Mendelova univerzita v Brně
Provozně ekonomická fakulta

Redakční systém pro web nezávislých herních vývojářů

Vedoucí práce:
Ing. Pavel Turčínek Ph.D.

Jan Richter

Brno, 2017

Rád bych touto cestou poděkoval Ing. Pavlu Turčínkovi Ph.D. za odborné vedení, přínosné konzultace a podnětné rady k bakalářské práci. Dále bych rád poděkoval lidem ochotným věnovat svůj čas uživatelským testům redakčního systému a svým rodičům za emoční i finanční podporu při celém vysokoškolském studiu, jehož výstupem je právě i tato práce.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Redakční systém pro web nezávislých herních vývojářů**
vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací.*
Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.
Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 8. května 2017 .....

Abstract

Richter, J. Content management system for independent game developers' web. Bachelor thesis. Brno: Mendel University, 2017.

Bachelor thesis describes basic knowledge of content management systems with use of PHP frameworks. Then it designs content management system for independent game developers' web through diagrams. Main part is the description of the content management system's implementation, followed by testing from perspective of installing and content managing. In the discussion thesis evaluates the use of frameworks and examines its advantages and disadvantages.

Key words

Laravel Framework, MySQL, CRM, independent game developers, HTML, ERD

Abstrakt

Richter, J. Redakční systém pro web nezávislých herních vývojářů. Bakalářská práce. Brno: Mendelova univerzita v Brně, 2017.

Bakalářská práce popisuje základní přehled vývoje redakčních systémů s použitím PHP frameworků. Dále navrhuje redakční systém vhodný pro potřeby nezávislých herních vývojářů prostřednictvím diagramů. Hlavní částí práce je popis implementace redakčního systému s následným testováním z pohledu instalace i správy obsahu. V diskuzi práce zhodnocuje použití frameworků a rozebírá jejich výhody a nevýhody.

Klíčová slova

Laravel Framework, MySQL, redakční systém, nezávislí herní vývojáři, HTML, ERD

Obsah

1 Úvod

Redakčních systémů je na trhu v současné době mnoho, většina z nich však tvoří obecný základ bez většího zaměření (nabízí uživatelský systém a základní správu příspěvků). Ve velkém množství případů je vždy nutné takové redakční systémy upravovat, ať už přepisováním kódu webu nebo přidáváním velkého množství pluginů, které pak snižují i celkové zabezpečení webu (např. blogovací systém Wordpress je negativně známý svým svými průběžně odhalenými děrami v rozsáhlém zdrojovém kódu i v populárních pluginech).

Dalším problémem je samotný postoj nezávislých herních vývojářů k jejich webové prezentaci – většina takových vývojářů se snaží udržet nejnižší možný rozpočet v rámci svých možností, což ale končí tím, že právě použijí jen základní blogovací systémy a podstatné prvky prezentace si musí dodělávat sami nebo je jednoduše vynechají a raději se více věnuji svým herním projektům, web je pak ale nepřehledný jak pro potenciální zákazníky (hráče), tak i herní redaktory (existují případy, kdy nezávislí herní vývojáři na svůj jednoduchý web zapomněli vložit kontaktní formulář nebo alespoň přímý osobní kontakt). V horším případě web dokonce vytvoří ve stylu připomínajícím webové stránky 90. let minulého století, tím se myslí stránky s minimální grafickou úpravou vytvořenými v programech jako Microsoft FrontPage, výsledně připomínající spíše klasickou stránku určenou pro tisk.

Právě tato zkušenost při navštěvování webových stránek nezávislých herních vývojářů byla motivací k vytvoření redakčního systému, který spolehlivě uspokojí veškeré základní potřeby, které může menší a nezávislé herní studio po své webové prezentaci požadovat.

1.1 Cíl práce

Cílem práce je vytvoření redakčního systému pro nezávislé herní vývojáře (tím jsou myšleny menší studia, která nejsou ve velkém rozsahu financována třetími stranami, tedy studia pracující s minimálním rozpočtem, obvykle na základě vlastního zájmu vytvořit kvalitní hru i za cenu obrovského množství času s pravděpodobně minimální finanční odměnou), který bude splňovat typické požadavky takových vývojářů.

Základním typickým prvkem, který obsahuje většina webů herních studií, je vývojářský blog (dále devblog). Odůvodnění toho, proč má vývojář zájem psát blog o své práci, lze vysvětlit poznáním, že velké množství zákazníků (hráčů) teprve rozpracovaných her (alpha, beta nebo early access stádia hry) mají zájem sledovat poznámky vývojářů k vývoji a veškeré nové informace o hře – na tento požadavek právě vývojáři reagují prostřednictvím svého blogu. Zároveň se tak samozřejmě i přiblíží ke svým hráčům a potenciálně mají mnohem bližší přístup ke zpětné vazbě na stav hry z pohledu samotných hráčů.

Další prvek redakčního systému představuje prostor k prezentaci her vývojářů, tedy stránka, která dokáže přehledně vypsat hry studia a po rozkliknutí k ním ukázat

veškeré podrobné informace, videa ze hry a odkazy na obchody, kde lze danou hru zakoupit (popř. pokud je hra dostupná zdarma, tak kde ji lze stáhnout).

Vypsané předchozí prvky by určitě šly za nepříliš velkého množství času zařídit s určitým kompromisem i přednastavenými redakčními systémy, nicméně chybí stále jedna důležitá část, kterou studia zpravidla potřebují – přehledné a kvalitní prostředí určené herním redaktorům. PR je aktuálně pro herní studia velmi důležité i vzhledem k tomu, že např. na herní platformě Steam v roce 2016 vyšlo více jak 4000 her (Galyonkin, 2016), mezi kterými se jeden titul velmi lehce ztratí, pokud na něj herní média neupozorní (samotná existence her na platformě Steam již neznamená „vstupenku do světa známých vývojářů“). Proto je cílem na redakčním systému také vytvořit presskit, tj. přehledný soupis s veškerými informacemi o samotném studiu a jejich hrách, a systém žádostí o recenzentské kopie her.

Od výsledné práce se očekává plně funkční redakční systém splňující výše zmíněné podmínky (přítomnost funkčního vývojářského blogu, prostoru pro prezentaci tvorby nezávislého herního studia a prostředí určené pro herní redaktory – presskit a vyřizování žádostí o recenzentské kopie her), chybět bude pouze unikátní webový design – od každého zákazníka redakčního systému se očekává, že si zajistí vlastní design pro zachování unikátní webové prezentace.

2 Metodika

Postup plánování a samotné tvorby již začal u cíle práce zmíněném v úvodní kapitole na předcházejících stranách. Tento teoretický náčrt situace a představy výsledku doplňuje rešerše zpracovaná v následující kapitole. Ta se zaměří jak na obecný aktuální stav webových technologií (HTML, PHP, frameworky), tak i na redakční systémy, a to jak z obecného z hlediska (aby bylo ujasněno, co se za redakční systém považuje), tak i z praktického hlediska – jaké jsou aktuální populární redakční systémy a do jaké míry splňují požadavky na webovou prezentaci nezávislých herních vývojářů.

Následným krokem je výběr vhodného softwarového řešení pro tvorbu redakčního systému, a to jak z pohledu návrhu, tak i programování a výsledné implementace. Kapitola Materiál a metody pojímá pohled na základní použitý aplikační software při práci a zjednodušeně přibližuje i použití PHP frameworku Laravel a aplikačních nadstaveb, které budou v redakčním systému aplikovány.

Po takové teoretické přípravě už zbývá jen poslední starost před programováním samotného systému, a to tvorba kompletního návrhu funkčnosti systému z pohledu případů užití (UML use case diagram), kde se vyjasní i možné role uživatelů webu, a z pohledu datového, resp. databázového (ERD). Tato část práce se nachází v kapitole Návrh.

Poté už nebrání nic tomu se začít věnovat programování redakčního systému, které je v kapitole Výsledky rozděleno do několika souvislých bloků vzhledem k několika nezávislým strukturám databáze pro svůj specifický účel (devblog, herní databáze a globální databáze).

Po zhotovení produktu, který lze předběžně považovat za plně použitelný ke svému účelu, dojde k poslední fázi samotné tvorby systému, a to k jeho testování. K tomu dojde ze dvou naprosto odlišných pohledů – administrátorského a uživatelského. Administrátorský test se zaměří na instalaci redakčního systému na několik webových hostingů a vyřeší problémy, které mohou někteří poskytovatelé webového hostingu vykazovat. Uživatelský test předloží několika náhodně vybraným osobám (testerům) přihlašovací údaje k administračnímu uživatelskému účtu, zadá několik základních úloh, se kterými se může typický administrátor redakčního systému setkat, a bude sledovat kvalitu (přehlednost a funkčnost) celkového návrhu administračního prostředí. Veškerá zpětná vazba testerů bude brána v potaz a bude v celé kapitole Testování i rozebírána.

