

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

Obsah

Úvod	4
Účel a rozsah dokumentu.....	4
1. Prvý šprint – Trójska vojna	5
1.1. US1 Návrh šablóny GUI	5
1.1.1. Analýza	5
1.1.2. Návrh	6
1.1.3. Implementácia.....	7
1.1.4. Testovanie	9
1.2. US4 Editácia profilu	9
1.2.1. Analýza	9
1.2.2. Návrh	10
1.2.3. Implementácia.....	11
1.2.4. Testovanie	12
1.3. US 6 – Zadanie znalostí.....	12
1.3.1. Analýza	12
1.3.2. Návrh	13
1.3.3. Implementácia.....	14
1.3.4. Akceptačné testy	14
1.4. US 11 - Vytvorenie prvej bázy pojmov.....	14
1.4.1. Analýza	14
1.4.2. Návrh	14
1.4.3. Implementácia.....	15
2. Druhý šprint – Prvá púnska vojna.....	15
2.1. US2 Prihlasovanie používateľa	15
2.1.1. Analýza	15
2.1.2. Návrh	16
2.1.3. Implementácia.....	17
2.1.4. Testovanie	19
2.2. US3 Prezeranie profilu.....	20
2.2.1. Analýza	20
2.2.2. Návrh	21
2.2.3. Implementácia.....	22
2.2.4. Testovanie	22
2.3. US7 Zobrazenie vlastných znalostí	22

2.3.1.	Analýza	23
2.3.2.	Návrh	23
2.3.3.	Implementácia.....	24
2.3.4.	Akceptačné testy	24
2.4.	US12 Kategorizácia znalostí.....	25
2.4.1.	Analýza	25
2.4.2.	Návrh	25
2.4.3.	Implementácia.....	25

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 vzľad. Dobrý výber farieb je kľúčový a bude systém charakterizovať.

Na obrázku 1.2 je znázorný grafický návrh obrazovky systému.

Znalosť	Kategória	Akcia
Java pokročilý	Programovanie	✎ ✕
C++ začiatočník	Programovanie	✎ ✕

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 zrozumiteľné 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>
</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 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 verzií 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 vyplňanie 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.

