

Jihočeská univerzita v Českých Budějovicích

Přírodovědecká fakulta



IS PROJEKT

Michal Kolář

České Budějovice 2016

Obsah

| | |
|--|-----------|
| Úvod | 4 |
| Cíle práce | 5 |
| Počáteční postup | 6 |
| 1 Responsivní web design | 7 |
| 1.1 Popis | 7 |
| 1.2 Bootstrap | 7 |
| 2 Backend | 9 |
| 2.1 Symfony | 9 |
| 2.1.1 Databáze | 9 |
| 2.1.2 Šablonovací engine - TWIG | 10 |
| 2.1.3 Administrace - Sonata | 11 |
| 3 Modul zprávy - ukázka | 12 |
| 3.1 Konfigurační soubor administrace | 13 |
| 3.2 Definice položek administračního modulu zprávy | 14 |
| 3.3 Administrační Controller zpráv - createAction | 16 |
| 3.4 TWIG - šablona výpisu zpráv | 17 |
| 3.5 Controller - výpis zprávy | 18 |
| 4 Vysílání | 19 |
| 4.1 Živé vysílání | 19 |
| 4.2 Aktuální záznam a video archiv | 19 |
| 4.3 Nahrávání souborů | 20 |
| 4.4 Miniatura | 20 |
| 4.5 Přehrávání souborů | 20 |
| 4.6 Speciální přístup | 21 |
| 5 Uživatelé | 22 |
| 5.1 Uživatel | 22 |

| | | |
|-----|----------------------------------|-----------|
| 5.2 | Práva | 22 |
| 5.3 | Resetování hesla | 24 |
| 5.4 | Editace osobních údajů | 25 |
| 5.5 | Změna hesla | 25 |
| | Závěr | 26 |

Seznam obrázků

| | | |
|---|--|----|
| 1 | Bootstrap column table | 8 |
| 2 | Symfony | 9 |
| 3 | Zprávy | 12 |
| 4 | Administrační rozhraní pro zprávy | 15 |
| 5 | Usecase diagram uživatelů | 23 |
| 6 | Menu - Vysílání | 24 |
| 7 | Sekvenční diagram resetování hesla | 25 |

Seznam tabulek

Úvod

Cílem IS projektu je dát studentovi návyky a dovednosti týmové práce, které se mu pak hodí do dalších pracovních let. V tomto případě se konkrétně jedná o firemní projekt firmy Arakis Belleville, s.r.o. pro spolek Atia gihara, který potřebuje nový moderní internetový portál pro svoje zákazníky. Důraz se u tohoto webového portálu klade hlavně na jednoduchost a přehlednost bez zbytečných prvků pro zachování intuitivního ovládání. Na projektu se podílejí kromě studenta další tři lidé, kteří dopomáhají hladkému průběhu implementace jednotlivých částí systému. Jedná se o Luděk Jirman (project manager), Jakub Červený (senior programátor), Jakub Gall (grafický návrhář).

Cíle práce

Cílem práce je vytvoření responsivního webového portálu podle grafické předlohy. Hlavní funkce portálu bude nahrávání video záznamů přednášek spolku klientem skrze administrační rozhraní a následné poskytnutí těchto multimediálních souborů platícím uživatelům. Video záznamy se budou uchovávat na serveru zhotovitele, kde budou koexistovat s IS. Předpokládá se, že ke každému záznamu bude potřeba uchovávat název, popis, délka videa a datum nahrání. K uchovávání těchto perzistentních dat bude sloužit databázový server zhotovitele s MySQL databází. Samotný přístup k video souborům bude zakázán a bude řešen pouze skrze controller frameworku, kde bude hlídáno jestli je uživatel přihlášen a má práva na přehrání záznamu. Předpokládá se, že v rámci systému budou existovat čtyři uživatelské role, které budou sloužit k přístupu k jednotlivým sekcím portálu na základě předplaceného tarifu. Předplacený tarif není součástí systému, klient tyto věci rozhoduje autonomně a role bude přidávat manuálně v administračním rozhraní. Role budou pojmenovány podle názvu sekcí, které zpřístupňují. Jedná se o Video archiv, Aktuální záznam, Živé vysílání a Administrace. V rámci administrace bude mít zákazník možnost spravovat uživatele a jejich atributy jako uživatelské jméno, heslo, jméno, příjmení, telefonní číslo, variabilní speciální číslo uživatele. V rámci systému budou mít uživatelé možnost tyto atributy měnit v profilové části portálu. Tato data budou opět perzistentní a ukládaná do databáze. Důležitá součást portálu bude sekce pro poskytování live streamu přednášek, která bude napojena na vzdálené interaktivní rozhraní. V rámci systému budou moci administrátoři posílat jednotlivým uživatelům nebo skupinám soukromé zprávy, které budou mít zároveň možnost posílat i emailem.

