b>Vstupné podmienky</b>  | prihlásený používateľ  |   |   |
| <b>Výstupné podmienky</b> | Žiadne   |   |   |
| <b>Krok</b>               | <b>Akcia</b>   | <b>Očakávaná akcia</b>                              | <b>Skutočná akcia</b>                               |
| 1.                        | V menu kliknem na tlačidlo zobrazenia formuláru pre vyhľadávanie                 | Zobrazenie formuláru pre vyhľadávanie               | Zobrazenie formuláru pre vyhľadávanie               |
| 2.                        | Ako kritérium vyhľadania zadám reťazec znakov, ktorý vyhovuje pre viacerých      | Autocomplete mi ponúkne všetkých študentov, ktorých | Autocomplete mi ponúkne všetkých študentov, ktorých |

|    |  |   |   |
|----|--|---|---|
|    | študentov  | meno alebo priezvisko začína na reťazec, ktorý som zadal              | meno alebo priezvisko začína na reťazec, ktorý som zadal, max. 50 študentov |
| 3. | Kliknem na tlačidlo vyhľadania                       | Systém zobrazí zoznam mien študentov spĺňajúcich kritérium vyhľadania | Systém zobrazí zoznam mien študentov spĺňajúcich kritérium vyhľadania,      |
| 4. | Vyberiem ľubovoľného študenta zo zobrazeného záznamu | Systém zobrazí znalosti študenta                                      | Systém zobrazí znalosti študenta  |

Tab. 3.3 Zobrazenie znalostí študenta Test US12-02

|                           |  |                                       |                                       |
|---------------------------|--|---------------------------------------|---------------------------------------|
| <b>Názov</b>              | Zobrazenie profilu   | ID Testu                              | US12-03                               |
| <b>Rozhranie</b>          | Obrazovka prihláseného používateľa                                     | ID UC                                 | ID 12                                 |
| <b>Účel</b>               | zobrazenie chybovej správy pri vyhľadávaní neexistujúcich používateľov |                                       |                                       |
| <b>Vstupné podmienky</b>  | prihlásený používateľ  |                                       |                                       |
| <b>Výstupné podmienky</b> | Žiadne   |                                       |                                       |
| <b>Krok</b>               | <b>Akcia</b>   | <b>Očakávaná akcia</b>                | <b>Skutočná akcia</b>                 |
| 1.                        | V menu kliknem na tlačidlo zobrazenia formuláru pre vyhľadávanie       | Zobrazenie formuláru pre vyhľadávanie | Zobrazenie formuláru pre vyhľadávanie |
| 2.                        | Zadám priezvisko študenta, ktorý neexistuje v systéme                  | Žiadna akcia systém                   | Žiadna akcia systém                   |
| 3.                        | Kliknem na tlačidlo vyhľadania   | Systém zobrazí chybovú správu         | Systém zobrazí chybovú správu         |

Tab. 3.4 Zobrazenie znalostí študenta Test US12-03

## 3.2. US5 Zobrazenie kľúčových slov v systéme

Užívateľ si bude môcť zobraziť zoznam všetkých znalostí v systéme.

### 3.2.1. Analýza

Je potrebné vytvoriť ucelený zoznam znalostí, ktoré si môže používateľ pridať do svojho profilu. Pretože zoznam kľúčových slov má pomôcť používateľovi nájsť tie znalosti, na ktoré si nespomenul bežným vyplňaním formulára, je vhodné aby bol takýto zoznam umiestnený priamo pri pridávaní znalostí. Zoznam by mal umožňovať nie len zobraziť jednotlivé znalosti, ale mal by priamo ponúkať možnosť si nájdenú znalosť pridať priamo cez zoznam.

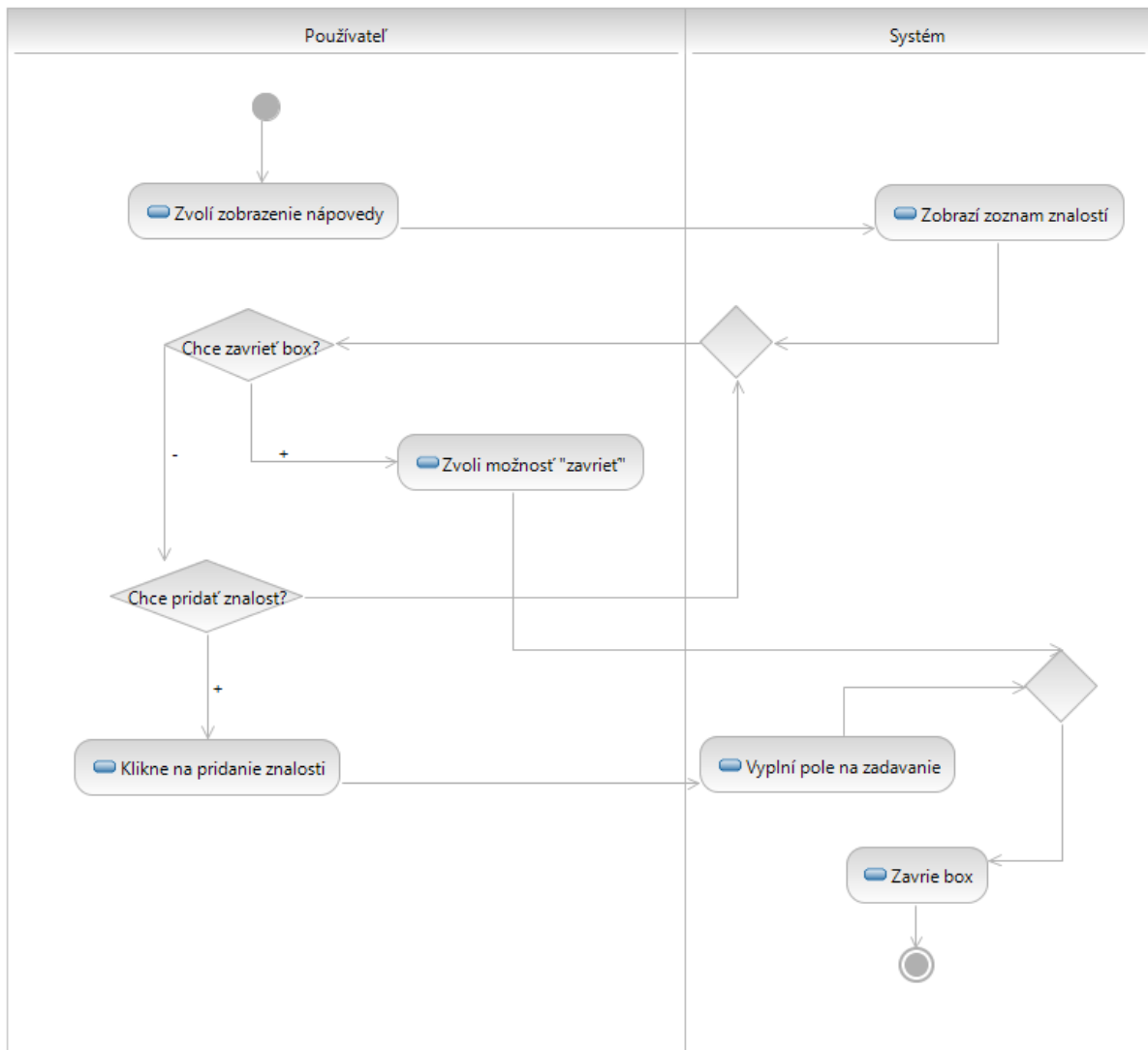
### 3.2.2.Návrh

Zoznam znalostí budeme reprezentovať pomocou dialógu. Ten sa vyvolá po stlačení tlačidla „nápoveda“ . Obr.3.4 znázorňuje prvotný grafický návrh zoznamu kľúčových slov, vychádzajúc z analýzy. Obsahuje štruktúrovaný výpis znalostí podľa prislúchajúcej kategórie. Znalosti sú ďalej rozlíšené, ak ich už používateľ má pridané alebo nie. Pre znalosti, ktoré nemá používateľ pridané sa zobrazí ikona na pridanie znalosti.



Obr.3.4 Návrh zoznamu kľúčových slov

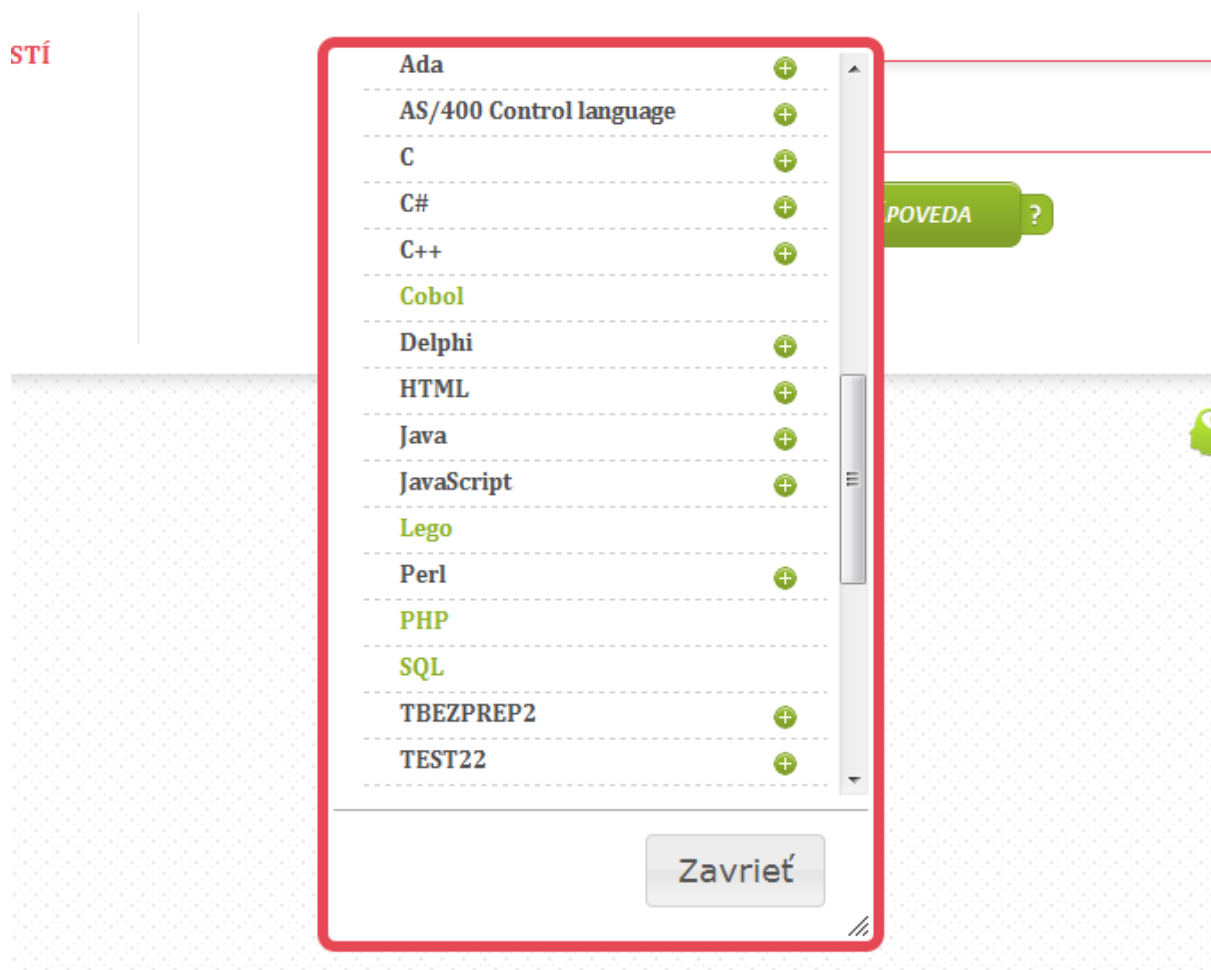
Obr.3.5 znázorňuje diagram aktivít pre prácu so zoznamom znalostí. Používateľ začína vyvolaním boxu po stlačení tlačidla. Počas obdobia, keď má otvorený zoznam, môže kedykoľvek zoznam zavrieť. Preto je umiestnení rozhodovací blok hneď na začiatku diagramu. Ak sa rozhodne zavrieť, klikne na príslušné tlačidlo a systém box zavrie. Ak používateľ neuzavrel box, tak pokračuje v práci a má na výber pridanie znalosti. Ak znalosť pridať nechce, znova sa dostáva na začiatok diagramu. Ak chce pridať znalosť, klikne na príslušnú ikonu, systém vykoná zmenu v poli pre zadávanie a uzavrie box.