Prípady použitia Editácia profilu

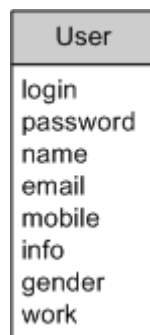
ID	1

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

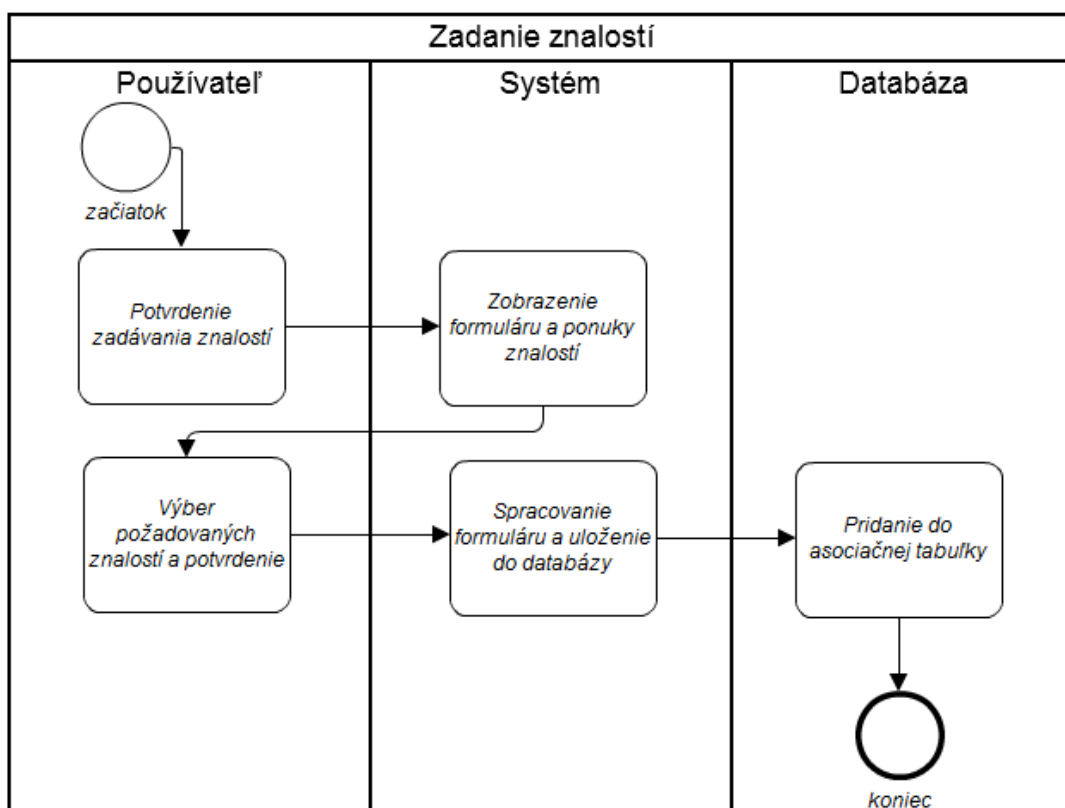
1.3.2.Návrh

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.

Názov	Pridanie znalostí	ID 1	
Rozhranie	Obrazovka formuláru	ID UC6	
Účel	Pridanie nových znalostí do profilu		
Vstupné podmienky	Prihlásený používateľ		
Výstupné podmienky	Žiadne		
Krok	Akcia	Očakávaná akcia	Skutočná akcia
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 tabulky 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

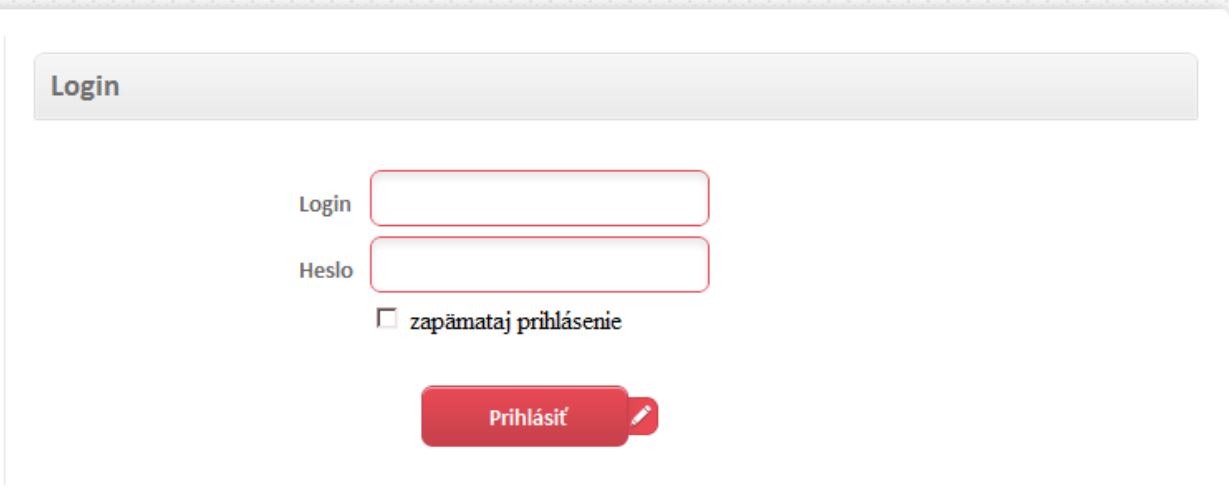
Aktéri: používateľ

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.