Postup

Při setkání s klientem se určily předpokládané funkce systému. Na základě těchto předpokladů byla ve spolupráci s grafikem navržnuta grafická předloha, kterou si zákazník pečlivě prohlídl a odsouhlasil. Předloha byla následně rozřezána v grafickém editoru na menší části, ze kterých byl postupně složen základní design webu - HTML5 šablona.

Mít hotovou šablonu, která se může prezentovat zákazníkovi, se prakticky vždy ukazuje jako dobrý krok, protože v raném vývoji webu je snadné reorganizovat, mazat, měnit jednotlivé prvky, což ve finálním produktu přináší různá úskalí a hlavně práci navíc. Kromě toho zákazník dostane představu, co ve finále uvidí a dostane, byť se jedná o generovaný obsah.

Při vývoji šablony bylo využito potenciálu frontend frameworku Bootstrap, HTML5 a CSS3. Pro další nestandardní prvky, jako posunování obsahu, dynamické schovávání/objevování elementů či asynchronní načítání dat, byla použita JavaScriptová knihovna JQuery. Na překódované předloze se vyzkoušela responzivita webu a interaktivní prvky. Zákazník byl vesměs spokojen a jeho následné designové připomínky byly zapracovány do šablony, čímž se odstartovala další fáze vývoje - backend.

Pro potřeby portálu byl zvolen PHP framework Symfony, který slouží jako mocný nástroj se snadnou integrací základních, vývojáři oblíbených balíčků, které jsou snadno ohybatelné na konkrétní řešení.

Jednotlivé funkce systému byly nejprve vyvíjeny na lokálním serveru a poté byly postupně uvolněny na zkušební server, kde si je mohl prohlédnout zákazník, čímž se zajistila zpětná vazba a zamezilo se tak přebytečné práci spojené s předěláváním hotových modulů, které si zákazník původně představoval jinak.

Jakmile bylo zadání hotové a byly zapracovány zákaznickovy připomínky, založil se produkční server, kam se nahrála čistá verze webu se zapnutým cachováním. Web byl tímto dokončen.

1 Responsivní web design

1.1 Popis

Jedná se o způsob stylování HTML dokumentu, které zaručí, že zobrazení stránky bude optimalizováno pro všechny druhy nejrůznějších zařízení (mobily, notebooky, netbooky, tablety atd.). Především díky vlastnosti Media Queries, která je zahrnuta ve specifikaci CSS3, lze rozpoznat vlastnosti zařízení, na kterém je stránka prohlížena a přizpůsobit tak samotnou stránku a její obsah.

1.2 Bootstrap

Pro usnadnění práce existuje na světě několik frameworků. Z minulých zkušeností byl zvolen free a open-source framework Bootstrap, který je velmi oblíben hlavně kvůli mřížkovému systému dělení celku na menší části. V rámci frameworku jsou například připraveny CSS třídy pro sloupce, kde 100% šířka elementu odpovídá 12 jednotkám. S číslem 12 se mnohem lépe pracuje, neboť jde lépe dělit než třeba číslo 10 nebo 100. Pokud tedy chceme usadit v řádce sloupcové elementy vedle sebe, pak musíme elementům přiřadit třídu s jednotkami o velikosti právě takové, aby daly dohromady 12.