Dále proběhne diskuze o kladech a záporech frameworků, o WYSIWYG editoru a o hostingových službách dostupných zdarma. V závěru poté budou zhodnoceny dosažené výsledky celé práce.

3 Přehled literatury

3.1 Webová stránka, HTML a PHP

Webová stránka je dokument, který je zpracován tak, aby mohl být otevřen ve webovém prohlížeči (dnes jak na PC, tak na mobilních zařízeních). Struktura webové stránky je zpravidla dána značkovacím jazykem HTML, grafické prvky jsou dány jazykem CSS. Ve webovém prostředí se obvykle používá i skriptovací jazyk JavaScript (na straně uživatele) a skriptovací jazyk PHP (na straně serveru). Kombinace výše zmíněných jazyků tvoří typický web.

Webové stránky však mají své nedostatky, se kterými se již mnozí vývojáři smířili. Jedná se o velké množství externího obsahu na každém webu, různé zpracovávání stránek jednotlivými prohlížeči (zprávy typu: „Tato stránka je optimalizována pro Google Chrome“) nebo vynucování zjevně méně přehlednější struktury značek pro validitu webu (Henick, 2010, s. 272–276).

Aktuální verze HTML, HTML5, je velmi rozdílná proti předchozím. Práci s grafickými prvky přenechává CSS (kaskádovým stylům), namísto toho nově HTML5 podporuje např. přehrávání videa i bez dalších pluginů jako Adobe Flash (*W3schools*, ©2016). Problém v rámci HTML5 nastává mezi vývojáři, zákazníky i některými tvůrci, protože mnozí začínají mít problém rozeznat, co doopravdy do HTML patří a co nikoliv – někteří prvky CSS považují za HTML (což není správné) a prohlížeče se stále předhání v tom, kolik „HTML5 prvků“ budou umět (Hogan, 2011, s. 15).

PHP je skriptovací jazyk, který se zpravidla používá na straně serveru (např. Apache web server nebo IIS) pro dynamické generování HTML dokumentů – samotný uživatel nevidí žádný PHP kód, získává pouze výsledek práce skriptů na serveru. PHP se často kombinuje s databázovými systémy, jako jsou např. MySQL či PostgreSQL. PHP aktuálně patří mezi nejpopulárnější serverové skriptovací jazyky, používá ho např. Facebook a Wikipedia (MacIntyre, 2010, s. 2–3).

3.2 PHP frameworky

PHP frameworky přináší rozšířenou funkcionalitu pro webové stránky, odlehčují práci programátorovi za cenu toho, že se musí naučit pracovat se specifickou syntaxí vybraného frameworku. Mezi známé a populární PHP frameworky patří kupříkladu Symfony, Laravel, ZendPHP, YiiFramework, CodeIgniter nebo Nette.

Nette je český open-source framework vyvíjený od roku 2008, který je chválený za svůj výkon, ladící nástroje (Tracy) a kvalitní šablonovací systém Latte. (*Nette*, ©2017) Ačkoliv je na oficiálních stránkách framework chválený za nejaktivnější komunitu v ČR, vývojář při hledání řešení problémů po diskuzních fórech mnohdy nemusí dojít k úspěchu (obzvlášť krizový troubleshooting s Nette čeká každého programátora, který nezvládá český jazyk).

Laravel se prezentuje svým jednoduchým routingem, kvalitní autentizací, spolehlivým mapováním databázových prvků do kódu (Eloquent ORM), šablonovým

systémem Blade a prací s databází rozšířenou o migrace (verzování databáze) (*Laravel*, ©2017a). Další výhodou je i délka podpory jedné verze frameworku – opravy funkčních chyb jsou zaručeny na období dvou let od vydání verze, opravy bezpečnostních chyb jsou zaručeny až na tři roky od vydání (*Laravel*, ©2017b). Právě z těchto důvodů, společně se zájmem autora poznat nový a aktuálně populární framework, byl tento PHP framework zvolen za ideální pro tvorbu redakčního systému.

3.3 Redakční systémy

Redakční systémy, nebo také systémy pro správu obsahu (CMS, resp. WCM pro prostředí webu), zjednodušují práci s obsahem webu, celkově nabízí přehlednou administraci webu i těm, kteří neovládají práci se samotným kódem webu. (Johnston, 2011)

Každý redakční systém má své dvě „tváře“ – jednu pro obyčejného návštěvníka, druhou pro správce (to může být někdo jiný než člověk, který systém na web instaloval). Vzhledem ke své interaktivitě redakční systémy zpracovávají požadavky jinak, v URL se obvykle neodkazuje na přímý soubor na serveru, ale URL se posílá routovacímu souboru, který sám zajišťuje, které části systému mají pracovat na výstupu pro uživatele (co uživatel dostane na výstupu). (Verens, 2010, s.8)

3.4 Existující redakční systémy

Drupal je content-management framework, nicméně se používá s moduly právě jako redakční systém. Prezentuje se jednoduchou správou obsahu, stabilním výkonem a bezpečností. Celý systém je postavený na sestavování z balíčků modulů a vzhledů (themes) (*Drupal*, ©2016). Trpí však velkou spotřebou paměti (není tak vhodný pro instalace na menší a levné webové hostingy), chabou zpětnou podporou starších verzí a často velmi zastarávajícími moduly, které potřebují před implementací dodatečně upravit, popř. v ideálním případě kompletně přepsat pro plnohodnotnou použitelnost v aktuálních verzích Drupalu (Locke, 2011).

PHP-Fusion je redakční systém, který se chválí hlavně jednoduchým přizpůsobením, rychlostí, bezpečností a lokalizací (*PHP-Fusion*, ©2017). Své nejlepší období ale zažil před více než deseti lety a aktuálně již lze tvrdit, že spíše technologicky zaostává a většina webových administrátorů raději volí modernější řešení redakčních systémů, které se na trhu objevují.

Wordpress, založený v roce 2003, byl původně pouze blogovací systém, nicméně v průběhu let se vyvinul v plnohodnotný redakční systém díky velkému množství pluginů a widgetů, které jsou k dispozici, obvykle zdarma (*WordPress*, ©2016). Velkým problémem Wordpressu je jeho obrovský zdrojový kód, chaotické konvence pro pojmenovávání metod a proměnných a hlavně děravé (nedostatečně zabezpečené) doplňky (funkční – pluginy, i vzhledové – themes) (Pataki, 2016).

Všechny výše zmíněné redakční systémy mají jednu podstatnou věc společnou – jsou obecné, tedy nemají žádné přímé zaměření, zpravidla jen opravdu poskytují

základní content management (skvělé např. pro čistý blog). To vede zájemce o více specifický web k používání většího množství pluginů, popř. programování vlastních.

4 Materiál a metody

Vzhledem k vlastním zkušenostem i popularitě kombinace základních prostředků pro tvorbu webových stránek bylo rozhodnuto neodbočovat ze zaběhlých konvencí, proto základní strukturu webu tvoří HTML na uživatelské straně a PHP kód na straně serverové. Databáze je řešená pomocí MySQL, resp. její vývojovou odnoží MariaDB, a výchozím formátem je úložiště MyISAM.

Pro základní vzhledové prvky jsou použity kaskádové styly (CSS) a javascriptová knihovna jQuery. Tyto prostředky jsou však (mimo Bootstrap) použity v malém množství, pouze v nutných případech. To odpovídá cílu práce, kdy o výsledný webový design se postará až skutečný administrátor redakčního systému, který ho bude implementovat ve svém vlastním zájmu pro svůj projekt (své herní studio a své hry).

4.1 Základní software

Přehledná tvorba UML diagramů je zajištěna programem Visual Paradigm (získaný se studentskou licencí). Ten zvládá velké množství diagramů, např. diagram tříd, objektový diagram, časový diagram, stavový diagram či use case diagram. Právě ten je vytvořen a použit k popisu případů užití všech uživatelů redakčního systému.

Pro návrh databáze je použit software MySQL Workbench, kvalitní a volně dostupný nástroj pro kompletní návrh relačních SQL databází se schopností exportovat celý návrh přímo na umístěný databázový server. MySQL Workbench je relativně nový software, považuje se za nástupce dříve populárního programu DBDesigner.

Jako vývojové prostředí byl zvolen PHPStorm. Jedná se o profesionální nástroj pro programátory s tím odpovídající cenou produktu, ke tvorbě kódu celé této práce však bylo využito studentské jednoroční licence. PHPStorm nabízí mnoho funkcí klíčových pro moderní tvorbu webových stránek, mimo klasické systémy napovídání a mnoho maličkostí ulehčujících celkový proces programování nabízí i příkazový řádek, přes který lze ovládat PHP Composer (utilita pro správu závislostí) i PHP Artisan (CLI nástroj, který používá PHP framework Laravel). Nevýhodou tohoto prostředí by mohla být implicitní nutnost programovat v anglickém rozložení klávesnice vzhledem k nastavení klávesových zkratek, pro některé začínající programátory se může naopak jednat o výzvu na takové rozložení přejít.