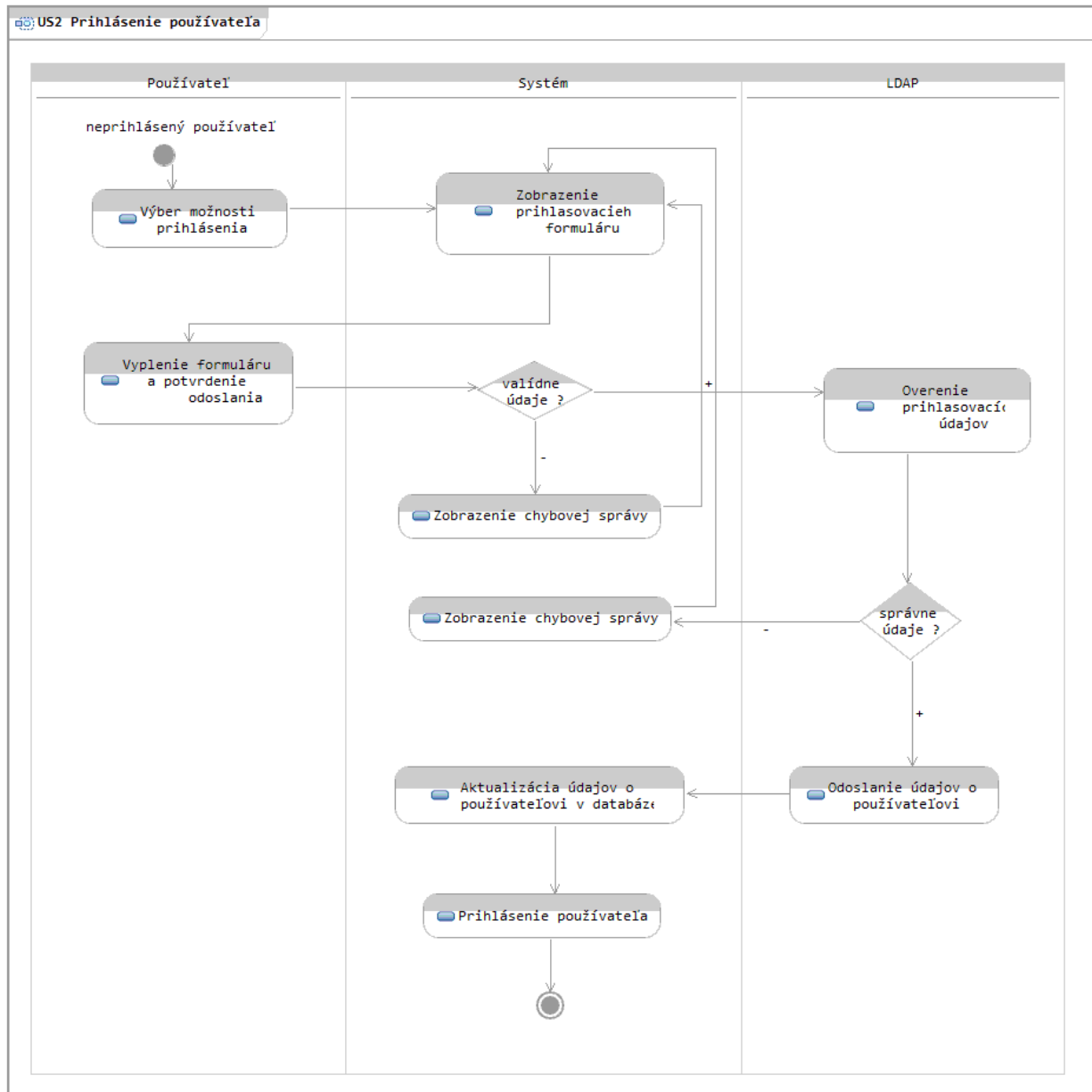


The image shows a login form with the following elements:

- A header bar with the text "Login".
- A label "Login" next to a text input field.
- A label "Heslo" next to a text input field.
- A checkbox with the text "zapamätaj prihlásenie" below the password field.
- A red button with the text "Prihlásiť" and a white 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.

Názov	Úspešné prihlásenie do systému	ID Testu	XX-01
Rozhranie	Obrazovka prihlásenia	ID UC	XX
Účel	Prihlásenie používateľa s prihlasovacími údajmi z AIS		
Vstupné podmienky	Správne prihlasovacie údaje		
Výstupné podmienky	Žiadne		
Krok	Akcia	Očakávaná reakcia	Skutočná akcia
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

Názov	Nevyplnenie poľa "Heslo"		ID Testu	XX-02
Rozhranie	Obrazovka prihlásenia		ID UC	XX
Účel	Zobrazenie chybovej hlášky: "Pole heslo nemôže byť prázdne"			
Vstupné podmienky	Žiadne			
Výstupné podmienky	Žiadne			
Krok	Akcia	Očakávaná reakcia	Skutočná akcia	
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