Například pokud chceme zaplnit elementy celou dostupnou šířku kontejneru, musíme zvolit col-md-6 col-md-6 nebo col-md-4, col-md-4, col-md-4 apod. V momentě, kdy se elementy vedle sebe nevejdou, seskládají se pod sebe.

| | | | | | | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| .col-md-1 | .col-md-1 | .col-md-1 | .col-md-1 | .col-md-1 | .col-md-1 | .col-md-1 | .col-md-1 | .col-md-1 | .col-md-1 | .col-md-1 | .col-md-1 |
| .col-md-8 | | | | | | | | .col-md-4 | | | |
| .col-md-4 | | | | .col-md-4 | | | | .col-md-4 | | | |
| .col-md-6 | | | | | | .col-md-6 | | | | | |

Obrázek 1: Bootstrap column table

Kromě tříd col-md-*, existují další verze, které se zapínají podle dostupné šířky displeje:

- col-lg-* pro šířku větší než 1199px
- col-md-* pro šířku větší než 991px
- col-sm-* pro šířku větší než 749px
- col-xs-* pro šířku menší než 750px

Tyto třídy jdou použít souběžně, čímž získáme kontrolu kolik elementů se v řádce objeví v různých rozlišeních.

2 Backend

2.1 Symfony



Obrázek 2: Symfony

Pro vývoj backendu byl vybrán PHP framework Symfony ve verzi 2.8, který je známý pro svou širokou základnu uživatelů, podporou, bohatou dokumentací a hlavně snadnou rozšiřitelností o mnoho dostupných balíčků (bundles), které velmi usnadňují vývoj díky snadné znovupoužitelnosti kódu.

2.1.1 Databáze

Pro ukládání a čtení dat byla použita MySQL databáze, kterou řídí Doctrine ORM balíček cílící na mapování objektů na relační databázi.

2.1.2 Šablonovací engine - TWIG

Součástí standardní verze Symfony je šablonovací engine TWIG, který slouží k vylepšení celkové čitelnosti frontendového kódu a výraznému zjednodušení zápisu.

```
{% for message in messages %}
    {{ message.title }}
{% else %}
    Nebyly nalezeny žádné zprávy.
{% endfor %}
```

vs.

```
<?php
if(count($messages) {
foreach ($messages as $message) {
    echo $message->title;
}
} else {
    echo "Nebyly nalezeny žádné zprávy.";
}
?>
```

TWIG nabízí řadu předpřipravených optimalizovaných filtrů, které dělají operace nad předanou proměnnou. Uvádějí se za proměnnou. Ze základních funkcí použitých v projektu lze třeba uvést:

- truncate - slouží k omezení počtu znaků ve stringu - parametry:
 - 1. počet znaků
 - 2. boolean parametr, jež značí zda-li se má slovo uříznout uprostřed nebo se má zachovat celé

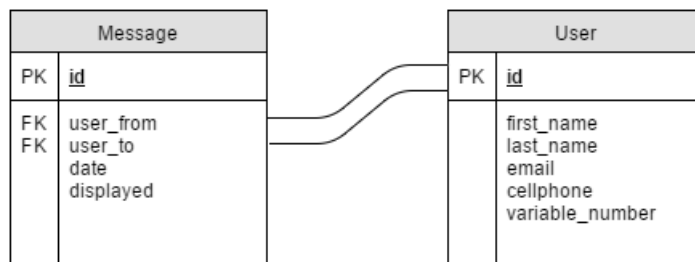
- 3. string, který se má dosadit za ořízlé slovo, pokud je 2. parametr false. Standardně "..."
- escape - projde text a vyescapuje HTML/JS tagy
- replace - projde text a zamění všechny stringy z prvního parametru za string z druhého parametru
- date - formátuje čas na požadovaný výstup
např: `message.date—date('j. n. Y')`
 - j - den bez počáteční nuly
 - n - měsíc bez počáteční nuly
 - Y - rok, plné zobrazení - 4 číslice

2.1.3 Administrace - Sonata

Administrace je postavena na balíku "SonataDoctrinePHPCRAdminBundle", jež nabízí pěkné prostředí a snadnou integraci vlastních modulů skrze konfigurační soubory pro vytváření, upravování, mazání dané entity modulu.