Obr.3.5 Návrh zoznamu kľúčových slov

### 3.2.3. Implementácia

Nakoľko v dobe riešenia ešte nebola dokončená hierarchia znalostí v systéme, bol zoznam implementovaný bez kategórií. Vznikol tak jeden dlhý zoznam znalostí. Ostatná funkcionálna bola zachovaná. Pridávanie znalosti po stlačení ikony, automaticky vyplní pole na pridávanie znalosti jej názvom. Po odoslaní sa štandardne znalosť zapíše. V budúcnosti by sa malo dať pridávať viacej znalostí v jednom kroku, bez nutnosti znovunačítania stránok. V čase implementácie zoznamu kľúčových slov tento spôsob pridávania ešte nebol implementovaný. Výslednú podobu zoznamu je možné vidieť na obrázku XY.



Obr.3.6 Implementovaný zoznam kľúčových slov

Nasledujúci kód znázorňuje implementáciu jQuery dialog boxu, v ktorom sa zobrazuje zoznam znalostí.

```
<?php  
  
$this->beginWidget('zii.widgets.jui.CJuiDialog', array(  
    'id'=>'mydialog',  
    // additional javascript options for the dialog plugin  
    'options'=>array(  
        'autoOpen'=>false,  
        'buttons'=>array(  
            'Zavrieť'=>'js:function(){ $(this).dialog("close");}',
```

```

    ),
  ));
// tu je html obsah dialogu
$this->endWidget('zii.widgets.jui.CJuiDialog');
?>

```

Nasledujúci od zázorňuje implementáciu pridania znalosti po kliknutí na ikonu.

```

<?php
jQueryRows = "";
jQueryRows .= "
    jQuery('.plus_icon').click(function(){
        jQuery('#mydialog').dialog(\"close\");
        jQuery('#SkillAutocomplete_id_user').val($(this).attr('rel'));
        jQuery('#SkillAutocomplete_id_skill').val($(this).attr('alt'));
    });
";
Yii::app()->getClientScript()->registerScript("", $jQueryRows);
?>

```

### 3.2.4. Testovanie

Testovanie prebiehalo v jedinom scenári, ktorý overil zobrazenie a zároveň aj pridanie znalosti.

|                           |  |   |   |
|---------------------------|--|---|---|
| <b>Názov</b>              | Zobraz a pridaj znalosť v zozname              | ID Testu  | US5-01  |
| <b>Rozhranie</b>          | Obrazovka pridanie znalostí                    | ID UC   | ID 6  |
| <b>Účel</b>               | Overenie zoznamu kľúčových slov                |   |   |
| <b>Vstupné podmienky</b>  | Prihlásený používateľ                          |   |   |
| <b>Výstupné podmienky</b> | Zobrazenie chybovej hlášky                     |   |   |
| <b>Krok</b>               | <b>Akcia</b>                                   | <b>Očakávaná akcia</b>  | <b>Skutočná akcia</b>   |
| 1.                        | Stlačenie tlačítka „nápoveda“                  | Zobrazenie zoznamu kľúčových slov                             | Zobrazenie zoznamu kľúčových slov                             |
| 2.                        | Kliknutie na ikonu pridania znalosti v zozname | Zavretie zoznamu a vyplnenie názvu znalosti do vstupného poľa | Zavretie zoznamu a vyplnenie názvu znalosti do vstupného poľa |

Tab. 3.5 Editácia profilu Test US5-01

## 4. Štvrtý šprint – Storočná vojna

V rámci štvrtého šprintu sme riešili nasledovné používateľské príbehy.

- US9 – Vyhľadanie študenta
- US14 – Hierarchia znalostí + GUI pre zadávanie
- US16 – Nastavenie práv
- US18 – Nastavenie ratingu pri zadávaní znalosti
- CR4 – Použitie hierarchie znalostí v systéme

### 4.1. US9 Vyhľadanie študenta

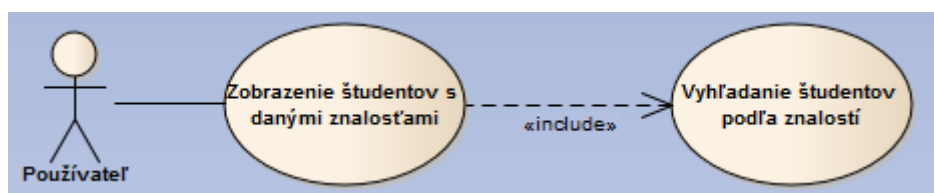
Ako používateľ mám záujem vyhľadať študentov, ktorí majú spoločné určité znalosti.

#### 4.1.1. Analýza

V niektorých prípadoch môže byť nápomocné vyhľadávanie používateľov podľa znalostí, ktoré majú zadané. Preto je možné zvoliť množinu znalostí a vyhľadať všetkých používateľov, ktorí ich majú.

Zobrazí sa vyhľadávací formulár, ktorý po výbere znalostí a potvrdení prehľadne zobrazí všetkých používateľov, ktorí vyhovujú požiadavkám vyhľadávania.

Na diagrame použitia, znázornenom na obrázku číslo 4.1, je jediná funkcia ktorú môže používateľ zavolať. Na to, aby si zobrazil používateľov s danými spoločnými znalosťami musí zadať znalosti do vyhľadávacieho formuláru, ktorý systém po odoslaní spracuje a poskytne zoznam hľadaných používateľov.



Obr 4.1 - UC Diagram: Vyhľadanie študentov podľa znalostí

| Prípád použitia | Vyhľadanie študentov podľa znalostí            |
|-----------------|--|
| ID              | 7  |
| Stručný popis   | Používateľ si vyhľadá študentov podľa znalostí |
| Primárny aktéri | Prihlásený používateľ                          |



|                             |   |
|-----------------------------|---|
| <b>Vedľajší aktéri</b>      | Žiadny  |
| <b>Vstupné podmienky</b>    | Používateľ je prihlásený  |
| <b>Hlavný scenár</b>        | Prípado použitia začína výberom množiny znalostí pre vyhľadávanie, System zobrazí používateľovi všetkých študentov s danými znalosťami. |
| <b>Výstupné podmienky</b>   | Žiadne  |
| <b>Alternatívne scenáre</b> | Žiadne  |

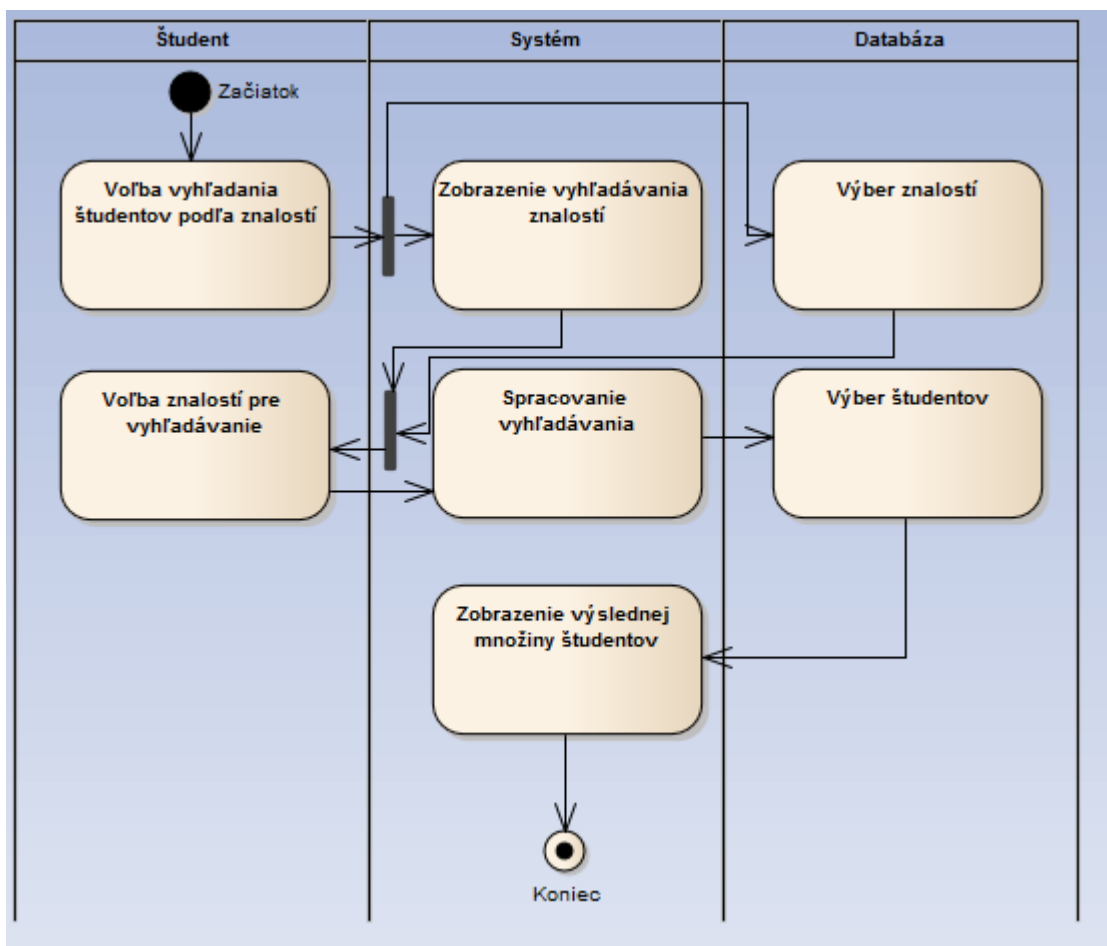
Tab. 4.1 UseCase scenár ID7

### 4.1.2. Návrh

Táto funkcionálna vyhľadáva študentov pomocou asociačnej tabuľky `user_skill`, zobrazenej na obrázku č. 4.2, ktorá uchováva vzťah používateľ – znalosť.

Výsledkom je zobrazenie tých používateľov z tabuľky, ktorí majú všetky hľadané znalosti.

Diagram aktivít znázornený na obrázku číslo 4.2 popisuje spôsob, akým sa vyhľadávajú používatelia podľa znalostí. Používateľ si zvolí zobrazenie vyhľadávacieho formuláru. Na to systém potrebuje poskytnúť formulár a súčasne v ňom mať načítané všetky dostupné znalosti z databázy, preto je tento krok znázornený súčasne. Ďalej je postup už priamočiary. Používateľ si vyberie znalosti podľa ktorých chce vyhľadávať, odošle požiadavku. Systém spracuje hľadané znalosti, vyhľadá v databáze študentov s danými znalosťami a zobrazí ich v zozname späť používateľovi.



Obr. 4.2. Activity diagram: Zobrazenie vlastných znalostí

### 4.1.3. Implementácia

Implementácia vo frameworku Yii bola vykonaná vyhľadávaním z asociačnej tabuľky user\_skill v databáze a následným výpisom študentov s danými znalosťami.

```

$mysql = '
SELECT
A.id_user,
A.id_skill,
B.name,
B.surname
FROM
user_skill as A
INNER JOIN
"user" as B
ON
A.id_user = B.id_user';

//filtrovanie podľa znalostí
for ($i=0; $i<sizeof($_FUN_ID_SKILL); ++$i) {
if ($i==0) {
    $mysql .= ' WHERE ';
} else {

```

```

    $sql .= ' OR ' ;
  } // end if..else..
  $sql .= 'A.id_skill = ' . $_FUN_ID_SKILL[$i] . ' ';
} // end for..

```

#### 4.1.4. Akceptačné testy

Pre testovanie bol navrhnutý scenár použitia pre vyhľadanie používateľov podľa znalostí.

|                           |  |   |   |
|---------------------------|--|---|---|
| <b>Názov</b>              | Vyhľadanie používateľov podľa znalostí                 | <b>ID</b>   | US9-01  |
| <b>Rozhr</b>              | Obrazovka vyhľadávania                                 | <b>ID UC</b>  | 7   |
| <b>Účel</b>               | Zobrazenie používateľov s danými spoločnými znalosťami |   |   |
| <b>Vstupné podmienky</b>  | Prihlásený používateľ                                  |   |   |
| <b>Výstupné podmienky</b> | Žiadne   |   |   |
| <b>Kro</b>                | <b>Akcia</b>   | <b>Očakávaná reakcia</b>                            | <b>Skutočná akcia</b>                               |
| 1.                        | Požiadavka na zobrazenie vyhľadávacej                  | Zobrazenie obrazovky so znalosťami                  | Zobrazenie obrazovky so znalosťami                  |
| 2.                        | Výber znalostí a potvrdenie vyhľadávania.              | Zobrazenie študentov spĺňajúcich kritéria hľadania. | Zobrazenie študentov s danými hľadanými znalosťami. |

Tab. 3.2 Akceptačný test UC7

## 4.2. US14 Pridanie novej znalosti

Používateľ bude mať k možnosť pridať novú znalosť do databázy cez používateľské rozhranie.