4.2 PHP framework Laravel

PHP framework Laravel pracuje s architekturou MVC (Model-view-controller) – tedy celková struktura souborů i kódu je rozdělena do několika částí, které lze obecně nazvat modelem (datová část kódu), view (kód tvořící výstup pro uživatele) a controllerem (řídící logika aplikace).

Pro přístup na takový web je nutné (přes soubor .htaccess) zablokovat přímý přístup k jednotlivým souborům na webovém serveru a namísto toho vyřešit veškerou komunikaci s uživatelem webu tzv. routingem – tím se myslí správné propojení

a spolupráce všech klíčových souborů architektury k vrácení požadované části webu. V případě frameworku Laravel je routing řešený v jednom souboru (routing.php), kde se nachází jednotlivé cesty webem – programátor specifikuje typ požadavku (GET, POST, DELETE), URL adresu, na které se takový požadavek sleduje, a controller, který na takový požadavek bude reagovat.

Controller si na základě požadavku získá veškerá potřebná data z databáze (skrze model) a poté data přepošle na view – u frameworku Laravel je view tvořen souborem psaným v šablonovém systému Blade, prakticky se však jedná pouze o HTML rozšířené o speciální závorky vkládající výstup z controlleru. Do těchto závorek lze umístit i další PHP logiku, nicméně pokud to není vyloženě nutné, nejedná se o nejlepší přístup v rámci MVC architektury – ideální je vždy veškerou řídící logiku řešit již v controlleru a view použít pouze jako výstup na obrazovku uživatele.

Laravel aktivně propaguje mezi své programátory svůj ORM systém Eloquent. Ten slouží k mnohem jednoduššímu a příjemnějšímu přístupu k databázi bez nutnosti psát databázové dotazy v SQL syntaxi. V praxi činnost programátora opravdu ulehčuje, ale spíše jen v případech méně složitých požadavků (najdou se případy, kdy pro programátora je nakonec opravdu jednodušší nakonec sepsat databázový dotaz v klasické SQL syntaxi).

ORM, neboli objektově relační mapování, se nejvíce projevuje ve struktuře modelů frameworku Laravel. Ty mají pevně danou strukturu (i po stránce názvu je např. požadováno, aby model měl název v jednotném čísle) a obsahují pouze název primárního klíče tabulky (pokud je jiná než defaultní pro Laravel), dobrovolné prvky (např. bool proměnná určující, zdali se do tabulky budou ukládat časové atributy created_at a updated_at, které Laravel vytváří implicitně) a samotné vazby na jiné tabulky v databázi. Tyto vazby jsou dány velmi jednoduchou jednořádkovou metodou, která je jen tvořena odkazem na cizí klíč, názvem modelu a návratovou metodou udávající typ vazby. V případě 1:n vazby se používají metody `belongsTo()` a `hasMany()`, v případě m:n vazby se používají metody `belongsToMany()` z obou stran vztahu (samotná spojovací tabulka nemá žádnou modelovou reprezentaci v kódu, nepotřebuje ji).

Dále Laravel nabízí vlastního CLI (command-line interface, příkazový řádek) klienta s názvem PHP Artisan. Ten podporuje příkazy pro spuštění webového serveru s Laravelem (`artisan serve`), aplikaci šablon do projektu (např. vzorová autentizace uživatele) a vytváření controllerů a modelů (`artisan make:controller název`).

4.3 Bootstrap

Bootstrap je základní sada volně dostupných vzhledových prvků ve formě kaskádových stylů a javascriptových tříd ulehčujících základní grafický návrh webu. Pro finální implementaci reálného webu není příliš vhodné, aby web spoléhal jen na Bootstrap (weby by tak nebyly vzhledově unikátní a tak ani příliš poutavé pro zkušeného

uživatele), nicméně ve fázi návrhu, programování a distribuce redakčního systému se jedná o skvělou příležitost, jak rozvrhnout vzhledové prvky i s minimální prací s kaskádovými styly. I právě díky Bootstrapu je redakční systém už od začátku tvorby responsivní.

Použití Bootstrapu na webu samozřejmě má i své nevýhody, hlavně jde o zatížení uživatelské strany, po které se požaduje načtení velkého množství CSS a JS tříd (i těch nevyužitých), a méně či více nápadné omezení kreativity webových designerů – na moderních webech postavených na Bootstrapu pozorný člověk může spatřit podobné styly konstrukce určitých prvků webu. (Sandu, ©2017)

4.4 WYSIWYG editor TinyMCE

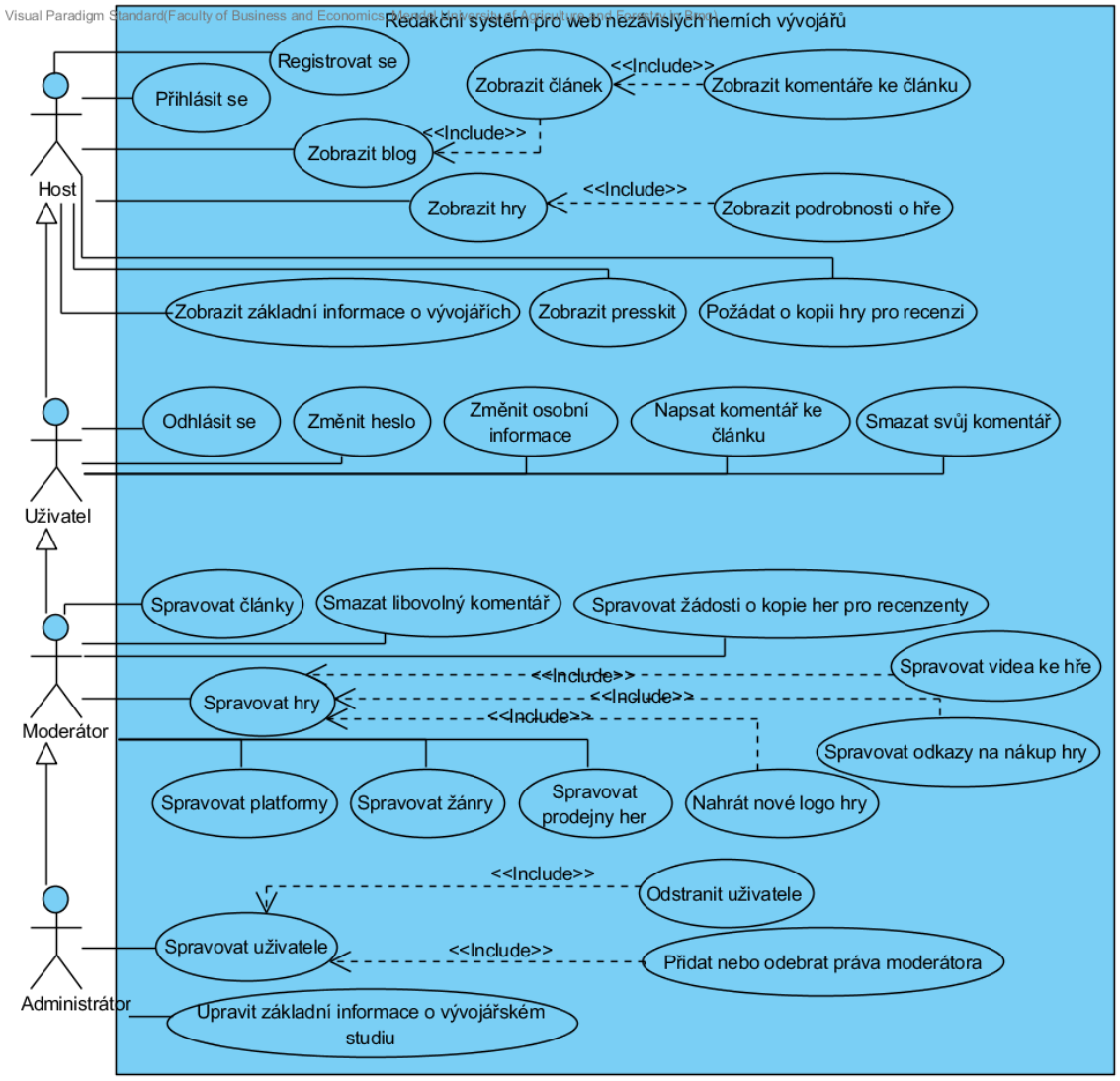
TinyMCE je WYSIWYG (what you see is what you get – co uživatel vidí na vstupu, dostane i na výstupu) editor napsaný v JavaScriptu, šířený jako open-source software. Již v základním nastavení nabízí jednoduché zpracování úseků textu (změny velikosti a řezu písma) či vytváření tabulek a seznamů.