3 Modul zprávy - ukázka

V rámci zpráv bylo potřeba uchovat od koho zpráva přišla, komu patří a kdy byla odeslána. Po komunikaci se zákazníkem byla doplněna položka "přečteno", aby měl přehled, zda si uživatelé zprávy přečetli. Dalším požadavkem byla možnost rozeslat jednu zprávu více uživatelům najednou + možnost odeslání této zprávy emailem.



Obrázek 3: Zprávy

V rámci ukázky jsou zde uvedeny kousky použitého kódu s komentáři v jejich nativní syntaxi, které vysvětlují jednotlivé řádky.

3.1 Konfigurační soubor administrace

```
app.admin.notificationMessage:
#definuje položky
    class: NotificationBundle\Admin\NotificationMessageAdmin

#entita, se kterou se bude pracovat
#controller pro administraci zpráv
    arguments: [
        ~,
        NotificationBundle\Entity\NotificationMessage,
        NotificationBundle:CRUD
    ]

#jméno service
#objektově relační mapování
#skupina Zprávy
#štítek Seznam zpráv
    tags:
        - {
            name: sonata.admin,
            manager_type: orm,
            group: Zprávy,
            label: Seznam zpráv
        }
```

3.2 Definice položek administračního modulu zprávy

```
$formMapper
//přidá multi-select box s výběrem uživatelů
//načtených z metody getUsers()
->add('users', 'choice', array(
    'placeholder' => 'Vyberte příjemce',
    'choices' => $this->getUsers(),
    'multiple' => true,
    'label' => 'Komu',
))

//přidá pole nadpis - plain text
->add('header', 'text', array('label' => 'Nadpis'))

//přidá pole obsah - JavaScriptový html editor - CK Editor
->add('content', 'ckeditor', array('label' => 'Obsah'))

//přidá checkbox, jehož stav odchytáváme na controlleru,
//kde se rozhodne jestli se mu má zpráva odeslat i na email
->add('sendmail', 'checkbox', array(
    'label' => 'Odeslat mail',
    'mapped' => false,
    'required' => false
))
;
```

Seznam zpráv

Komu *

jmeno2 prijmeni2
Lorem Ipsumovič
Lorem Ipsum
user 2

Nadpis *

Obsah *

✂️ 📄 📧 📧 📧 ⬅️ ➡️ ABC 🔗 🗨️ 🚩 🖼️ 📊 📄 Ω 🔄 📄 Zdroj

B *I* **S** U ☰ ☱ ☲ ☳ ☴ ☵ ☶ ☷ ” Styl Formát ?

Odeslat mail

📄 Vytvořit a zpět do seznamu

➕ Vytvořit a přidat další

Obrázek 4: Administrační rozhraní pro zprávy

3.3 Administrační Controller zpráv - createAction

```
//udělá novou instanci podle příchozích dat z formuláře
$object = $this->admin->getNewInstance();
$this->admin->setSubject($object);
$form = $this->admin->getForm();
$form->setData($object);
$em = $this->getDoctrine()->getEntityManager();

//načte vybrané uživatele z odeslaného formuláře
$selectedUsers = $form->get('users')->getData();
foreach ($selectedUsers as $userId)
{
    //dosadí ID přihlášeného admina, co odeslal zprávu
    $object->setUserIdFrom($this->getUser()->getId());
    //dosadí ID uživatele, kterému má zpráva dojít
    $object->setUserIdTo($userId);
    //nastaví příznak přečteno na false
    $object->setDisplayed(false);
    //zapíše aktuální datum
    $object-> setDate(new \DateTime());

    //pokud je checkbox poslat mail zaškrtnuté
    //odešli zprávu na email
    if ($form->get('sendmail')->getData())
        $this->sendMail($object);

    //uloží zprávu do databáze
    $em->persist($object);
    $em->flush();
    $em->clear();
}
```