### 4.2.1. Analýza

Ak sa daná znalosť nenachádza v systéme, používateľ by mal mať možnosť ju pridať. Zobrazí sa formulár, kde používateľ bude mať možnosť pridať novú znalosť aj s ďalšími údajmi o znalosti ako je typ znalosti, prepojenie na inú znalosť a typ prepojenia na inú znalosť.

Na diagrame použitia, znázornenom na obrázku Obr.4.3 je jediná funkcia, ktorú používateľ môže zavolať.



Obr.4.3 UC Diagram: Pridanie novej znalosti

| Prípád použitia             | Pridanie novej znalosti   |
|-----------------------------|---|
| <b>ID</b>                   | 8   |
| <b>Stručný popis</b>        | Používateľ pridá novú znalosť do systému  |
| <b>Primárny aktéri</b>      | Prihlásený používateľ   |
| <b>Vedľajší aktéri</b>      | Žiadny  |
| <b>Vstupné podmienky</b>    | Používateľ je prihlásený  |
| <b>Hlavný scenár</b>        | <ol style="list-style-type: none"> <li>1. Používateľ vyberie možnosť pridania novej znalosti</li> <li>2. Systém zobrazí formulár pre pridanie novej znalosti</li> <li>3. Používateľ zadá novú znalosť a typ danej znalosti</li> <li>4. Systém pridá novú znalosť aj s typom do databázy</li> <li>5. Koniec</li> </ol> |
| <b>Výstupné podmienky</b>   | Žiadne  |
| <b>Alternatívne scenáre</b> | <p>Scenár sa aktivuje v bode 2.</p> <ol style="list-style-type: none"> <li>3a. Používateľ okrem zadania novej znalosti a jej typu, zadá aj znalosť, na ktorú je nová znalosť prepojená a typ daného prepojenia.</li> <li>4a Systém pridá všetky údaje do databázy</li> </ol>  |

Tab. 4.2 UseCase scenár ID7

### 4.2.2.Návrh

Pri návrhu formulára sa bralo do úvahy, že znalosť môže mať viacero typov. Preto sa pri dropdown boxe nachádza tlačidlo, ktoré pridá ďalší dropdown box pre zadanie typu znalosti.

Znalosť môže ale nemusí byť prepojená na inú znalosť daným typom ich prepojenia. Používateľ nemusí vybrať tieto možnosti (znalosť, na ktorú sa nová znalosť prepojí a typ prepojenia), ale ak si ich vyberie tak sa samozrejme aj tieto údaje odošlú a uložia do databázy. Takisto aj tu sa nachádza tlačidlo, ktoré pridá do formulára nové drop down boxy pre prepojenie na inú znalosť a typ prepojenia.

Grafický návrh celého formulára je vidieť na Obr.4.4.

PPRIDANIE ZNALOSTÍ &gt;

ZOBRAZENIE ZNALOSTÍ

### Pridanie znalostí

Znalosť  \*

Typ  +

Prepojená na  +

Typ prepojenia  +

**PRIDAJ ZNALOSŤ** +

\* - povinné polia

Obr. 4.4. Návrh formulára pre pridanie novej znalosti

### 4.2.3. Implementácia

Najdôležitejšou časťou bolo spravenie funkcie `actionCreateSkill()` v kontroleri `SkillController.php`. Táto funkcia obsahuje a volá dotazy pre prácu s databázou, a to, tri inserty do troch rôznych tabuliek a 1 select na výber id pre novú zadávanú znalosť:

```
// pridavanie znalosti do systemu
public function actionCreateSkill() {
    if((isset($ POST['newSkill']) && (isset($ POST['skill type']))) {
        $newSkill = $_POST['newSkill'];
        $arrayOfSkillType = $_POST['skill_type'];

        $connection = Yii::app()->db;

        //pridanie novej znalosti do unique skills
        $sql = $connection->createCommand("INSERT INTO unique_skills(skill_name)
VALUES ('$newSkill');");
        $sql->execute();

        // zistime si id novej znalosti
        $sql2 = "SELECT id FROM unique_skills WHERE skill_name='$newSkill';";
        $command=$connection->createCommand($sql2);
        $row=$command->queryRow(); // query and return the first row of result
        $idOfNewSkill = $row['id'];

        //pridame novu znalost do skills
        $sql3_str = "INSERT INTO skills(skill_type,unique_skill_id)";
        $value_sql3_str="VALUES";
        foreach($arrayOfSkillType as $k2 => $skillType){
            $value sql3 str = $value sql3 str . "(";
            $value sql3 str = $value sql3 str . "$skillType" . ",";
            $value sql3 str = $value sql3 str . "$idOfNewSkill";
            $value_sql3_str = $value_sql3_str . "),";
        }
        $value sql3 str = substr($value sql3 str, 0, -1) . ";";
        $sql3_str = $sql3_str . $value sql3 str;

        $sql3 = $connection->createCommand($sql3_str);
        $sql3->execute();

        if((isset($ POST['join skill']) && (isset($ POST['join skill type']))) {
            $arrayOfJoinSkill = $ POST['join skill'];

            $bool = False;
            foreach($arrayOfJoinSkill as $k => $joinSkill){
                if($joinSkill==""){ $bool = True;}
            }

            //podmienka ktora vrati 1, ak nie su vsetky connection skilly a typy
```

```

prepojeni vyplnene
        if($bool != 1){
            //$arrayOfJoinSkill = $_POST['join_skill'];
            $arrayOfJoinSkillType = $_POST['join_skill_type'];
            //pridame prepojenost novej znalosti s inou uz existujucou
            $sql4 str = "INSERT INTO hierarchy_skills(id skill1, id skill2,
            connection type)";

            $value_sql4_str="VALUES";
            foreach($arrayOfJoinSkill as $k => $joinSkill){
                $value_sql4_str = $value_sql4_str . "(";
                $value_sql4_str = $value_sql4_str . "'$idOfNewSkill'" .
                ",";
                $value_sql4_str = $value_sql4_str . "'$joinSkill'" .
                ",";
                $value_sql4_str = $value_sql4_str .
                "'$arrayOfJoinSkillType[$k]'" .
                ",";
                $value_sql4_str = $value_sql4_str . ")";
            }
            $value_sql4_str = substr($value_sql4_str, 0, -1) . ";";
            $sql4_str = $sql4_str . $value_sql4_str;

            $sql4 = $connection->createCommand($sql4 str);
            $sql4->execute();
        }
    }
}

$this->render('/skill/createSkill');
}

```

Ďalej sa do SkillController.php pridala jednoduchá funkcia getConnectionTypes(), ktorá vráti pole s typmi prepojení znalostí:

```

/**
 * metoda vrati pole so typmi prepojenia
 */
public function getConnectionTypes() {
    $connection = Yii::app()->db;

    $sql = "
        SELECT
            id,
            connection type
        FROM
            connection_types
    ";

    $connection types = $connection->createCommand($sql)
        ->query();

    $ret = array();
    $c = 0;
    foreach ($connection types as $row) {
        $ret[$row['id']] = $row['connection_type'];
    } // end foreach..

    return $ret;
} // end public function getConnectionTypes

```

#### 4.2.4. Testovanie

Pre testovanie boli navrhnuté 2 scenáre:

1. Pridanie znalosti, ktorá má viacero typov, bez akéhokoľvek prepojenia na inú znalosť.
2. Pridanie znalosti, ktorá má viacero prepojení na iné znalosti.

|                           |   |  |  |
|---------------------------|---|--|--|
| <b>Názov</b>              | Pridanie novej znalosti   | ID Testu                                   | US9-01   |
| <b>Rozhranie</b>          | Obrazovka prihláseného používateľa                                    | ID UC                                      | ID 8   |
| <b>Účel</b>               | Pridanie novej znalosti do systému                                    |  |  |
| <b>Vstupné podmienky</b>  | Prihlásený používateľ   |  |  |
| <b>Výstupné podmienky</b> | Žiadne  |  |  |
| <b>Krok</b>               | <b>Akcia</b>  | <b>Očakávaná akcia</b>                     | <b>Skutočná akcia</b>                              |
| 1.                        | V menu kliknem na tlačidlo zobrazenia formuláru pre pridanie znalosti | Zobrazenie formuláru pre pridanie znalosti | Zobrazenie formuláru pre pridanie znalosti         |
| 2.                        | Zadáam len novú znalosť a jej typ                                     | V textfieldu Znalosť bude napísaná znalosť | V textfieldu Znalosť je znalosť ktorú sme napísali |
| 3.                        | Kliknem na tlačidlo pre pridanie nového dropdown boxu pre typ         | Zobrazí sa nový dropdown box               | Zobrazebý nový dropdown box                        |
| 4.                        | Kliknem na tlačidlo pridania znalosti                                 | Do databázy sa uložia údaje                | Do databázy bola uložená nová znalosť a aj jej typ |

Tab.4.3 Pridanie novej znalosti s prepojením na inú znalosť

|                           |   |  |  |
|---------------------------|---|--|--|
| <b>Názov</b>              | Pridanie novej znalosti   | ID Testu   | US09-02  |
| <b>Rozhranie</b>          | Obrazovka prihláseného používateľa  | ID UC  | ID 8   |
| <b>Účel</b>               | Pridanie novej znalosti do systému aj s prepojeniami na iné znalosti              |  |  |
| <b>Vstupné podmienky</b>  | Prihlásený používateľ   |  |  |
| <b>Výstupné podmienky</b> | Žiadne  |  |  |
| <b>Krok</b>               | <b>Akcia</b>  | <b>Očakávaná akcia</b>                                   | <b>Skutočná akcia</b>                                  |
| 1.                        | V menu kliknem na tlačidlo zobrazenia formuláru pre pridanie znalosti             | Zobrazenie formuláru pre pridanie znalosti               | Zobrazenie formuláru pre pridanie znalosti             |
| 2.                        | Zadáam novú znalosť, jej typ a prepojenie na inú znalosť spolu s typom prepojenia | Údaje, ktoré vyplním budú v textfielde a dropdown boxoch | Údaje, ktoré vyplním sú v textfielde a dropdown boxoch |
| 3.                        | Kliknem na tlačidlo pre pridanie nových dropdown                                  | Pridajú sa nové dropdown boxy                            | Pridali sa nové dropdown boxy                          |

|    |   |                             |   |
|----|---|-----------------------------|---|
|    | boxov pre zobrazenie znalostí ,na ktoré ma byť nová znalosť prepojená a typ prepojenia na znalosť |                             |   |
| 4. | Kliknem na tlačidlo pridania znalosti   | Do databázy sa uložia údaje | Do databázy bola uložená nová znalosť, jej typ, a prepojenie na inú znalosť spolu s typom daného prepojenia |

Tab.4.4 Pridanie novej znalosti s prepojením na inú znalosť

### 4.3. US18 Nastavenie ratingu pri zadávaní znalosti

Pri zadávaní znalosti musí zadávateľ určiť na akej úrovni študent ovláda túto znalosť.