Jedna z nadstaveb TinyMCE je i podpora obrázků, velmi jednoduše zpracovaná pro vkládání obrázků do textu z odkazu, hůře už v případě, že programátor chce docílit možnosti nahrávání obrázků přímo do souborového systému webu (o problémech s nahráváním obrázků se lze dočíst více v kapitole Výsledky, v podsekci věnované implementaci TinyMCE do administrační části redakčního systému).

5 Návrh

5.1 Případy užití – Use case diagram

Redakční systém má celkem čtyři aktory, a to hosta, uživatele, moderátora a administrátora. Celý zjednodušený use case diagram lze vidět níže na obrázku 1. Samotné scénáře případů užití jednotlivých aktorů jsou popsány v následujících podsekcích.



Obrázek 1: Diagram případů užití redakčního systému

5.1.1 Případy užití aktora host

Za hosta se považuje neregistrovaný nebo nepřihlášený návštěvník webové stránky. Mezi jeho základní případy užití patří právě registrace (vytvoření uživatelského

účtu, vyžaduje unikátní název účtu, e-mailovou adresu a heslo) a přihlášení na svůj uživatelský účet (vyžaduje e-mailovou adresu uživatele a heslo).

Mezi další případy užití se řadí veškeré pasivní funkce návštěvníka webu, tedy zobrazení devblogu, výpisu článků a po jeho rozkliknutí i zobrazení daného článku a komentářů k němu. Stejným způsobem se předpokládá, že host bude mít zájem zobrazit seznam her a po rozkliknutí dané hry i zájem o zobrazení podrobných informací k dané hře (zobrazení veškerých základních parametrů hry, popisu, videí a odkazů na obchod, kde se dá hra zakoupit, popř. na server, kde se dá hra stáhnout, pokud je dostupná oficiálně zdarma).

Dále si host může zobrazit veškeré základní informace o vývojářském studiu (sekce webu „About Us“) a může si také zobrazit presskit, který obsahuje soupis všech důležitých informací o studiu (fyzická adresa kanceláře, telefon, kontaktní e-mail apod.), výpis ze sekce „About Us“ a zkrácený výpis všech her studia.

Posledním předpokládaným případem užití hosta je zájem požádat o kopii hry pro recenzi, tzv. redaktorskou kopii hry. Jedná se o formulář, kam herní redaktor zadá svoji e-mailovou adresu, web nebo magazín, pro který píše, a vybere si hru, kterou má zájem recenzovat. Dobrovolně může také tento požadavek ještě komentovat, aby herní studio přesvědčil, že právě jemu má smysl kopii hry poslat, popř. tam může vložit i verifikaci zadané e-mailové adresy (tím se myslí např. odkaz na podstránku herního webu, kde je zadaný e-mail ve formuláři zmíněn).

5.1.2 Případy užití aktora uživatel

Za uživatele redakčního systému se považuje osoba, která se přihlásila prostřednictvím přihlašovacího formuláře (use case „Přihlásit se“), tedy osoba, která webovou stránku nejprve navštívila jako host. Aktor uživatel přebírá případy užití aktora host, ty jsou však obohaceny o další.

Uživatel má právo pracovat se svým vlastním profilem, tedy může změnit své heslo i další osobní informace (jméno, přezdívka, e-mail). Na přihlášeného uživatele pochopitelně spadá případ užití „Odhlásit se“, který uživatele webu přesměruje na hlavní stránku. Následně je takový uživatel považován za hosta, tedy předka uživatele v hierarchii aktorů.

Uživateli přibývá další případ užití v blogové části redakčního systému, a to možnost psát krátké textové komentáře k článku. Zároveň má právo své vlastní komentáře kdykoliv smazat.

5.1.3 Případy užití aktora moderátor

Za moderátora redakčního systému se považuje osoba, která se přihlásila prostřednictvím přihlašovacího formuláře a zároveň již před přihlášením měla od administrátora (nejvyšší aktor, popsán v následující podsekci) přidělená práva moderátora. Aktor moderátor tedy přebírá veškeré případy užití aktora uživatel, ty jsou však rozšířeny o bohaté možnosti správy celého redakčního systému.

Z praktického hlediska lze moderátora považovat i za editora webové stránky, jelikož má právo spravovat veškeré části v devblogu. Může přidávat nové články i měnit nebo mazat články stávající.

U článků má i přístup ke komentářům všech uživatelů, v závislosti na interní dohodě mezi moderátory a administrátorem o šíři moderování diskuze může smazat libovolný komentář u článků v devblogu.

Moderátor má přístup k seznamu žádostí o kopie her pro recenzenty, tudíž může žádosti po ověření delegovat člověku v týmu pověřenému pro generování aktivačních klíčů ke hře pro redaktory, který poté může odeslat klíč na e-mailovou adresu obsaženou v žádosti. Jakmile je žádost vyřízena (ať je tomu pozitivně či negativně), může tuto žádost tak označit – zmizí tím z hlavního výpisu aktuálních žádostí.

Před samotným spravováním her, ke kterým má moderátor také přístup, je vhodné se zaměřit na potřebné základní komponenty systému nutné pro vytvoření her k prezentaci – tím jsou myšleny případy užití „Spravovat platformy“, „Spravovat žánry“ a „Spravovat prodejny her.“ Všechny tyto tři případy užití jsou založeny na stejném principu, pouze jiném objektu činnosti, tedy jde o přidávání, mazání a editaci platforem (např. PC, Mac, PlayStation 4), žánrů (např. FPS, TPS, adventura, visual novel) a prodejen her (např. Steam Store, Itch.Io).

Samotný případ užití „Spravovat hry“ nabízí moderátorovi veškeré nástroje pro úpravu prezentace her na webové stránce redakčního systému, jedná se tedy o klasické přidávání her, jejich editace i mazání. Tento případ užití je však rozšířen o další možné a blíže specifické případy užití, a to spravování videí ke hře, spravování odkazů na nákup hry (u obou případů opět jde o přidávání, editaci i mazání) a nahrání nového loga ke hře. Pro nahrávání screenshotů (obrázků přímo ze hry) není připravený žádný přímý případ užití, avšak s touto funkcí se počítá již v základu případu „Spravovat hry“, protože moderátor může obrázky vkládat právě do popisu hry.

5.1.4 Případy užití aktora administrátor

Za administrátora redakčního systému se považuje osoba, která v systému existuje již od počátku – jedná se o úplně prvního uživatele se všemi možnými právy v rámci celého redakčního systému (administrátor tedy dědí všechny možné případy užití od moderátora a rozšiřuje je o své exkluzivní).

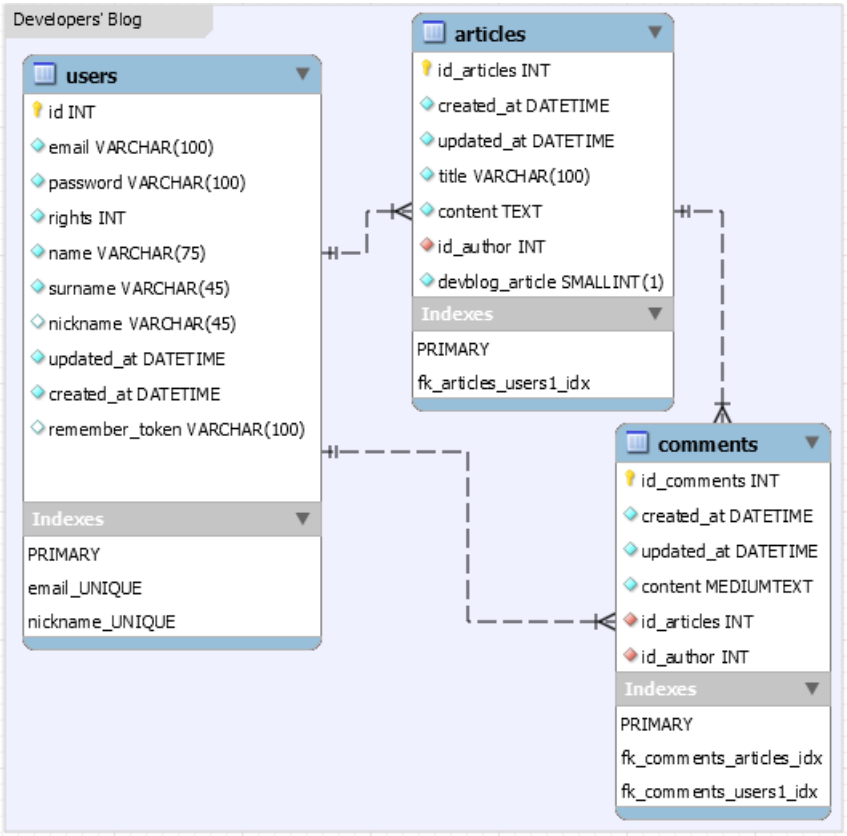
Administrátor může přímo spravovat uživatele, tedy sledovat jejich veškerou činnost na webu a v případě uživatelů narušujících chod webu může takovým uživatelům smazat celý účet (případ užití „Odstranit uživatele“). Ve správě uživatelů má administrátor také možnost přidávat nebo odebírat práva moderátora jednotlivým uživatelům.