3.4 TWIG - šablona výpisu zpráv

```
{# pokud existují zprávy #}
{% if messages|length > 0 %}
    {# kontejner pro zprávy #}
    <section class="messages row">
        {# načtení všech zpráv #}
        {% for msg in messages %}
            {# velká rozlišení 3 vedle sebe #}
            {# střední rozlišení 2 vedle sebe #}
            {# mála rozlišení 1 #}
            <article class="col-lg-4 col-md-6 col-xs-12">
                <time>{{ msg.date|date('j. n. Y') }}</time>
                {# pokud zpráva nebyla přečtena, přidej třídu #}
                <header {% if msg.displayed %}
                    class="displayed"{% endif %}>{{ msg.header }}</header>
                <div class="p-container">
                    {# zkrácení obsahu na 150 znaků #}
                    <p>{{ msg.content|truncate(150)|raw }}</p>
                </div>
                {# url na konkrétní zprávu #}
                <a href="{{ msg.url }}" class="more" title="VÍCE"></a>
            </article>
        {% endfor %}
    </section>
{# pokud neexistují žádné zprávy #}
{% else %}
    <div class="simple-notification-page">
        Žádné zprávy k zobrazení
    </div>
{% endif %}
```

3.5 Controller - výpis zprávy

```
/**
 * @Route("/zpravy/zprava/{id}", name="Zpráva")
 */
public function messageAction($id = null) {
    //zjistíme jestli jsme přihlášeni
    $user = $this->getUser();
    if ($user) {
        $em = $this->getDoctrine()->getManager();
        //hledáme zprávu podle user_id a meessage_id
        $params = array();
        $params['userIdTo'] = $user->getId();
        $params['id'] = $id;
        $message =
            $em->getRepository('NotificationBundle:NotificationMessage')
                ->findBy($params);
        //pokud zprávu najdeme, označíme ji "přečteno"
        //předáme jí do šablony
        if (count($message)) {
            $message[0]->setDisplayed(1);
            $em->flush();
            return $this->render(
                'NotificationBundle:Notification:message.html.twig',
                array('message' => $message[0])
            );
        }
    }
    return $this->render(
        'NotificationBundle:Notification:message.html.twig',
        array('message' => NULL));
}
```

4 Vysílání

V rámci kategorie vysílání byly stanoveny tři základní sekce:

- živé vysílání
- aktuální záznam
- video archiv

4.1 Živé vysílání

Podkategorie byla velmi diskutována se zákazníkem, který chtěl poskytovat live stream přednášek. Nakonec bylo v rámci úspor použito dosavadní řešení, které bylo používáno uživateli a do webu bylo tak integrováno pouze jako rámeček (iframe) s vnořeným obsahem.

4.2 Aktuální záznam a video archiv

Tyto sekce slouží k poskytování odvysílaných přednášek. Obě sekce se od sebe v zásadě moc neliší. Aktuální záznam se rozlišuje pouze příznakem a je zpřístupněn pouze lidem s patřičným přístupem. Pro specifické zobrazení používají od sebe lehce modifikovanou šablonu a controller.

Jak již bylo řečeno, jedná se o web s moderními technologiemi, tudíž nebylo potřeba použít pro přehrávání videa externí přehrávač, neboť drtivá většina nových prohlížečů nativně podporuje přehrávání multimediálních souborů. Vzhledem k nejlepší kompatibilitě bylo doporučeno používat formát MP4, který již několik let podporuje majorita prohlížečů.