#### 4.3.1. Analýza

Študent môže mať priradený veľký počet znalostí. Pre učiteľa alebo externého návštevníka systému je však podstatné na akej úrovni študent ovláda jednotlivé znalosti. Je veľkým rozdielom či sa študent s daným pojmom len stretol alebo s ním má skúsenosti z praxe. Preto je veľmi potrebné priradiť k vzťahu študent-znalosť aj ohodnotenie, resp. úroveň študentových vedomostí a skúseností pre túto znalosť.

| Prípád použitia          | Pridanie znalostí s nastavením ratingu   |
|--------------------------|--|
| <b>ID</b>                | 9  |
| <b>Stručný popis</b>     | Používateľ si pridá znalosti pomocou výberu a priradí každej zo znalostí aj ohodnotenie úrovne ovládania znalosti.   |
| <b>Primárny aktéri</b>   | Prihlásený používateľ  |
| <b>Vedľajší aktéri</b>   | Žiadny   |
| <b>Vstupné podmienky</b> | Používateľ je prihlásený   |
| <b>Hlavný scenár</b>     | <ol style="list-style-type: none"> <li>1. Prípád použitia začína po zobrazení formuláru na pridanie znalostí.</li> <li>2. Používateľ zvolí a pridá požadované znalosti do profilu.</li> <li>3. Používateľ pri každej znalosti vyberie úroveň ovládania znalosti z ponúkaného výberu.</li> <li>4. Používateľ potvrdí zmeny.</li> <li>5. Systém formulár spracuje a priradí používateľovi v databáze príslušné znalosti spolu s príslušným ohodnotením.</li> </ol> |



**Výstupné podmienky** Žiadne

**Alternatívne scenáre** Žiadne

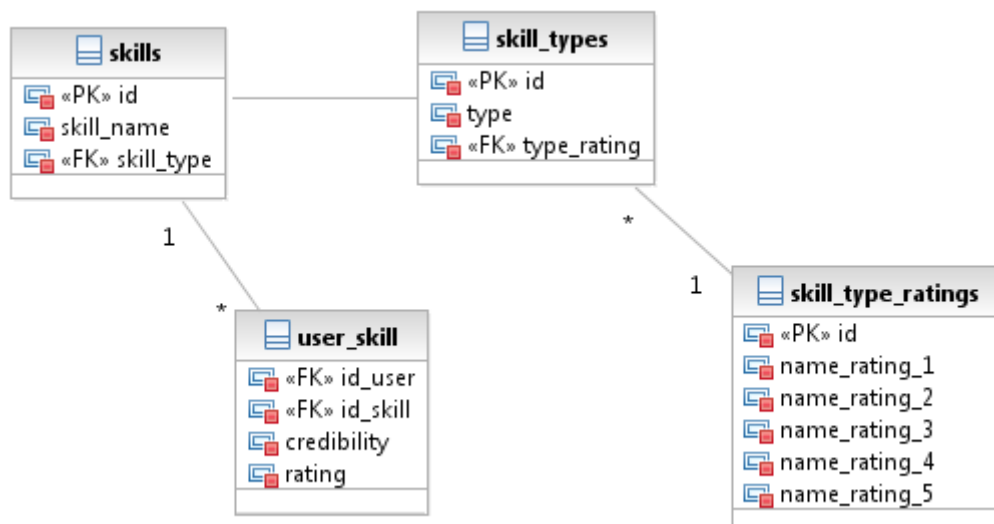
**Tab. 4.5 Prípad použitia ID9**

### 4.3.2.Návrh

Pridanie funkcionality na zadávanie úrovne ovládania znalosti si vyžiadali zmeny v logickom aj fyzickom modeli. V prepájacej tabuľke `user_skill` pribudol nový stĺpec `rating`, kde sa ukladá ohodnotenie od 1 do 5, kde 5 je najvyššie možné ohodnotenie.

Pre slovné vyjadrenie jednotlivých hodnôt ohodnotenia bola vytvorená tabuľka `skill_type_rating`, kde sú slovné vyjadrené všetky možné ohodnotenia pre všetky typy znalostí. V tabuľke `skill_types` potom pribudol cudzí kľúč pre tabuľku `skill_type_rating`. Takto môžeme ľahko docieľiť, že jeden súbor slovných ohodnotení od 1 do 5 je využiteľný pre viac typov znalostí.

Časť fyzického modelu, ktorá prešla pre tento US úpravami je znázornená na Obr.4.5.



**Obr.4.5 Fyzický model pre US18**

### 4.3.3.Implementácia

Pre implementáciu funkcionality pridania znalosti s ohodnotením bolo nutné zmeniť niektoré existujúce metódy SkillControllera.

```

public function addNewSkill($id_user, $skill, $cred, $lvl) {
    $connection = Yii::app()->db;

    // este nema
    $cmd = $connection->createCommand("INSERT INTO user_skill (id_user, id_skill, credibility,
rating) values (:id_user, :id_skill, :credibility, :rating)");
    $cmd->bindParam(":id_user", $id_user, PDO::PARAM_INT);
    $cmd->bindParam(":id_skill", $skill, PDO::PARAM_INT);
    $cmd->bindParam(":credibility", $cred, PDO::PARAM_STR);
    $cmd->bindParam(":rating", $lvl, PDO::PARAM_INT);
    $q = $cmd->execute();
  
```

```

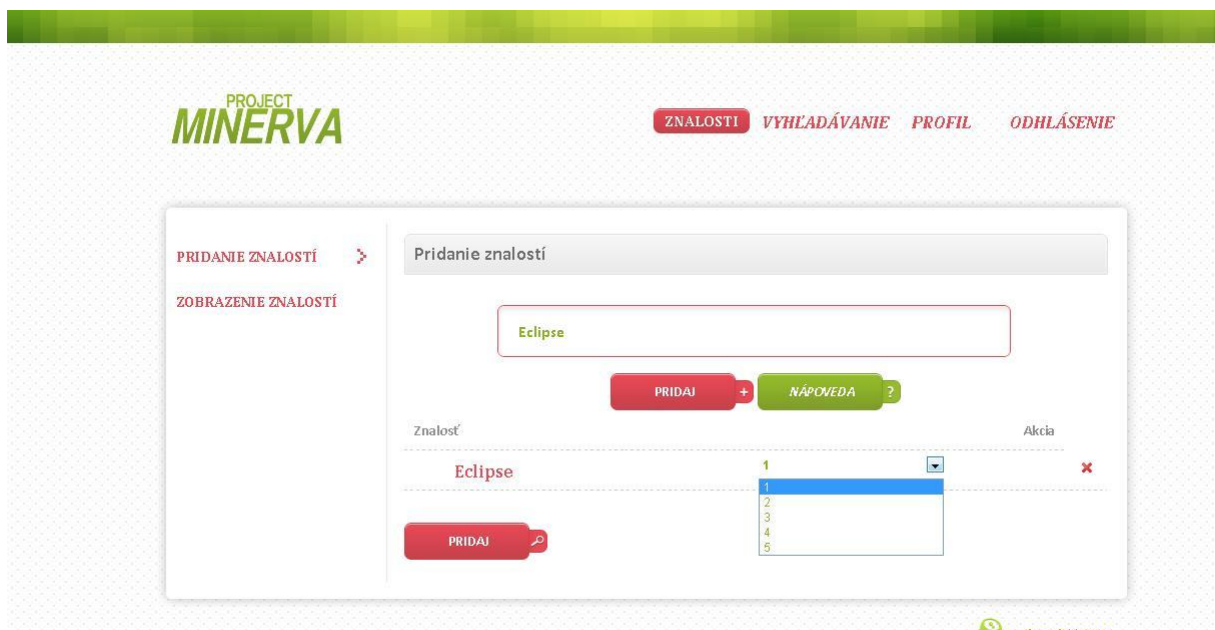
return 1;
}

public function actionAddSkill($id) {
    $model = new SkillAutocomplete();
    // if it is ajax validation request
    if (isset($_POST['ajax']) && $_POST['ajax'] === 'skill-form') {
        echo CActiveForm::validate($model);
        Yii::app()->end();
    }

    if (isset($_POST['SkillAutocomplete'])) {
        $perm = Yii::app()->user->getState('permissions');
        if((Yii::app()->user->getId() == $id /*pridavam skill sam sebe*/) ||
($perm['ADD_OTHER_SKILL'] /*skill mi pridava osoba co ma na to pravo*/))
        {
            foreach($_POST['cislo'] as $id => $skill) {
                // TODO credibility
                $this->addNewSkill(Yii::app()->user->getId(), (int) $skill, 'student',
                $_POST['level'][$id]);
            }
            $this->render('/skill/form', array('model' => $model, 'inserted' => true));
            Yii::app()->end();
        }
        else /*zakazana akcia*/
        {
            $this->render('error', array( 'code' => '404', 'message' => 'You are not authorized to perform
this action.));
            Yii::app()->end();
        }
    }
    $this->render('/skill/form', array('model' => $model));
}
}

```

Vzhľad formulára po pridaní funkcionality vidieť na Obr.4.6.



Obr.4.6 Formulár pre pridávanie znalostí s ohodnotením

### 4.3.4. Akceptačné testy

Pre testovanie bol navrhnutý scenár použitia pre pridanie znalosti s ohodnotením.

|                           |  |  |  |
|---------------------------|--|--|--|
| <b>Názov</b>              | Pridanie znalosti s ohodnotením  | ID Testu                                   | US18-01  |
| <b>Rozhranie</b>          | Obrazovka pridanie znalostí  | ID UC                                      | ID 9   |
| <b>Účel</b>               | Pridanie novej znalosti študentovi s ohodnotením úrovne ovládania znalostí |  |  |
| <b>Vstupné podmienky</b>  | Prihlásený používateľ  |  |  |
| <b>Výstupné podmienky</b> | Pridaná znalosť  |  |  |
| <b>Krok</b>               | <b>Akcia</b>   | <b>Očakávaná akcia</b>                     | <b>Skutočná akcia</b>  |
| 1.                        | V menu kliknem na tlačidlo zobrazenia formuláru pre pridanie znalostí      | Zobrazenie formuláru pre pridanie znalostí | Zobrazenie formuláru pre pridanie znalostí                           |
| 2.                        | Vyberiem pomocou autocomplete znalosť a pridám do zoznamu na pridanie.     | Znalosť sa pridá do zoznamu.               | Znalosť sa pridala do zoznamu  |
| 3.                        | Vyberiem úroveň ovládania znalosti   | Select sa nastaví na vybratú úroveň        | Select sa nastavil na vybratú úroveň                                 |
| 4.                        | Kliknem na tlačidlo pridania znalostí                                      | Do databázy sa uložia údaje                | V databáze bola priradená študentovi nová znalosť a jej ohodnotenie. |

Tab.4.6 Pridanie znalosti s ohodnotením študentovi

## Príloha A

---

### Dokumentácia k inžinierskemu dielu po 2.šprinte

S T U • •  
• • • • •  
F I I T •  
• • • • •

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE  
Fakulta informatiky a informačných technológií

## Znalosti a zručnosti študentov

### Tímový projekt

### Dokumentácia k inžinierskemu dielu

#### Tím č.16 – WeKnowIT

Bc. Jakub Šalmík

Bc. Patrik Polakovič

Bc. Adam Kobyda

Bc. Michal Chylik

Bc. Peter Ivanec

Bc. Anton Szórád

Akademický rok: 2011/2012

## Obsah

---

|   |    |
|---|----|
| Úvod .....  | 64 |
| Účel a rozsah dokumentu.....                      | 64 |
| 1. Prvý šprint – Trójska vojna .....              | 65 |
| 1.1. US1 Návrh šablóny GUI .....                  | 65 |
| 1.1.1. Analýza .....                              | 65 |
| 1.1.2. Návrh .....                                | 66 |
| 1.1.3. Implementácia.....                         | 67 |
| 1.1.4. Testovanie .....                           | 69 |
| 1.2. US4 Editácia profilu .....                   | 69 |
| 1.2.1. Analýza .....                              | 69 |
| 1.2.2. Návrh .....                                | 70 |
| 1.2.3. Implementácia.....                         | 71 |
| 1.2.4. Testovanie .....                           | 72 |
| 1.3. US 6 – Zadanie znalostí.....                 | 72 |
| 1.3.1. Analýza .....                              | 72 |
| 1.3.2. Návrh .....                                | 73 |
| 1.3.3. Implementácia.....                         | 74 |
| 1.3.4. Akceptačné testy .....                     | 74 |
| 1.4. US 11 - Vytvorenie prvotnej bázy pojmov..... | 75 |
| 1.4.1. Analýza .....                              | 75 |
| 1.4.2. Návrh .....                                | 75 |
| 1.4.3. Implementácia.....                         | 75 |
| 2. Druhý šprint – Prvá púnska vojna.....          | 76 |
| 2.1. US2 Prihlasovanie používateľa .....          | 76 |
| 2.1.1. Analýza .....                              | 76 |
| 2.1.2. Návrh .....                                | 77 |
| 2.1.3. Implementácia.....                         | 78 |
| 2.1.4. Testovanie .....                           | 80 |
| 2.2. US3 Prezeranie profilu.....                  | 81 |
| 2.2.1. Analýza .....                              | 82 |
| 2.2.2. Návrh .....                                | 82 |
| 2.2.3. Implementácia.....                         | 83 |
| 2.2.4. Testovanie .....                           | 83 |

|        |   |    |
|--------|---|----|
| 2.3.   | US7 Zobrazenie vlastných znalostí ..... | 84 |
| 2.3.1. | Analýza .....                           | 84 |
| 2.3.2. | Návrh .....                             | 84 |
| 2.3.3. | Implementácia .....                     | 85 |
| 2.3.4. | Akceptačné testy .....                  | 86 |
| 2.4.   | US12 Kategorizácia znalostí.....        | 86 |
| 2.4.1. | Analýza .....                           | 86 |
| 2.4.2. | Návrh .....                             | 86 |
| 2.4.3. | Implementácia .....                     | 87 |

# Úvod

---

## Účel a rozsah dokumentu

Každý študent disponuje určitými znalosťami a zručnosťami. V súčasnosti neexistuje vo vysokoškolskom prostredí systém, ktorý by tieto údaje dokázal zbierať a vyhodnocovať. Pritom sú to údaje, ktoré majú vysokú cenu jednak pre školu, pri vyberaní vhodných študentov na stáže, projekty, bakalárske či diplomové práce, ako aj pre firmy, ktoré sa väčšinou o študentov začínajú zaujímať už v čase ich štúdia na vysokej škole. V neposlednom rade ide o užitočné informácie aj pre študentov samotných, ktorí by si za predpokladu, že poznajú zručnosti a znalosti ostatných spolužiakov, mohli ľahšie vyberať spolupracovníkov do rôznych tímových projektov.