Názov	Neúspešné prihlásenie do systému.		ID Testu	XX-03
Rozhranie	Obrazovka prihlásenia		ID UC	XX
Účel	Zobrazenie chybovej hlášky: "Zadané zlé prihlasovacie údaje"			
Vstupné podmienky	Nesprávne prihlasovacie údaje			
Výstupné podmienky	Žiadne			
Krok	Akcia	Očakávaná reakcia	Skutočná akcia	
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

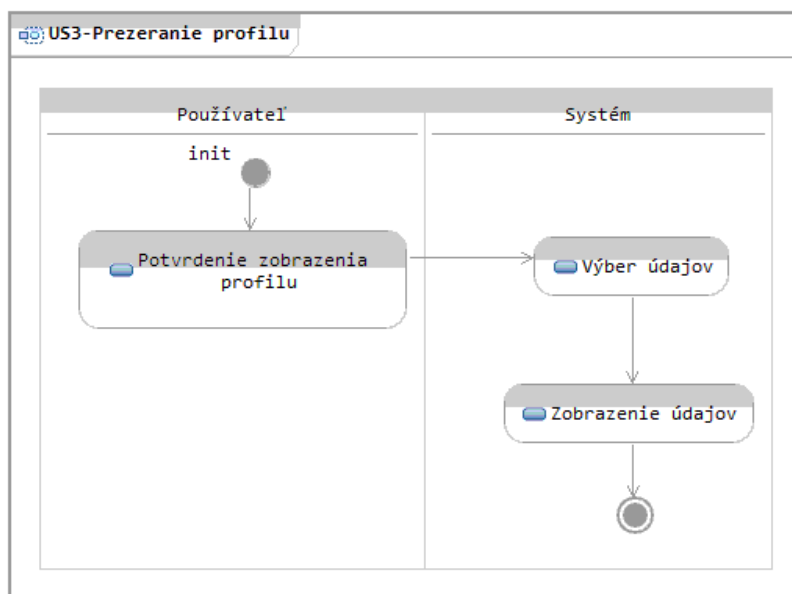
Flow:

1. Používateľ vyberie možnosť zobrazenia vlastného profilu
2. Systém získa osobné údaje používateľa
3. Systém zobrazí osobné údaje používateľa
4. Koniec

2.2.2.Návrh

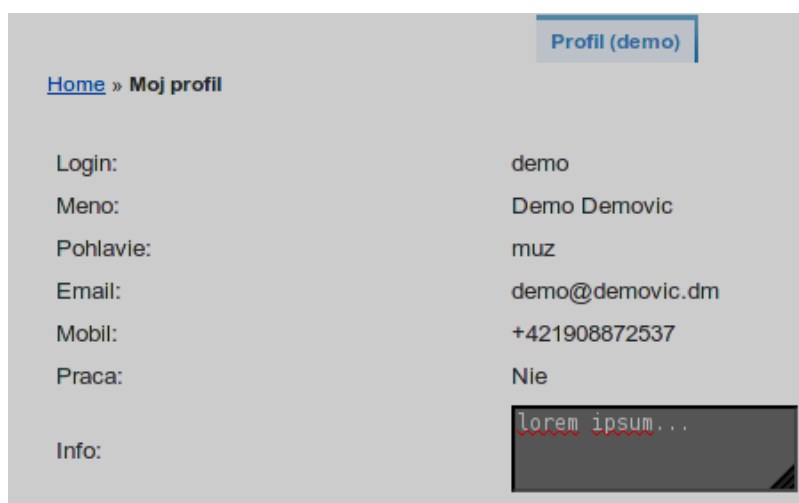
Dátový model: dátový model je zhodný s dátovým modelom z US4-Editácia profile.

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 používateľa	ID UC	
Účel	zobrazenie vlastného profilu		
Vstupné podmínky	prihlásený používateľ		
Výstupné podmínky	Žiadne		
Krok	Akcia	Očakávaná akcia	Skutočná akcia
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.

Názov	Zobrazenie znalostí	ID 1	
Rozhranie	Obrazovka formuláru	ID UC7	
Účel	Zobrazenie vlastných znalostí používateľa		
Vstupné podmienky	Prihlásený používateľ		
Výstupné podmienky	Žiadne		
Krok	Akcia	Očakávaná akcia	Skutočná akcia
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;
```