Posledním případem užití administrátora je upravení základních informací o vývojářském studiu. Tím je myšlena změna fyzické adresy kanceláře studia, změna kontaktní e-mailové adresy a dalších podobných údajů, které jsou použity primárně v presskitu.

5.2 ERD databáze redakčního systému

Databáze redakčního systému je rozdělena do tří logických celků, které spolu vzájemně nemají žádné vztahy. Jedná se o databázi určenou pro devblog, databázi pro prezentaci her a databázi pro globální proměnné redakčního systému.

5.2.1 Databáze pro devblog



Obrázek 2: ER diagram devblogu

Část databáze pro devblog obsahuje celkem tři tabulky – users, articles a comments, každá tabulka má svůj vlastní unikátní identifikátor (celočíselný primární klíč s automatickou inkrementací).

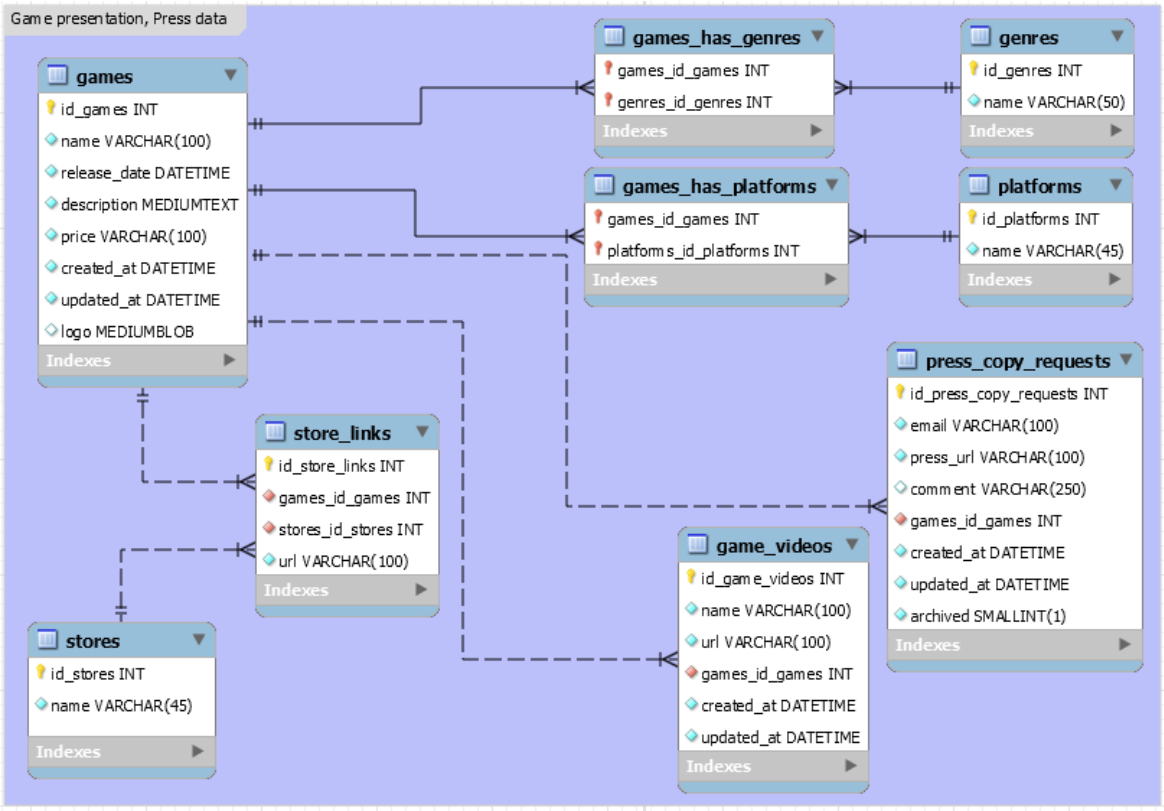
Tabulka users představuje uživatele, obsahuje jeho základní údaje (spravovatelné z profilu uživatele), dále sleduje časový údaj o vytvoření uživatele a o poslední změně uživatele (updated_at a created_at, jedná se o implicitní databázové proměnné pro PHP framework Laravel), úroveň jeho práv (číselná stupnice určující zdali se jedná o uživatele, moderátora nebo administrátora, kde nejnižší hodnota značí nejvyšší úroveň práv) a remember_token, což je token určený pro zapamatování si přihlášení v daném prohlížeči na daném zařízení. Uživatel má 1:n vazbu na

články (může být autorem článku za předpokladu, že je moderátorem nebo administrátorem) a také má 1:n vazbu na komentáře (může být autorem komentáře).

Tabulka articles představuje článek, základně obsahuje titulek a obsah článku (velké množství dat ve formátu TEXT, které správce upravuje a vytváří přes TinyMCE WYSIWYG editor), dále opět položky created_at a updated_at a nakonec jednoduchou proměnnou určující, zdali se jedná o článek pro devblog. Tato proměnná je použita pro speciální typ článku, který se neobjevuje v devblogu – jedná se o obsah sekce „About Us“ redakčního systému. Každý článek má svého autora (1:n vazba na uživatele) a může mít své komentáře (1:n vazba na komentář).

Tabulka comments představuje komentář, obsahuje textový obsah a implicitní časové položky created_at a updated_at. Každý komentář má svého autora (1:n vazba na uživatele) a každý komentář je určený k nějakému článku (1:n vazba na článek).

5.2.2 Herní databáze



Obrázek 3: ER diagram části databáze pro prezentaci her na webu

Část databáze pro prezentaci her obsahuje celkem 9 tabulek, z nich 2 jsou spojovací. Jedná se o tabulky games, stores, store_links, game_videos, press_copy_requests, genres, platforms a spojovací games_has_genres a ga-

mes_has_platforms. Každá nespojovací tabulka má svůj vlastní unikátní identifikátor (celočíselný primární klíč s automatickou inkrementací).

Tabulka games představuje základní část celé logiky databázového celku pro prezentaci her, představuje samotnou hru. Obsahuje název hry, datum vydání, popis hry (rozsáhlý výstup z WYSIWYG editoru), cenu hry (řetězcová hodnota pro případy, kdy by chtěl vývojář zadat cenu ve více měnách zároveň, např. „$7.49 / £4.99 / 7,49€“) a klasické implicitní časové hodnoty created_at a updated_at. Dále tabulka obsahuje atribut logo, který je ve formátu MEDIUMBLOB – ten umožňuje nahrát soubory do velikosti 16 MB (BLOB nabízí maximální velikost pouze 64 kB, což může být pro barevné a detailní logo nedostatečné).

Tabulka stores je primitivní, obsahuje pouze název obchodu – k této tabulce je v 1:n vazbě tabulka store_links, která k danému obchodu a dané hře přidává odkaz na nákup, popř. stažení, dané hry. Tabulka store_links má tedy 1:n vazbu na tabulku stores (každý odkaz vede na jeden vybraný obchod, popř. zdroj stažení) a 1:n vazbu na tabulku games (každý odkaz patří k jedné vybrané hře). Toto propojení, kde i tabulka store_links má svůj vlastní unikátní identifikátor (primární klíč), umožňuje, aby jedna hra měla více odkazů do jednoho obchodu – k takové situaci může dojít např. pokud vývojář na platformě prodává základní hru a rozšíření k ní či soundtrack, a bude chtít odkázat na každou (volitelnou) položku z nákupu hry zvlášť.

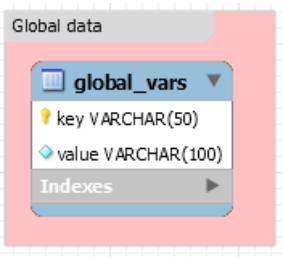
Tabulky genres i platforms se taktéž řadí mezi primitivní tabulky určené jen k udržení samotného názvu (žánru nebo platformy). Obě tabulky mají m:n vazbu na tabulku games, tedy každý žánr může patřit k více hrám a každá hra může mít více žánrů (být multižánrová). Stejně tak každá platforma může patřit k více hrám a každá hra může mít více platforem (být multiplatformní).

Tabulka game_videos slouží pro udržování herních videí (videa přímo ze hry, trailery nebo teasery), obsahuje název videa, URL k videu a implicitní časové hodnoty PHP frameworku Laravel. Tato tabulka má 1:n vazbu na tabulku games, tedy každé video patří k jedné hře a každá hra může mít více videí.