Cieľom nášho tímového projektu je vytvorenie systému na zhromažďovanie, uchovávanie a prezentáciu znalostí a zručností študentov.

Ako implementačné prostredie sme si zvolili v súčasnosti najpoužívanejší jazyk pre tvorbu webových aplikácií, jazyk PHP. Vybrali sme si aplikačný rámec Yii, ktorý sa nám po analýze dostupných možností v tejto oblasti, javil najvhodnejší. Voľba databázového systému bola podmienená požiadavkami zákazníka, preto je zvoleným databázovým systémom PostgreSQL.

Projekt je riešený agilnou vývojovou metodikou Scrum. V nej je vývoj rozdelený na dvojtýždňové úseky, tzv. šprinty. V rámci každého šprintu robíme analýzu, návrh, implementáciu a testovanie zvolených užívateľských príbehov. Pre pomenovanie jednotlivých šprintov sme vybrali zoznam najväčších vojen v histórii ľudstva.

Predkladaný dokument, ktorý predstavuje dokumentáciu k inžinierskemu dielu, zachytáva prvé dva šprinty vývoja systému.



# 1. Prvý šprint – Trójska vojna

---

V rámci prvého šprintu sme riešili nasledovné užívateľské príbehy.

- US1 – Návrh šablóny GUI
- US4 – Editácia profilu
- US6 – Zadanie znalostí
- US11 – Vytvorenie prvotnej bázy pojmov

## 1.1. US1 Návrh šablóny GUI

---

Grafický návrh používateľského rozhrania a jeho následná implementácia.

### 1.1.1. Analýza

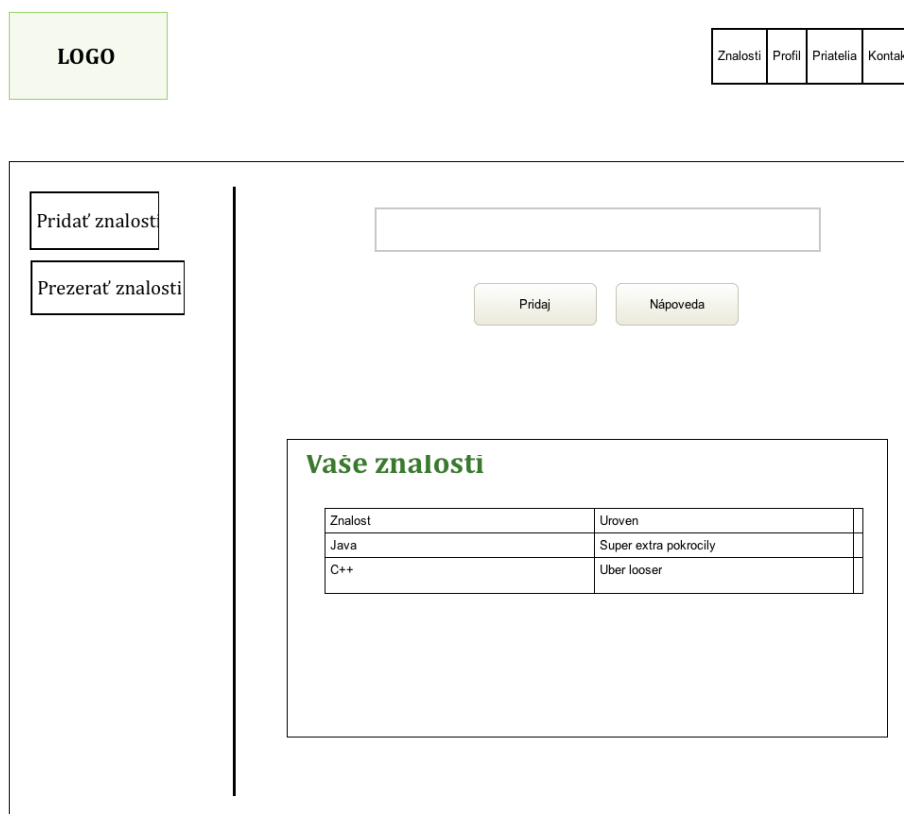
Prvotnou požiadavkou celého systému bola prehľadnosť a použiteľnosť rozhrania. Musíme zabezpečiť používateľskú spokojnosť a zabezpečiť lepšiu orientáciu v systéme.

V systéme sa budú nachádzať klasické prvky webových stránok ako sú:

- Formulárové polia
- Tlačidlá pre odosielanie údajov
- Tabuľky

Všetky prvky nesmú byť len graficky príťažlivé, ale hlavne funkčné a účelné.

Na obrázku 1.1 je znázornený wireframe (obrys) aplikácie, kde je znázornené rozloženie celej stránky. Obsahuje dve navigačné menu pre zlepšenie orientácie v systéme. Taktiež znázorňuje formulár a tabuľku (iba obrys nie reálny návrh).



Obr. 1.1 : Wireframe používateľského rozhrania

Keďže sme sa nedržali žiadneho z pôvodných systémov, ktoré boli pred tým realizované, návrh rozhrania sme robili podľa štandardov moderných webových technológií s prihliadnutím na to, čo nám v súčasnosti web poskytuje.

Požiadavky na rozhranie od zákazníka sú taktiež minimálne, keďže zákazník sa s podobným systémom ešte nestretol. To nám dalo voľnú ruku, samozrejme s prihliadnutím na štandardy webových technológií.

### 1.1.2.Návrh

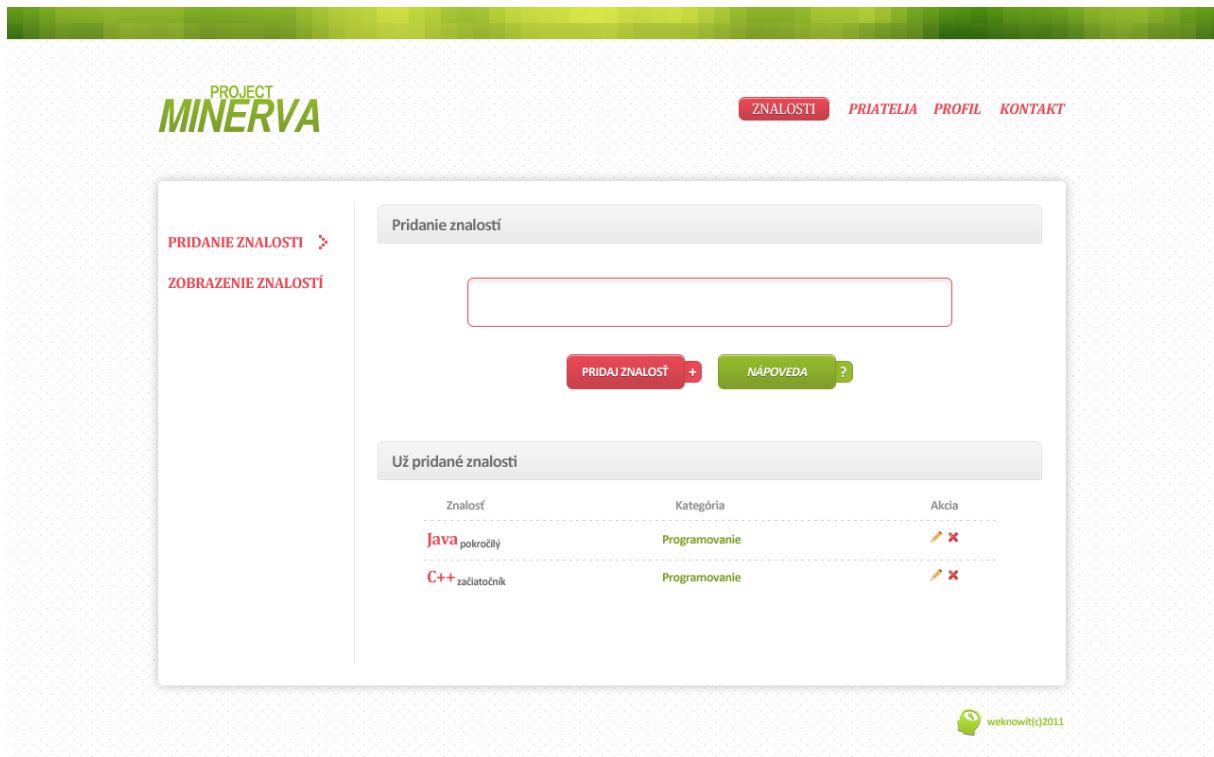
Pri návrhu bolo prioritou zvýrazniť prvky, ktoré sú dôležité. Horné horizontálne menu bude farebne zvýraznené, pre lepšiu orientáciu bude zvýraznená aj aktuálna položka v menu.

Taktiež navrhujeme aby centrálny panel bol opticky oddelený od bočného vertikálneho menu, ktoré bude mať iba orientačný charakter, čo ale neuberá na jeho dôležitosť v použiteľnosti systému.

Centrálny panel bude združovať všetky funkcie poskytujúce systémom. V závislosti od kombinácie horizontálneho a vertikálneho menu bude zobrazovať obsah, ktorý bude dobre čitateľný a štruktúrovaný.

Celý systém bude v moderných a výrazných farbách, aby sme predišli fádnosti, ale zachovali profesionalitu a čistý vzhľad. Dobrý výber farieb je kľúčový a bude systém charakterizovať.

Na obrázku 1.2 je znázorný grafický návrh obrazovky systému.



Obr. 1.2 : Grafický návrh obrazovky systému

### 1.1.3. Implementácia

Implementácia grafického návrhu sa skladala z viacerých častí.

#### 5. Nastrihanie obrázkov v požadovanej forme.

Všetky obrázky boli nastrihané a uložené vo formáte png, ktorý je dnes štandardom na poli webového dizajnu.

#### 6. Vytvorenie HTML štruktúry

Z predloženého návrhu bolo potrebné vymedziť jednotlivé prvky tak aby sa dali transformovať na elementy zrozumiteľného jazyku HTML, konkrétne verzii 5. Maximum prvkov bolo implementovaných pomocou elementov DIV. Samotné HTML elementy boli implementované tak, aby neobsahovali žiadne vnorené informácie o ich štýle. Všetky vlastnosti sú definované v súboroch css. Výslednú štruktúru kódu je možné vidieť nižšie.

```
<!DOCTYPE html>
<html lang="en">
  <head>

    <title>Projekt Minerva</title>

    <meta charset="utf-8" />
    <meta name="author" content="WeKnowIT">
    <meta name="description" content="Projekt Minerva, zameraný za zber zručností a znalostí študentov">
    <meta name="keywords" content="minerva, znalost, znalosti, zrucnost, zrucnosti, studenti, fiit">
```

```

<link rel="stylesheet" href="css/reset.css" media="all">
<link rel="stylesheet" href="css/style.css" media="all">
<link rel="stylesheet" href="css/form.css" media="all">

<script src="js/custom-form-elements.js" type="text/javascript"></script>

</head>

<body>

<div id='top_banner'></div>

<!-- header -->

<div id='top_container'>
  <div id='top_logo'>
    <a href='#'><img src='design/logo.png'></a>
  </div>
  <div id='top_menu'>
    <ul>
      <li><a href='#'>ZNALOSTI</a></li>
      <li><a href='#'>PRIATELIA</a></li>
      <li class='active_menu'><a href='#'>PROFIL</a></li>
      <li><a href='#'>KONTAKT</a></li>
    </ul>
  </div>
</div>

<!-- body -->

<div id='body_container'>
  <div id='body_top'></div>
  <div id='body_fix2'>
  <div id='body_fix'>
    <div id='body_left'>
      <ul>
        <li><a href='#'>PRIDANIE ZNALOSTÍ</a></li>
        <li class='active_menu'><a href='#'>ZOBRAZENIE ZNALOSTÍ</a></li>
        <li><a href='#'>VÝMAZANIE ZNALOSTÍ</a></li>
      </ul>
    </div>
    <div id='body_right'>
      <div id='content'>

        <!-- obsah -->

        <h1>Pridanie znalostí</h1>

        <p id='skill_form'>

          <input type='text' class='input_skill' value='Java EE'><br>
          <input type='button' class='button_add' value='PRIDAJ ZNALOSŤ'>
          <input type='button' class='button_help' value='NÁPOVEDA'>

        </p>

        <!-- koniec obsahu -->

      </div>
    </div>
  </div>
  </div>
  </div>
  <div id='body_bottom'></div>
  <div id='footer'>
    <div id='footer_head'></div>
    <div id='footer_text'>weknowit (c)2011</div>
  </div>
</div>

</body>
</html>

```

## 7. Vytvorenie CSS súborov

Pre potreby projektu boli vytvorené tri samostatné CSS súbory.