Samotná entita záznamu se skládá z:

- název
- popis
- délka
- datum
- miniatura - název obrázku s miniaturou
- povoleno - příznak rozlišující jestli má být video na webu aktivní
- aktuální záznam - příznak značící aktuální záznam

4.3 Nahrávání souborů

Problém v této disciplíně je, že při velkých souborech například větších než 1GB, můžeme narazit na problém ze strany prohlížečů, které skrze HTTP POST metodu dosáhnou svého limitu a více dat poslat nedokáží. Toto je pro účely portálu neakceptovatelné, a tak je tento problém řešen skrze posílání souborů po částech a jejich následné skládání na straně serveru opět v jeden celek.

4.4 Miniatura

Po úspěšném nahrání souboru se v administraci připne toto video do připraveného video tagu. Za pomoci javascriptu promítáme obsah videa do grafického elementu canvas. Po stisknutí tlačítka k sejmutí miniatury dojde k převedení obrazu canvasu do formy base64 a následně je asynchronně odeslán na backendový controller, kde je zpět konvertován na obrázek a uložen. Pokud nedojde k sejmutí miniatury, udělá se automaticky z prvního snímku.

4.5 Přehrávání souborů

Video záznam nelze jednoduše vnořit do šablony z důvodu zabezpečení. Při absolutním odkazu na soubor by si jej totiž mohl přehrát každý. Proto bylo v tomto

řešení využito controlleru hlídajícího, zda-li má uživatel patřičná práva pro přehrání záznamu. S tímto řešením ovšem nastávají podobné komplikace jako při nahrávání souborů, kdy je nutné video kouskovat a podle potřeby ho posílat po menších částech.

4.6 Speciální přístup

Tento modul v administraci slouží k přístupnosti libovolného videa, ať už je povolené nebo ne. Myšlenka za tímto přístupem je taková, že administrátor potřebuje na žádost zpřístupnit video pouze konkrétnímu uživateli, který měl například dovolenou a nestihl si tak pustit aktuální záznam. Pokud uživatel dostane přiřazený záznam, uvidí ho v sekci video archiv pod nadpisem "speciální přístup".

5 Uživatelé

Jako základ pro uchování údajů posloužila komponenta "FOSUserBundle", která poskytuje zabezpečený flexibilní framework pro načítání uživatelů z databáze.

Uživatele si zákazník přál zadávat skrze administraci sám, čímž odpadla celá sekce registrace.

5.1 Uživatel

K základní entitě user obsahující věci jako heslo, sůl, uživatelské jméno atd. byly přidány atributy:

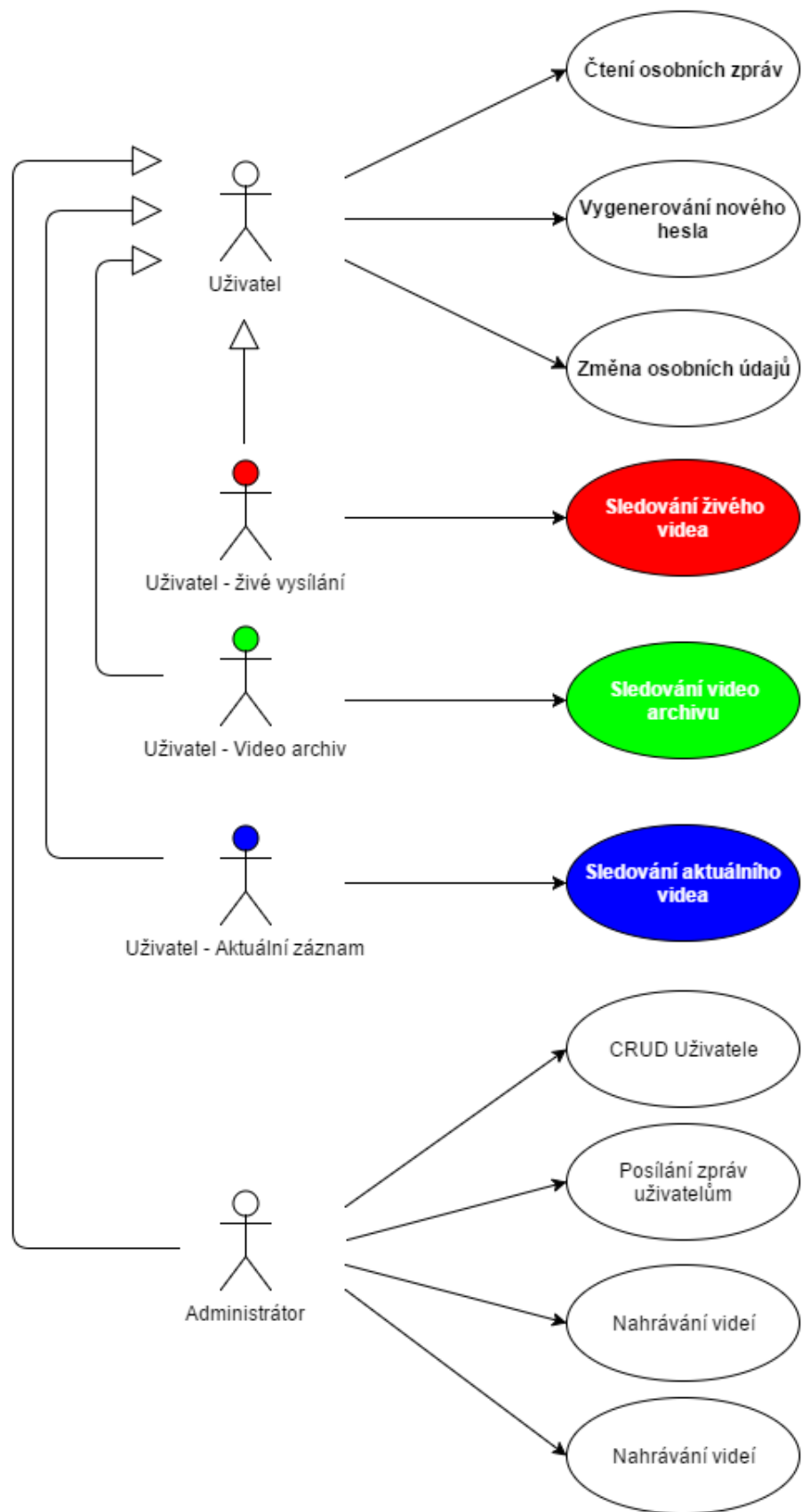
- jméno
- příjmení
- email
- telefon
- variabilní číslo - pro účely zákazníka

5.2 Práva

Balíček "FOSUserBundle" umožňuje asociovat skupiny s uživateli. Skupiny zahrnují jednotlivé role v systému. Navolené role ve skupině budou uděleny všem uživatelům skupiny.

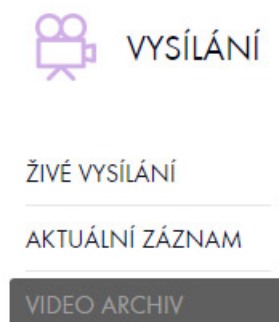
Pro primární účely tohoto webu byly přidány tři role:

- živé vysílání
- aktuální záznam
- video archiv



Obrázek 5: Usecase diagram uživatelů

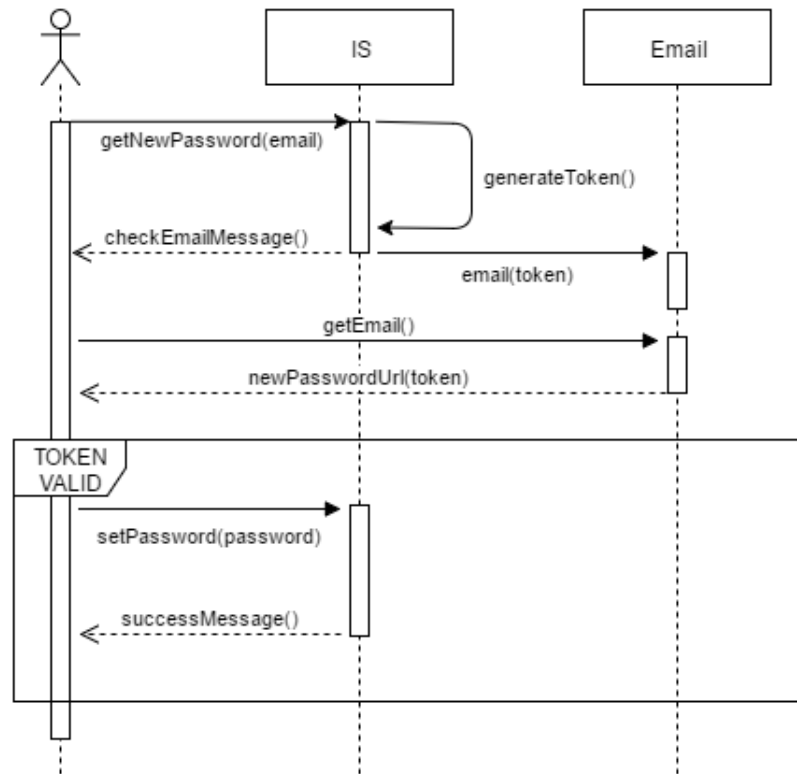
Uživatel, kterému chybí například role video archiv, tak uvidí tuto položku neaktivní. Pokud by se pokusil přistoupit do archivu, tak ho systém odmítne s oznámením o nepovoleném přístupu. Obdobně to platí i o přímých odkazech na video, kde se o stream stará controller kontrolující právě tyto vypsaná práva.



Obrázek 6: Menu - Vysílání

5.3 Resetování hesla

Uživatel má možnost resetovat své heslo skrze svůj osobní email. Pokud dojde ke ztrátě hesla, může se skrz umístěný odkaz u přihlašovací obrazovky dostat na stránku "Zapomenuté heslo", kde zadá svůj email. Když systém nalezne tento email v databázi, tak na něj pošle odkaz s vygenerovaným tokenem, který je platný po dobu pěti hodin. Pokud uživatel na tuto url přejde a pokud je token validní, tak mu systém nabídne zadat nové heslo.



Obrázek 7: Sekvenční diagram resetování hesla

5.4 Editace osobních údajů

Sekce nabízí k editaci veškeré atributy kromě skupin/rolí a variabilního symbolu. Tyto věci může měnit pouze administrátor. Při měnění osobních údajů musí uživatel zadat platné heslo, jinak nedojde k žádné změně.

5.5 Změna hesla

Skládá se ze tří kolonek. Uživatel musí uvést do první stávající heslo a poté musí dvakrát zadat nové heslo. Hesla nejsou čitelná a aby tak nedošlo k nechtěnému překlepu, musí se zadat dvakrát a shodně, jinak se proces opakuje.

Závěr

K samotnému počátku programování frontend části podle grafické předlohy jsem se dostal 1. listopadu 2015 a portál jsem kompletně dokončil 1. ledna 2016.

Před vývojem backendové části mi bylo nařízeno, ať použiji PHP framework Symfony, se kterým jsem se měl během vývoje seznámit. Během programování jsem se díky nestandardním požadavkům byl nucen hlouběji naučit ovládat a používat tento framework, což bohužel nešlo moc dobře neboť v tu dobu nikdo ve firmě neměl moc zkušeností a teprve se s ním začínalo. Proto se vývoj relativně jednoduchého webu o něco málo zpozdil, nicméně je to nástroj velmi mocný a jakmile člověk pochytí základy, tak to jde od ruky. Kdybych měl podobný projekt dělat znovu, tak bych ho zcela určitě zvládl za poloviční čas. Jsem si jistý, že po tomto frameworku určitě ještě v budoucnu sáhnu.

Zákazník byl s prací spokojen. Chválil si postupné uvolňování nových verzí jednotlivých modulů, které s ním byly diskutovány, což mělo za důsledek vytvoření systému, který přesně požadoval.