Poslední tabulkou v této části je tabulka press_copy_requests, která slouží k udržování žádostí o recenzentské kopie vybraných her. Tabulka obsahuje e-mail žadatele, URL odkaz na webovou stránku žadatele, nepovinný komentář k žádosti, implicitní časové hodnoty a jednobitovou proměnnou archived, která jednoduše slouží k rozlišení vyřízené a nevyřízené žádosti. Tato tabulka má 1:n vazbu na tabulku games, tedy každá žádost se týká určité hry a na každou hru může být více žádostí.

5.2.3 Databáze pro globální proměnné

Poslední část databáze určené pro globální proměnné představuje velmi primitivní řešení – celá databáze je obsažena v jedné tabulce s dvěma proměnnými, a to klíčem a hodnotou k danému klíči. Takové globální proměnné se dají použít pro obecné informace o studiu (např. adresa kanceláře, telefonní číslo kanceláře) nebo také pro nastavení chování redakčního systému.



Obrázek 4: ER diagram části databáze pro globální proměnné

6 Výsledky

Celá tato kapitola se věnuje samotné výsledné implementaci redakčního systému. Zdrojový kód k redakčnímu systému pro web nezávislých herních vývojářů lze najít na disku přiloženém k bakalářské práci.

6.1 Devblog – Vývojářský blog

Základní strana vývojářského blogu obsahuje výpis nejnovějších článků, které se prezentují ve značně zkrácené podobě. Tento požadavek zpracovává *DevblogController*, který aplikuje výstup na šablonu *devblog*. V takovém výpisu uživatel vidí titulek článku, jméno autora článku (popř. i přezdívku, pokud je nastavena), den vydání článku, počet komentářů a první odstavec článku. Extrakce prvního odstavce z článku pro ukázku (doplněnou odkazem na celý článek) je vytvořena základním použitím klasických PHP metod pro zpracování textu:

```
$article->content = substr( $article->content, 0,
                    strpos( $article->content, '' ) + 4 );
```

6.1.1 Zobrazení článku, komentáře

Při výběru článku z hlavní stránky devblogu se zobrazí článek na nové stránce na základě ID článku odeslaném v URL. Takový požadavek sleduje *DevblogController*, který aplikuje výstup na šablonu *devblog_article*. Takový článek obsahuje všechny základní informace, které už obsahoval ve výpisu všech článků, nicméně text článku (výstup z WYSIWYG editoru v administrátorském prostředí) je tentokrát už zobrazen v plném rozsahu.

Pod samotným článkem je však další podstatná část celé stránky, a to komentáře ke článku. Ty se vždy zobrazují seřazené tak, aby nejnovější komentář byl nejvýše. U komentáře může každý vidět jméno komentujícího (anonymní komentování není možné), datum a čas zápisu komentáře, a samotný jeho obsah. Navíc, pokud je uživatel autorem článku nebo moderátorem (popř. administrátorem), po pravé straně hlavičky komentáře vidí i červené tlačítko s glyfikonem odpadkového koše – to umožňuje vybraný komentář smazat. Celý kód v Blade šabloně pro toto tlačítko je zpracováno následujícím způsobem:

```
@if (Auth::user()->id == $comment->id_author ||
     Auth::user()->rights == 1 ||
     Auth::user()->rights == 2)
<form action="{{url('devblog/'.$article->id_articles)}}"
    method="POST">
{!! csrf_field() !!}
{!! method_field('DELETE') !!}
<input type="hidden" name="comment_id"
```

```
    value="{{$comment->id_comments}}" />
<button type="submit" class="btn btn-danger">
    <i class="fa fa-btn fa-trash"></i>
</button>
</form>
@endif
```

Vzhledem k požadavkům a syntaxi frameworku Laravel je z bezpečnostních účelů u každého formuláře nutné vždy odesílat CSRF token – ten se právě vkládá k formuláři metodou `csrf_field()`. Pro účely korektního routingu je také upřesněna metoda – `method_field('DELETE')` routeru dává najevo, že k POST žádosti nemá reagovat jako na POST, ale jako na DELETE (na klasickou POST žádost v tomto kontextu by tedy web reagoval tak, že žádost není platná).

Pod výpisem komentářů, pokud návštěvník stránky není host (tzn. je přihlášený uživatel, moderátor nebo administrátor), se objevuje i jednoduchý široký formulář pro odeslání nového komentáře. Vstupní formulář je řešený přes HTML textarea namísto jednořádkového inputu, nicméně neobsahuje žádný WYSIWYG editor – od komentářů se očekává jen zběžné vyjádření názoru k tématu. Tento formulář je navíc ošetřený tak, aby autor komentáře nemohl injektovat přes komentář libovolný kód do webu, tento vstup se tzv. escapuje, brání se tak klasickým injekčním praktikám (SQL Injection, Cross-Site Scripting).

6.1.2 Stránkování

Stránkování je na devblogu řešeno, po vzoru ostatních blogových služeb jako Google Blogspot a Wordpress, pouze relativním stránkováním, tzn. na stránce se objevuje pouze tlačítko pro zobrazení starších článků (narozdíl kupříkladu od vyhledávačů, které stránky výpisů číslují a umožňují mezi nimi tak rychlé skoky). Pokud se návštěvník dostane na stránku, kde se již nachází poslední existující článek (nejstarší článek v databázi), tlačítko pro zobrazení starších článků se již vůbec nezobrazuje.

Ačkoliv se jedná o poněkud omezující způsob stránkování, pro účely blogu jej lze považovat za naprosto dostatečný – pokud bude mít uživatel zájem o nějaký článek staršího data, pravděpodobně ho nalezne spíše skrze vyhledávání přes veřejný vyhledávač (možná i bez předchozí znalosti existence webu), než aby prohledával samotný web.

6.1.3 Správa článků

Správa článků je ukryta běžným uživatelům webu v administrátorské sekci (k té dostává přístup každý moderátor a administrátor webu přes tlačítko u správy profilu a odhlášení). Po rozkliknutí tlačítka „Manage articles“ se vypíše seznam všech devblogových článků v databázi redakčního systému, s odkazem na celý článek, přesným datem vydání a tlačítky pro editaci a odstranění článku. O správu článků se stará *ArticleEditorController* a Blade šablony *admin_article_list* a *admin_article_editor*.

V horní části stránky se také nachází tlačítko pro vytvoření nového článku, to uživatele odkáže na novou stránku s WYSIWYG editorem. Na tuto stránku se odkáže správce i při kliku na editaci vybraného článku, kde do tohoto editoru se vloží také stávající obsah článku (a titulek).

6.1.4 Správa komentářů

Správa komentářů je ukryta běžným uživatelům webu v administrátorské sekci (k té dostává přístup každý moderátor a administrátor webu přes tlačítko u správy profilu a odhlášení). Po rozkliknutí tlačítka „Manage comments“ se vypíše seznam komentářů s odkazem na článek, ke kterému patří, se jménem a e-mailem autora, datem odeslání, obsahem komentáře a tlačítkem k odstranění vybraného komentáře. Správu komentářů zpracovává *CommentController* a Blade šablona *admin_comment_list*.

6.2 Prezentace her vývojáře

Základní strana prezentace her vývojáře obsahuje výpis všech her ve zkrácené podobě – na první pohled si uživatel všimne malého loga hry, které je staženo přímo z databáze (JPG obrázek uložený jako MEDIUMBLOB). Po straně tohoto loga se nachází název hry, platforma, žánr, datum vydání, cena a odkaz na stránku věnovanou přímo dané hře.

Požadavek na zobrazení výpisu her zpracovává *GamesController*, který aplikuje výstup na šablonu *games*.

6.2.1 Zobrazení vybrané hry

Po výběru hry ze seznamu se zobrazí všechny dostupné informace o hře na nové stránce, na základě ID hry odeslané v URL (GET metoda). O zpracování tohoto požadavku se stará opět *GamesController*, který tentokrát aplikuje výstup na šablonu *game_page*.

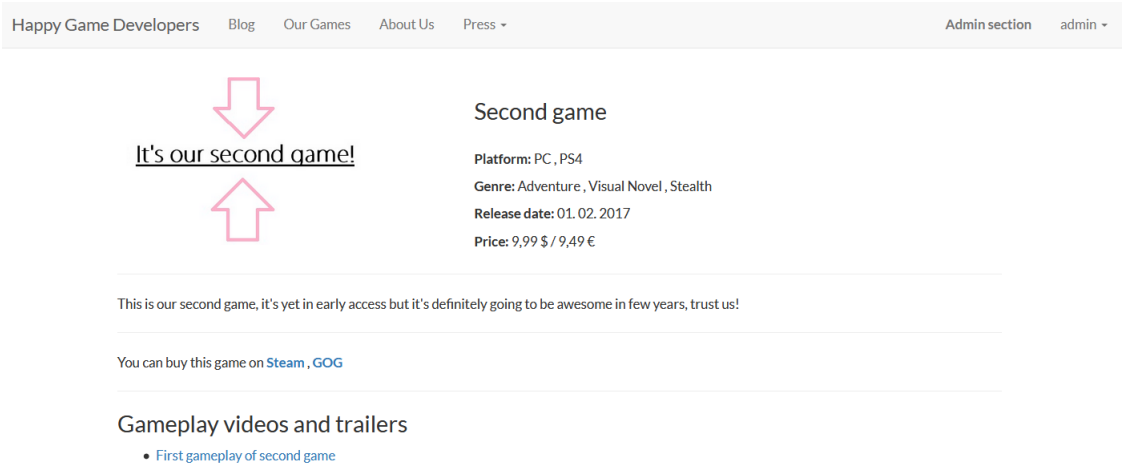
V horní části stránky se nejprve vypíše základní přehled o hře, totožný s vybraným prvkem v předchozím seznamu her. Obsahuje tedy malé logo hry, název, platformu, žánr, datum vydání a cenu.