- reset.css – obsahuje vynulovanie všetkých atribútov, pre všetky definované html elementy. Tento postup eliminuje väčšinu problémov spojených s rozdielnym základným nastavením webových prehliadačov, teda rôznym zobrazovaním rovnakých prvkov.
- style.css - obsahuje definície všetkých elementov používaných na jednotlivých stránkach.
- form.css – obsahuje definície štýlov pre formulárové elementy ako checkbox, radio alebo select. Tieto elementy sa za normálnych okolností nedajú ľubovoľne meniť, lebo ich vzhľad je definovaný prehliadačom, preto bol použitý hotový skript dostupný na <http://ryanfait.com/resources/custom-checkboxes-and-radio-buttons/>, ktorý následne umožní elementy formulára upravovať pomocou CSS.

## 8. Optimalizácia pre prehliadače

Po dokončení predchádzajúcich krokov, bolo nutné spraviť optimalizáciu pre najpoužívnejšie prehliadače. Problém so zlým zobrazením sa vyskytol len pri formulároch v internet exploreri, ktoré boli následne vyriešené. Iné chyby sa neobjavili hlavne vďaka opatreniam uvedeným v predchádzajúcich bodoch.

### 1.1.4. Testovanie

Testovanie spočívalo v overení implementovaného rozhrania vo všetkých majoritných webových prehliadačoch. Testovalo sa vždy len na aktuálnej verzii prehliadača. Jediná výnimka bola udelená prehliadaču Internet Explorer, kde sa testovalo na verzii 9 aj 8, kvôli obrovským rozdielom medzi týmito dvoma verziami. Testy preukázali, že všetky prehliadače zobrazujú obsah korektne.

## 1.2. US4 Editácia profilu

---

Užívateľ bude mať možnosť úpravy základných údajov vo svojom profile.

### 1.2.1. Analýza

Keďže pre väčšinu používateľov systému bude registráciou do systému ich prvé prihlásenie, je umožnenie editovania základných údajov profilu nutnosťou. Pri prvej návšteve editácie profile to bude preto skôr vypĺňanie prázdnych položiek profilu, to je však z hľadiska systému presne rovnaká činnosť ako úprava už vyplnených údajov.

Tento jednoduchý profil bude obsahovať naozaj len základné informácie o užívateľovi, teda neobsahuje žiadne znalosti. Prihlasovacie údaje, teda login a heslo, nebude možné meniť, keďže sa využívajú prihlasovacie údaje z AIS. Ostatné položky už budú upravovateľné.

Cieľom je ponúknuť jednoduchý a prehľadný formulár, ktorý bude umožňovať úpravu všetkých základných údajov. Samozrejmosťou je, že bude obsahovať prednastavené hodnoty podľa aktuálnych údajov o prihlásenom používateľovi.

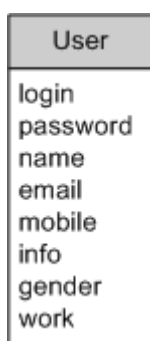
Pred vytvorením prezentačnej vrstvy profilu užívateľa je nutné vytvoriť doposiaľ neexistujúcu údajovú vrstvu, teda reprezentáciu profilu užívateľa v rámci databázy.

| Prípád použitia             | Editácia Profilu  |
|-----------------------------|---|
| <b>ID</b>                   | 1   |
| <b>Stručný popis</b>        | Používateľ si edituje základné údaje svojho profilu prostredníctvom formulára.  |
| <b>Primárny aktéri</b>      | Prihlásený používateľ   |
| <b>Vedľajší aktéri</b>      | Žiadny  |
| <b>Vstupné podmienky</b>    | Používateľ je prihlásený  |
| <b>Hlavný scenár</b>        | <ol style="list-style-type: none"> <li>5. Prípád použitia začína po zobrazení formuláru na editáciu profilu.</li> <li>6. Používateľ prepíše položky formulára, ktoré chcel editovať.</li> <li>7. Používateľ odošle formulár.</li> <li>8. Systém spracuje formulár a uloží údaje do databázy.</li> </ol> |
| <b>Výstupné podmienky</b>   | Žiadne  |
| <b>Alternatívne scenáre</b> | Žiadne  |

Tab.1.1 Prípád použitia ID 1

## 1.2.2.Návrh

Prvým krokom návrhu bolo vytvorenie logického a fyzického modelu profilu užívateľa. Logický model je znázornený na obrázku.



Obr.1.3 Logický model entity User

### Logický Model entity User

Pre fyzický model sa doplnil jednoznačný identifikátor `id_user`, ktorý je zároveň primárnym kľúčom a má vlastnosť `autoincrement`. Stĺpce `gender` a `work` sú dátového typu číselník. `Gender` môže nadobúdať hodnoty `male`(muž) a `female`(žena). `Work` môže nadobúdať taktiež dve hodnoty, `áno`(yes) a `nie`(no) a vyjadruje údaj o tom či užívateľ je momentálne zamestnaný.

| User |                |             |
|------|----------------|-------------|
| PK   | <u>id_user</u> | INTEGER     |
|      | login          | VARCHAR(20) |
|      | password       | VARCHAR(40) |
|      | name           | VARCHAR(40) |
|      | email          | VARCHAR(40) |
|      | mobile         | VARCHAR(13) |
|      | info           | TEXT(200)   |
|      | gender         | CHAR(6)     |
|      | work           | BIT         |

Obr.1.4 Fyzicky model entity User

Ďalším krokom bol návrh formulára a jeho logiky na základe prípadu použitia ID 1, ktorý sme uviedli vyššie. Návrh formulára je možné vidieť na obrázku č.3. Formulár obsahuje potrebné polia na vyplnenie osobných údajov. Polia meno a osobné číslo sú needitovateľné, ale sprehľadňujú celý formulár. Emailová adresa sa overuje interaktívne pomocou technológie ajax.

disabled input, neda sa menit obsah

ajaxové overenie emailu

Priezvisko a meno: Mak Jozef, Ing.

Osobné číslo: 48079

Email: jozef.mak@stu. ❌

Telefón: +421 911 222 890

Pohlavie:  Muž  Žena

Práca:  Ano  Nie

O mne: Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries.

**Uložiť**

Obr.1.5 Navrh formulara pre editáciu profilu

### 1.2.3.Implementácia

Pri implementácii boli využité výhody frameworku Yii, ktorý využívame pre náš projekt. Po vytvorení tabuľky User v databáze bol využitý CRUD generátor, ktorý automaticky vygeneroval potrebný formulár, jeho logiku a ukladanie spracovaných údajov do databázy. Takto bol implementovaný MVC vzor. Pochopiteľne, bolo potrebné mierne upraviť ako View (čiže samotný formulár), tak súbor UserController (teda jeho logiku a ukladanie údajov do databázy).

Úpravy logiky sa týkali najmä odstránenia možnosti úpravy užívateľského mena a hesla. Na prezentačnej vrstve (view) bolo potrebné urobiť zmeny súvisiace s implementáciou dizajnu. Zmena HTML kódu a vytvorenie štýlov pre použité elementy. Výsledok je možné vidieť na obrázku č.4.

V UserController bolo umožnené ajaxové overovanie formulára ako aj na prezentačnej vrstve. Nakoniec sa definovali pravidlá pre overovanie v modeli User a metóde Rules().

### Úprava profilu

|                   |   |
|-------------------|---|
| Meno a priezvisko | <input type="text" value="Michal Chylik"/>                      |
| Osobné číslo      | <input type="text" value="Mike22"/>                             |
| Email             | <input type="text" value="michal.chylik@stuba.sk"/>             |
| Mobil             | <input type="text" value="0911111111"/>                         |
| Pohlavie          | <input checked="" type="radio"/> Muž <input type="radio"/> Žena |
| Práca             | <input checked="" type="radio"/> Áno <input type="radio"/> Nie  |
| O mne             | <input type="text" value="Random text"/>                        |

Obr.1.6 Formular pre editáciu profilu

## 1.2.4. Testovanie

Testovali sa všetky prípady zadaných hodnôt do polí formulára a ich následné spracovanie systémom. Rovnako sa testovalo aj ajaxové overovanie emailovej adresy. Chyby objavené neboli.

## 1.3. US 6 – Zadanie znalostí

---

Používateľ má umožnené pridávať do svojho profilu znalosti, alebo znalosti niekomu inému podľa oprávnení.

### 1.3.1. Analýza

Tento systém je zameraný na správu a zobrazovanie znalostí používateľov. Preto je nepochybne dôležitým krokom v tomto procese priradenie znalostí používateľovi.



Ide o jednoduchý formulár, ktorý umožní prehľadne pridávať znalosti sebe samému, alebo používateľovi, ktorému máme oprávnenie znalosti pridať.

Vstupné podmienky: prihlásený používateľ

Výstupné podmienky: žiadne

Aktéri: používateľ

Hlavný scenár:

1. Zobrazenie formuláru na pridanie znalostí,
2. Používateľ zvolí a pridá požadované znalosti do profilu,
3. Používateľ potvrdí zmeny,
4. Systém formulár spracuje a priradí používateľovi v databáze príslušné znalosti

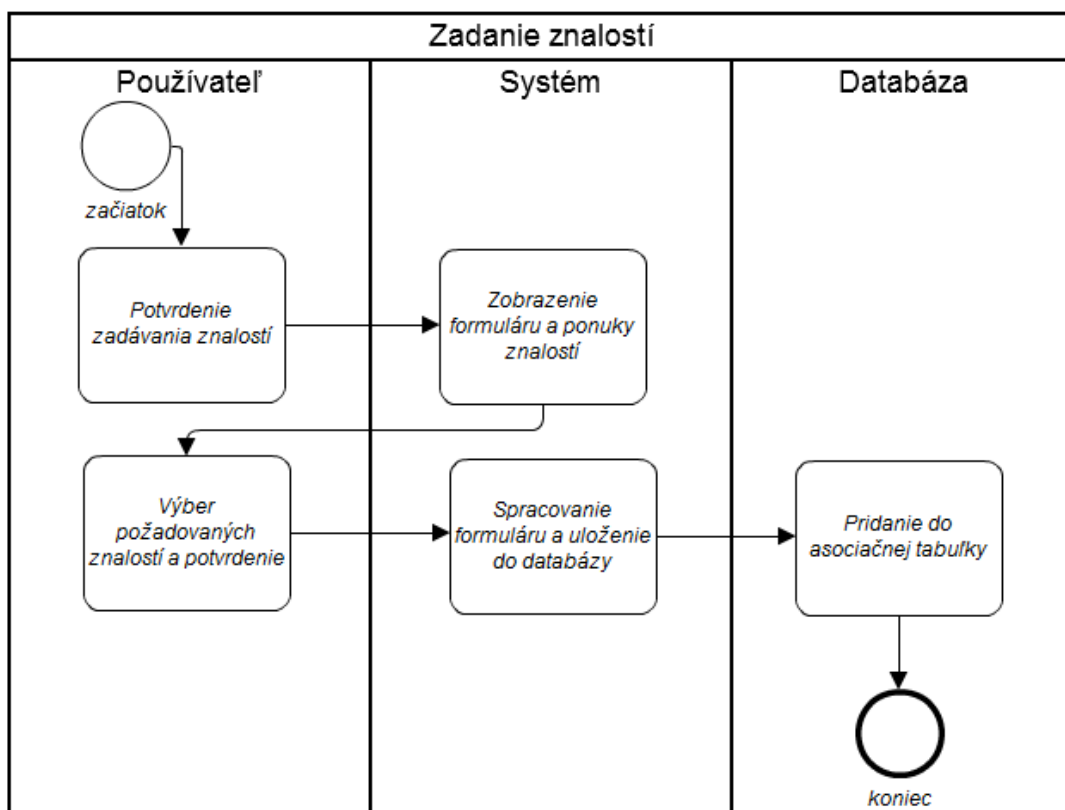
### 1.3.2.Návrh

Funkcionalita je navrhnutá v databáze pomocou asociačnej tabuľky, ktorá obsahuje cudzie kľúče, a to ID používateľa a priradené znalosti z tabuľky znalostí.

| USER KNOWLEDGE |            |           |
|----------------|------------|-----------|
| FK             | ID_USER    | INTEGER   |
| FK             | ID_ZNALOST | INTEGER[] |

**Tab. 1.2 fyzicky model entity user knowledge**

Aktivity diagram:



Obr. 1.7 ActivityDiagram ZadanieZnalosti

### 1.3.3.Implementácia

Implementácia návrhu prebiehala pomocou frameworku Yii, kde sa po vytvorení tabuľky pre ukladanie vygeneroval CRUD generátorom formulár pre zadanie požadovaných znalostí do databázy. Tento formulár bol ešte upravený pre správne zobrazovanie a spôsob ukladania údajov do databázy.

### 1.3.4.Akceptačné testy

Pre testovanie bol navrhnutý scenár použitia pre zadanie znalostí používateľa.

|                           |                                     |                        |                       |
|---------------------------|-------------------------------------|------------------------|-----------------------|
| <b>Názov</b>              | Pridanie znalostí                   | ID 1                   |                       |
| <b>Rozhranie</b>          | Obrazovka formuláru                 | ID UC6                 |                       |
| <b>Účel</b>               | Pridanie nových znalostí do profilu |                        |                       |
| <b>Vstupné podmienky</b>  | Prihlásený používateľ               |                        |                       |
| <b>Výstupné podmienky</b> | Žiadne                              |                        |                       |
| <b>Krok</b>               | <b>Akcia</b>                        | <b>Očakávaná akcia</b> | <b>Skutočná akcia</b> |

|    |   |  |  |
|----|---|--|--|
| 1. | Požiadavka na zobrazenie obrazovky s formulárom pridávania znalostí | Zobrazenie obrazovky                       | Zobrazenie obrazovky                       |
| 2. | Vybratie požadovaných znalostí a potvrdenie                         | Priradenie znalostí a uloženie do databázy | Priradenie znalostí a uloženie do databázy |

Tab. 1.3 AkcTest ZadanieZnalosti

## 1.4. US 11 - Vytvorenie prvotnej bázy pojmov

Ako používateľ chcem mať k dispozícii pojmy opisujúce znalosti, aby som s nimi mohol pracovať.

### 1.4.1. Analýza

Vytvorenie prvotnej bázy znalostí a zručností je základný kameň nášho projektu. Bez nej by sa nič iné nedalo spraviť. Z tejto prvotnej bázy pojmov sa bude vychádzať pri hierarchizácii zručností a znalostí študentov. Je dôležité oddeliť tieto znalosti od grúp, čo bude pole znalostí. Podľa logického modelu budú tri rôzne druhy grúp: práca, predmet, projekt.

### 1.4.2. Návrh

Ako databázu použijeme PostgreSQL.

Návrh tabuľky je nasledovný

| Názov stĺpca v tabuľke | Typ           |
|------------------------|---------------|
| Id                     | integer       |
| skill_name             | character(80) |

Tab. 1.4 Navrh tabuľky Baza pojmov

### 1.4.3. Implementácia

Ako GUI sa použil pgAdmin III. Tabuľka s prvotnou bázou pojmov sa vytvorila nasledovne:

```
-- Table: skills

CREATE TABLE skills
(
  id serial,
  skill_name character(80),
  CONSTRAINT skills_id_key UNIQUE (id )
)
WITH (
  OIDS=FALSE
);
ALTER TABLE skills
  OWNER TO postgres;
```

## 2. Druhý šprint – Prvá púnska vojna

---

V rámci druhého šprintu sme riešili nasledovné užívateľské príbehy.

- US2 – Prihlasovanie používateľa
- US3 – Prezeranie profilu
- US5 – Zobrazenie kľúčových slov v systéme
- US7 – Zobrazenie vlastných znalostí
- US8 – Zobrazenie znalostí študenta
- US12 – Kategorizácia znalostí

### 2.1. US2 Prihlasovanie používateľa

---

Ako používateľ sa chcem prihlásiť do systému, aby som v ňom mohol pracovať.

#### 2.1.1. Analýza

Nakoľko väčšina systémov na škole používa pre prihlásenie do systému meno a heslo z AIS, tak sme sa aj my rozhodli pre tento spôsob prihlásenia. Tiež chceme aby systém bol jednoduchý pre používanie a teda prihlásenie bude používať javascript pre kontrolu vyplnených polí.

Vstupné podmienky: neprihlásený používateľ

Výstupné podmienky: žiadne

Aktéri: používateľ


Flow:

1. Používateľ vyberie možnosť prihlásenia
2. Systém zobrazí prihlasovací formulár
3. Používateľ vyplní formulár a potvrdí jeho vyplnenie
4. V prípade:
  - a. nevyplnenia povinných údajov:
    - i. systém zobrazí chybovú správu
    - ii. tok pokračuje v bode 2.
5. Systém overí správnosť údajov cez LDAP
6. V prípade:
  - a. neúspešného overenia:

- i. systém zobrazí chybovú správu
  - ii. tok pokračuje v bode 2.
7. Systém prihlási používateľa
  8. Systém zobrazí úvodnú obrazovku používateľa
  9. Koniec

## 2.1.2. Návrh

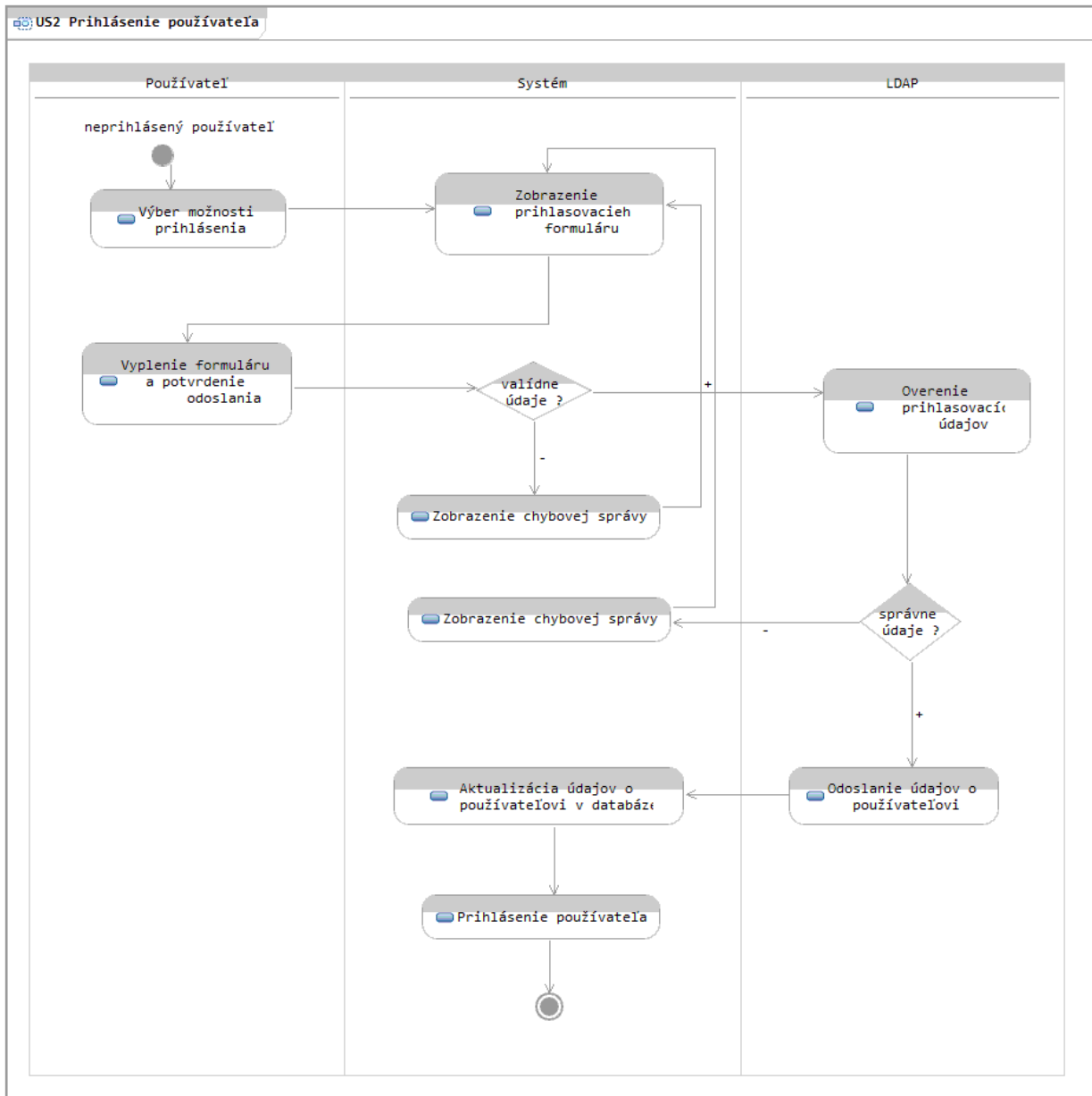
Na obrázku č. 2.1 je zobrazená obrazovka pre prihlásenie používateľa. Prihlasuje sa pomocou údajov z AIS, a preto je tu vynechaná možnosť zaregistrovať sa. Možnosť „zapamätať prihlásenie“ sa podľa uváženia všetkých bezpečnostných rizík zanechá alebo odstráni.



The image shows a login form with a header bar labeled "Login". Below the header, there are two input fields: "Login" and "Heslo". Below the "Heslo" field, there is a checkbox labeled "zapamätaj prihlásenie". At the bottom of the form, there is a red button labeled "Prihlásiť" with a pencil icon.

**Obr. 2.1 Prihlásenie používateľa**

Obrázok č. 2.2. nám znázorňuje diagram aktivít prípadu použitia, ktorý sa realizuje pri prihlásení používateľa. Znázorňuje aj alternatívne toky, ktoré sa vykonajú pri chybe.



Obr. 2.2 : Diagram aktivít pre prihlásenie používateľa.

### 2.1.3.Implementácia

Najdôležitejšou súčasťou prihlásenia používateľa je overovanie údajov cez LDAP (Lightweight Directory Access Protocol). V nasledujúcom kóde je ukázané ako je toto overovanie realizované.

```
private function logonCheckLDAP()
{
    $basedn='ou=People,dc=stuba,dc=sk';
```

```
$ds = @ldap_connect('ldap://ldap.stuba.sk', 636);
//nastavovanie ldap
@ldap_set_option($ds, LDAP_OPT_PROTOCOL_VERSION, 3);
//tzv bind cize overenie mena hesla
$ldapBindResult = @ldap_bind($ds, 'uid='.$this->username.',ou=People,dc=stuba,dc=sk',
$this->password);
if (!$ldapBindResult)
{
    @ldap_close($ds);
    return false;
}
//filtrujeme co chceme dostat
$ldapFilter = array("uid", "userPassword", "employeetype", "uisid", "cn", "sn", "givenname");
//hladame
$ldapSearchResult = @ldap_search($ds, $basedn, 'uid='.$this->username, $ldapFilter);
if ($ldapSearchResult)
{
    $this->result = @ldap_get_entries($ds, $ldapSearchResult);
    @ldap_close($ds);
    return true;
}
else
{
    @ldap_close($ds);
    return false;
}
@ldap_close($ds);
return false;
}
```

Ak sú údaje dobré a overenie sa podarí tak do premennej **result** sa uloží výsledok funkcie `@ldap_search`, ktorá nám vráti meno, id, osobné číslo a typ zamestnanca priamo z AIS.

Ak sa používateľ prihlasuje do systému prvý krát (napr. študent prvého ročníka), systém to overí a automaticky v databáze uloží nového používateľa. V nasledujúcom kóde je možné vidieť ako to systém vykoná.

```
$connection=Yii::app()->db;

$command1 = $connection->createCommand("SELECT id_user FROM \"user\" WHERE login = :login
LIMIT 1");
$command1->bindParam(":login", $this->result[0]["uid"][0], PDO::PARAM_STR);

$user = $command1->query();

if($user->getRowCount() == 0)
{
    // ulozim noveho pouzivatelya
    $command2 = $connection->createCommand("INSERT INTO \"user\" (login, name)
VALUES (:login, :name)");
```

```

        $command2->bindParam(":login",    $this->result[0]["uid"][0], PDO::PARAM_STR);
        $command2->bindParam(":name",    $this->result[0]["cn"][0],
PDO::PARAM_STR);
$command2->execute();

        $user = $command1->query();

        $row = $user->read();
        $this->unique = array(

                "rid"    => $row['id_user'],

                "id"      => "sem AIS id",

                "perm" => "sem prava"

        );

        return true;
    }

```

Obe funkcie sa volajú z triedy UserIdentity, ktorá je komponentom, ktorá v sebe nesie informácie o používateľovi.