Pod tímto přehledem se nachází popis hry, který je v databázi uložen v textovém formátu – předpokládá se delší obsah. Popis je výstupem z WYSIWYG editoru, proto se předpokládá, že bude obsahovat odstavce, seznamy, citace (např. pozitivní kritika herních médií), obrázky (screenshoty) či tabulky (minimální a doporučené herní požadavky na hardware).

Pod popisem hry se nachází krátký a jednořádkový seznam všech herních obchodů, kde je hra dostupná. Tyto odkazy jsou prezentovány velmi jednoduše a odděleny čárkou. Pokud hra není dostupná v žádném obchodě, namísto výpisu se zobrazí omluvné vyjádření k aktuální nedostupnosti hry.

Pod seznam obchodů se nachází bodový (nečíslovaný) seznam ukázkových videí ze hry (trailery, gameplay videa či tzv. „let’s playe“). Správce hry (tedy moderátor

nebo administrátor) není nijak omezen z hlediska zdroje videa, může tedy volně odkazovat na YouTube, Vimeo či jinou službu umožňující sdílení audiovizuálního materiálu.



Obrázek 5: Ukázkový snímek obrazovky se zobrazením vybrané (v tomto případě smyšlené ukázkové) hry

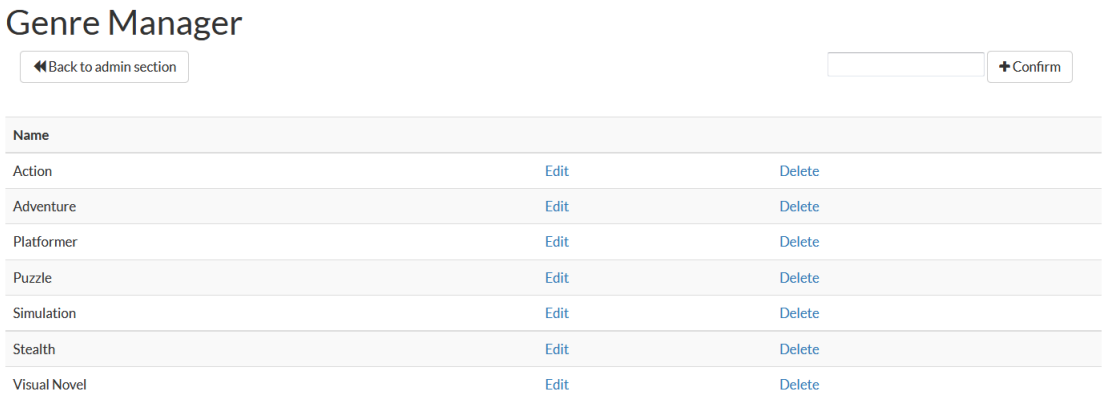
6.2.2 Správa žánrů, platforem a obchodů

Správa žánrů, platforem a obchodů je, vzhledem k jejich velmi podobné databázové struktuře, řešena takřka totožně. V administrátorské sekci věnované hrám tak administrátor si může vybrat, který z těchto prvků chce spravovat, poté je přesměrován na vybraný výpis. Žánry spravuje *GenreController* s výstupem na šablonu *admin_genres*, platformy spravuje *PlatformController* s výstupem na šablonu *admin_platforms* a obchody spravuje *StoreController* s výstupem na šablonu *admin_stores*.

Samotná stránka věnovaná správě je vytvořena tak, aby se s ní dalo pracovat velmi rychle a jednoduše. Správci se tak zobrazí výpis vybrané položky (žánrů, platforem či obchodů), na stejné stránce může hned přidat i novou položku. U každé položky má navíc možnost položku editovat (vloží se do formuláře pro přidání nové položky, ale potvrzení je přijato jako editace, nikoliv přidání nové položky) či mazat. Mazáním položky se kaskádově odstraní daná položka i z her, které ji používají. Pro ukázku lze níže vidět kód reagující na odstranění platformy:

```
public function deletePlatform($id){
    if (Auth::user()->rights != 1
        && Auth::user()->rights != 2) {
        return redirect('/');
    }
    $platformToDelete = Platform::find($id);
    $platformToDelete->games()->detach();
```

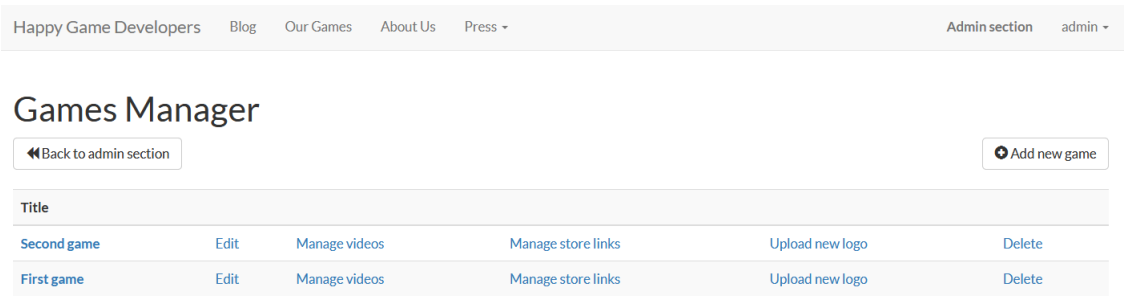
```
    $platformToDelete->delete();
    return redirect('/admin/platforms/');
}
```



Obrázek 6: Ukázkový snímek obrazovky stránky pro správu žánrů

6.2.3 Správa her

Správa her patří mezi nejkomplexnější prvky celé administrace redakčního systému vzhledem k relativně velkému množství databázových vazeb a logických propojení her s ostatními položkami redakčního systému. Při přechodu na správu her se vypíše přehledně do tabulky seznam všech her. Správce má dále možnost přidat do systému úplně novou hru nebo u vybrané hry změnit základní informace o hře, spravovat videa patřící k dané hře, spravovat odkazy na herní obchody k dané hře, nahrát nové logo ke hře nebo hru odstranit. Tento základní výběr spravuje controller *AdminGamesController* a výstup vykresluje šablona *admin_games_list*.



Obrázek 7: Ukázkový snímek obrazovky základního administračního menu pro správu her

Obrazovka pro přidání nové hry je prakticky totožná s editační (s tím rozdílem, že v případě editace se do formulářových prvků již vloží existující data k úpravě), základní struktura formuláře je rozdělena do dvou sloupců. V prvním sloupci je

položka pro název hry, datum vydání hry a cena hry, v druhém sloupci je vícepoložkový výběr platforem a žánrů. Na počítači výběr více položek je uskutečněn skrze použití kláves CTRL a SHIFT (systémová konvence), u mobilních zařízení Android bylo ověřeno, že při výběru vyskočí systémový výběr více položek – správci pracující z mobilních zařízení tedy nejsou nijak ochuzeni. Pod oběma sloupci se pak načítá přes zbytek obrazovky TinyMCE WYSIWYG editor pro vytvoření popisu ke hře. Celou tuto funkcionalitu spravuje opět controller *AdminGamesController*, výstup však vykresluje šablona *admin_games_editor*.