## 2.1.4. Testovanie

Nasledovné tri testovacie scenáre sme navrhli pre otestovanie prihlásenia používateľa.

|                           |   |   |   |
|---------------------------|---|---|---|
| <b>Názov</b>              | Úspešné prihlásenie do systému                        | <b>ID Testu</b>                                   | XX-01   |
| <b>Rozhranie</b>          | Obrazovka prihlásenia                                 | <b>ID UC</b>                                      | XX  |
| <b>Účel</b>               | Prihlásenie používateľa s prihlasovacími údajmi z AIS |   |   |
| <b>Vstupné podmienky</b>  | Správne prihlasovacie údaje                           |   |   |
| <b>Výstupné podmienky</b> | Žiadne  |   |   |
| <b>Krok</b>               | <b>Akcia</b>  | <b>Očakávaná reakcia</b>                          | <b>Skutočná akcia</b>                                     |
| 1.                        | Zadanie prihlasovacieho mena do príslušnej kolónky.   | Zobrazenie prihlasovacieho mena v kolónke "Login" | Prihlasovacie meno zobrazené vo formulárovom poli "Login" |



|    |  |  |   |
|----|--|--|---|
| 2. | Zadanie prihlasovacieho hesla do príslušnej kolónky. | Zobrazenie hesla v tvare hviezdíčiek v kolónke "Heslo" | Vyhviezdičkované heslo zobrazené vo formulárovom poli "Heslo" |
| 3. | Kliknutie na tlačidlo "Prihlásiť"                    | Prihlásenie používateľa do systému.                    | Prihlásený používateľ a zobrazenie úvodnej obrazovky.         |

Tab. 2.1 Prihlasovanie Test1

|                           |   |   |   |       |
|---------------------------|---|---|---|-------|
| <b>Názov</b>              | Nevyplnenie poľa "Heslo"                                    |   | <b>ID Testu</b>   | XX-02 |
| <b>Rozhranie</b>          | Obrazovka prihlásenia                                       |   | <b>ID UC</b>  | XX    |
| <b>Účel</b>               | Zobrazenie chybovej hlášky: "Pole heslo nemôže byť prázdne" |   |   |       |
| <b>Vstupné podmienky</b>  | Žiadne  |   |   |       |
| <b>Výstupné podmienky</b> | Žiadne  |   |   |       |
| <b>Krok</b>               | <b>Akcia</b>  | <b>Očakávaná reakcia</b>                          | <b>Skutočná akcia</b>   |       |
| 1.                        | Zadanie prihlasovacieho mena do príslušnej kolónky.         | Zobrazenie prihlasovacieho mena v kolónke "Login" | Prihlasovacie meno zobrazené vo formulárovom poli "Login"                               |       |
| 2.                        | Kliknutie na tlačidlo "Prihlásiť"                           | Zobrazenie chybovej hlášky.                       | Zobrazená chybová hláška: "Pole heslo nemôže byť prázdne" pri formulárovom poli "Heslo" |       |

Tab. 2.2 Prihlasovanie Test2

|                           |  |  |  |       |
|---------------------------|--|--|--|-------|
| <b>Názov</b>              | Neúspešné prihlásenie do systému.                            |  | <b>ID Testu</b>  | XX-03 |
| <b>Rozhranie</b>          | Obrazovka prihlásenia  |  | <b>ID UC</b>   | XX    |
| <b>Účel</b>               | Zobrazenie chybovej hlášky: "Zadané zlé prihlasovacie údaje" |  |  |       |
| <b>Vstupné podmienky</b>  | Nesprávne prihlasovacie údaje                                |  |  |       |
| <b>Výstupné podmienky</b> | Žiadne   |  |  |       |
| <b>Krok</b>               | <b>Akcia</b>   | <b>Očakávaná reakcia</b>                               | <b>Skutočná akcia</b>  |       |
| 1.                        | Zadanie prihlasovacieho mena do príslušnej kolónky.          | Zobrazenie prihlasovacieho mena v kolónke "Login"      | Prihlasovacie meno zobrazené vo formulárovom poli "Login"                            |       |
| 2.                        | Zadanie prihlasovacieho hesla do príslušnej kolónky.         | Zobrazenie hesla v tvare hviezdíčiek v kolónke "Heslo" | Vyhviezdičkované heslo zobrazené vo formulárovom poli "Heslo"                        |       |
| 3.                        | Kliknutie na tlačidlo "Prihlásiť"                            | Vypísanie chybovej hlášky.                             | Zobrazená chybová hláška: "Zadané zlé prihlasovacie údaje" nad tlačidlom "Prihlásiť" |       |

Tab. 2.3 Prihlasovanie Test3

## 2.2. US3 Prezeranie profilu

Ako prihlásený používateľ systému chcem mať možnosť zobraziť si vlastný profil, teda moje osobné údaje.

### 2.2.1. Analýza

Vstup: prihlásený používateľ

Výstup: obrazovka s profilom

Aktéri: používateľ

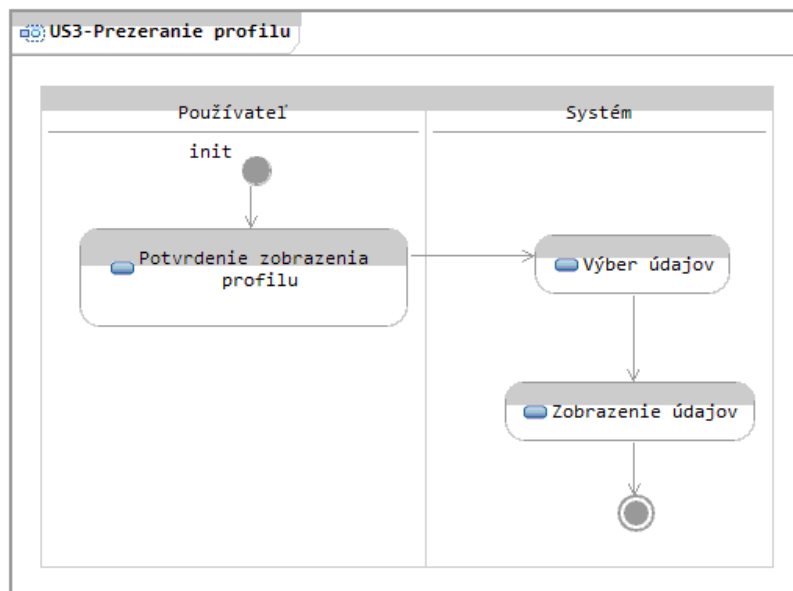
Flow:

5. Používateľ vyberie možnosť zobrazenia vlastného profilu
6. Systém získa osobné údaje používateľa
7. Systém zobrazí osobné údaje používateľa
8. Koniec

### 2.2.2. Návrh

Dátový model: dátový model je zhodný s dátovým modelom z US4-Editácia profilu.

Aktivity diagram:



Obr. 2.3 Activity diagram Prezeranie profilu

Návrh obrazovky:



Obr. 2.4 Navrh obrazovky Prezeranie profilu

### 2.2.3. Implementácia

Pri implementácii bola najdôležitejšia časť kódu v kontroleri *SiteController.php*, kde sa vybrali údaje z databázy:

```
public function actionShowProfil()
{
    $conn = Yii::app()->db;

    $query = $conn->createCommand()
        ->select('login, name, email, mobile, info, gender, work')
        ->from('user')
        ->where('login=\'' . Yii::app()->user->getId() . '\'')
        ->limit(1)
        ->query();

    if($query->getRowCount() != 1)
    {
        throw new CHttpException(404, 'The specified content does not exist.');
```

### 2.2.4. Testovanie

Pre testovanie bol navrhnutý jediný scenár a to zobrazenie existujúcich údajov o používateľovi:

| Názov     | Zobrazenie profilu     | ID Testu |  |
|-----------|------------------------|----------|--|
| Rozhranie | Obrazovka prihláseného | ID UC    |  |

|                           |   |                          |                          |
|---------------------------|---|--------------------------|--------------------------|
|                           | používateľa                                       |                          |                          |
| <b>Účel</b>               | zobrazenie vlastného profilu                      |                          |                          |
| <b>Vstupné podmienky</b>  | prihlásený používateľ                             |                          |                          |
| <b>Výstupné podmienky</b> | Žiadne  |                          |                          |
| <b>Krok</b>               | <b>Akcia</b>                                      | <b>Očakávaná akcia</b>   | <b>Skutočná akcia</b>    |
| 1.                        | V menu kliknem na tlačidlo pre zobrazenie profilu | Zobrazenie môjho profilu | Zobrazenie môjho profilu |

Tab. 2.4 Prezeranie profilu Test1 1

## 2.3. US7 Zobrazenie vlastných znalostí

Používateľ má možnosť prezerat všetky svoje pridané znalosti.

### 2.3.1. Analýza

Každý používateľ by mal mať možnosť prezerania informácií o sebe minimálne kvôli kontrole informácií. Preto je potrebné mu poskytnúť toto zobrazenie.

Zobrazí sa jednoduchý formulár, ktorý prehľadne zobrazí všetky používateľove pridané znalosti.

Vstupné podmienky: prihlásený používateľ

Výstupné podmienky: žiadne

Aktéri: používateľ

Hlavný scenár: 1. Požiadavka na zobrazenie vlastných znalostí  
2. Informatívne zobrazenie vlastných znalostí

### 2.3.2. Návrh

Funkcionalita berie informácie z databázy pomocou asociačných tabuliek používateľa so znalosťou.

Cieľom je vypísať tie znalosti z tabuliek, ktoré sú priradené jednému konkrétnemu používateľovi, v tomto prípade prihlásenému používateľovi.

Aktivity diagram:



Obr. 2.5. Activity diagram Zobrazenie vlastných znalostí

### 2.3.3. Implementácia

Implementácia návrhu pomocou frameworku Yii bola realizovaná pomocou vyhľadávania v databáze znalostí používateľa, a následným výpisom týchto znalostí

```

$sql = "
SELECT
    A.id_skill,
    B.name_skill
FROM
    user_skill as A
INNER JOIN
    skill as B
ON
    A.id_skill = B.id_skill
WHERE
    A.id_user = " . $id_user . "
";
$skills = $connection->createCommand($sql)->query();
$ret = array(); $c = 0;
foreach($skills as $row){
    $ret[$c]['id'] = $row['id_skill'];
    $ret[$c++]['name'] = $row['name_skill'];
}
  
```

### 2.3.4. Akceptačné testy

Pre testovanie bol navrhnutý jediný scenár použitia pre zobrazenie znalostí používateľa.

|                           |  |                                    |                                    |
|---------------------------|--|------------------------------------|------------------------------------|
| <b>Názov</b>              | Zobrazenie znalostí  | ID 1                               |                                    |
| <b>Rozhranie</b>          | Obrazovka formuláru  | ID UC7                             |                                    |
| <b>Účel</b>               | Zobrazenie vlastných znalostí používateľa                    |                                    |                                    |
| <b>Vstupné podmienky</b>  | Prihlásený používateľ  |                                    |                                    |
| <b>Výstupné podmienky</b> | Žiadne   |                                    |                                    |
| <b>Krok</b>               | <b>Akcia</b>   | <b>Očakávaná akcia</b>             | <b>Skutočná akcia</b>              |
| 1.                        | Požiadavka na zobrazenie obrazovky so znalosťami používateľa | Zobrazenie obrazovky so znalosťami | Zobrazenie obrazovky so znalosťami |

Tab. 2.5 Zobrazenie vlastných znalostí test1

## 2.4. US12 Kategorizácia znalostí

Ako používateľ chcem mať k dispozícii kategorizované pojmy v určitej hierarchii opisujúce znalosti, aby som s nimi mohol pracovať.

### 2.4.1. Analýza

Keďže už máme k dispozícii prvotnú bázu pojmov, je nutné pre tieto pojmy spraviť hierarchiu. Táto hierarchia je dôležitá, aby sa vedelo, ktorá znalosť študenta je nadriadená ktorej. Takže ak máme pre pojem ".NET" priradený pojem "C#" znamená to, že pojem "C#" je podriadený. Zase si tieto znalosti netreba mýliť s grupami, ktoré podľa návrhu logického modelu predstavujú pole znalostí. V tomto príklade by grupa mohla byť "Programovanie". Grupy sa budú riešiť v nasledujúcich šprintoch.

### 2.4.2. Návrh

V databáze PostgreSQL je navrhnutá nasledovná tabuľka:

| Názov stĺpca v tabuľke | Typ     | Vysvetlenie             |
|------------------------|---------|-------------------------|
| Id                     | integer | unikátne id položky     |
| id_skill1              | integer | id nadriadenej znalosti |
| id_skill2              | integer | id podriadenej znalosti |

Tab. 2.6 Navrh tabuľky hierarchy\_skills

### 2.4.3.Implementácia

Ako GUI sa použil pgAdmin III. Prepájacia tabuľka sa vytvorila nasledovne:

```
-- Table: hierarchy_skills  
  
CREATE TABLE hierarchy_skills  
(  
    id serial,  
    id_skill1 integer,  
    id_skill2 integer,  
    CONSTRAINT skills_id_key UNIQUE (id)  
)  
WITH (  
    OIDS=FALSE  
);  
ALTER TABLE hierarchy_skills  
    OWNER TO postgres;
```