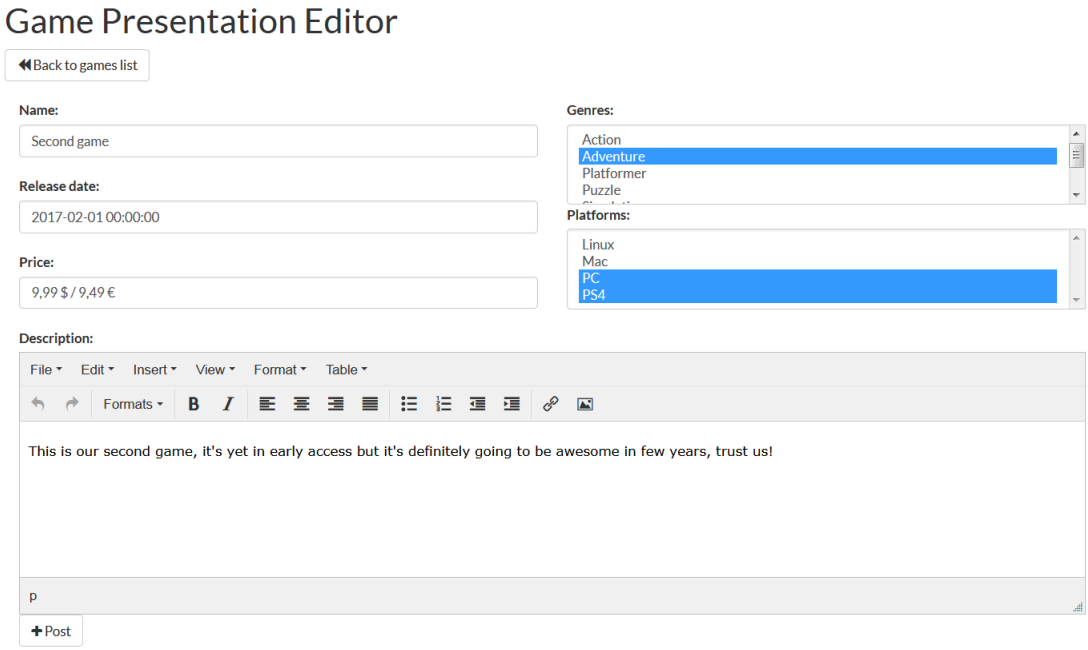
Editace základních informací o hře z hlediska řídící logiky (v controlleru) je ztvárněna následujícím kódem:

```
public function editGame($id_game, Request $request){
    if (Auth::user()->rights != 1
            && Auth::user()->rights != 2) {
            return redirect('/');
        }
    $edGame = Game::find($id_game);
    $edGame->name = $request->name;
    $edGame->release_date = $request->releaseDate;
    $edGame->price = $request->price;
    $edGame->description = $request->description;

    $edGame->genres()->detach();
    for($i=0; $i < count($request->genreSelect); $i++){
     $edGame->genres()->attach($request->genreSelect[$i]);
    }

    $edGame->platforms()->detach();
    for($i=0; $i < count($request->platformSelect); $i++){
     $edGame->platforms()->
                    attach($request->platformSelect[$i]);
    }

    $edGame->save();
    return redirect('/admin/games');
}
```

Pro pohled na samotný vícenásobný výběr je zde k nahlédnutí i úsek z šablony, ze které se přijímají vybrané žánry:

```
<select multiple class="form-control"
                  id="genreSelect" name="genreSelect[]">
@foreach ($genres as $genre)
    <option
    @if (in_array($genre->id_genres, $selectedGenreIDs))
```

```
            selected
        @endif
            value="{{$genre->id_genres}}">{{$genre->name}}
        </option>
@endforeach
</select>
```



Obrázek 8: Ukázkový snímek obrazovky formuláře pro přidání a editaci základních informací o hře

Správa videí k dané hře je řešená v podobném duchu jako byla již řešena správa žánrů, platforem a obchodů. Jediný praktický rozdíl je ve velikosti formuláře, kde se přidávají nebo upravují dvě položky – název videa a odkaz na video. Požadavky na správu videí řeší controller *GameVideoController* a výstup zpracovává šablona *admin_game_videos*.

S minimálními rozdíly je velmi podobným způsobem zpracovaná i správa odkazů na nákup (popř. stažení) hry, správce po prokliknutí se dočká jednoduché a přehledné tabulky s formulářem pro přidání na stejné stránce. Přidávací formulář je v tomto případě složen z URL a výběru obchodu, ke kterému odkaz patří. Požadavky na správu odkazů na koupi hry řeší controller *StoreLinkController* a šablona *admin_store_links*.

Posledním funkčním „podmenu“ správy jednotlivých her je možnost nahrání nového loga. To je zpracované velmi jednoduše přes základní formulářový vstup pro soubory, používaný již ve starších verzích PHP. Na samotnou velikost nahrávaného obrázku není daný žádný limit vzhledem k očekávané úpravě celkového designu

webu skutečným administrátorem, pro tuto ukázkovou testovací verzi (s Bootstrapem) se však doporučuje volit velikost loga 350 × 50 pixelů. Tyto malé obrázky nejsou uloženy přímo v souborovém systému hostingu, ale nachází se v databázi jako MEDIUMBLOB. Logo (a obrázky obecně) zpracovává *ImageUploadController* s šablonou *admin_logo_upload*. Samotné zpracování souboru z formuláře a jeho upload do databáze je řešen velmi jednoduše skrze ORM frameworku Laravel:

```
public function uploadNewLogo($id_game, Request $request){
$newLogo = file_get_contents(
            $request->file('logo-to-upload'));
$activeGame = Game::find($id_game);
$activeGame->logo = $newLogo;
$activeGame->save();
return redirect('/admin/games')->with('success',
            'A new logo for game '.$activeGame->name.
                        'has been successfully set.');
}
```

6.3 Další funkce redakčního systému

6.3.1 Homepage (Úvodní strana)

Úvodní strana redakčního systému je vytvořena s myšlenkou návštěvníkovi předvést nejnovější obsah na webové stránce. Svojí velikostí nepřesahuje velikost typické obrazovky uživatele (pouze na počítači, u mobilních zařízení je třeba se smířit s určitým kompromisem) a je rozdělena na dvě části – nejnovější článek a nejnovější hry.

Nejnovější článek je ztvárněn prakticky stejným způsobem jako u samotného výpisu článku v devblogu. Návštěvníkovi se tedy zobrazí název článku, jméno autora, datum vydání článku, počet komentářů a první odstavec z článku doplněný odkazem na celý článek.

V případě nejnovějších her se vypíší tři nejnovější hry, ve snaze však udržet jednoduchý vzhled domovské stránky se nezobrazí žádný popis ani název hry, pouze logo hry, které po kliknutí přesměruje návštěvníka na stránku se všemi informacemi o hře.

Celou homepage zpracovává *HomeController* s výstupem v šabloně *homepage*.

6.3.2 Uživatelský systém

Uživatelský systém redakčního systému je ztvárněn základní šablonou PHP frameworku Laravel verze 5.1. Ta umožňuje klasickou registraci (ta je však doplněna o vlastní položky jako např. volitelná přezdívka), přihlášení i odhlašovaní. Vlastním kódem je pak doplněna úprava uživatelského profilu i změna hesla. V celém kódu,

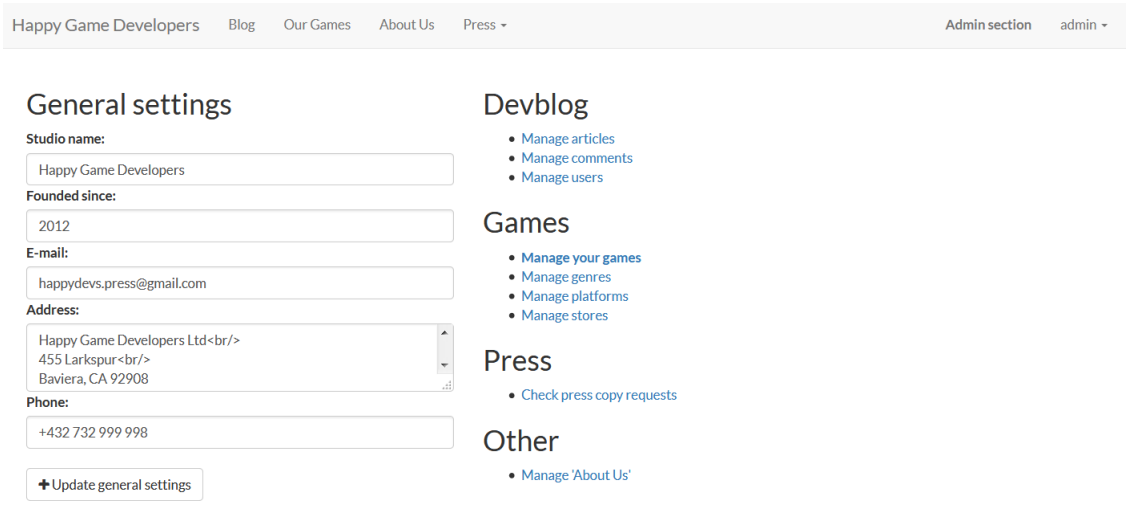
jak je už možné z předchozích ukázek poznat, se aktivně používá Laravelský middleware (rozděluje typy požadavků v routeru) a třída *Auth*, která aktivně u každé důležité metody controlleru opravdu kontroluje, zdali přihlášený uživatel má práva k operaci, kterou požaduje.

Hesla jsou uložena prostřednictvím hašovací funkce Bcrypt, níže lze vidět ukázkové použití v reakci na žádost o změnu hesla v controlleru *UserSettingsController*:

```
$user = User::find(Auth::user()->id);
 if (Hash::check($request->oldPass, $user->password)) {
  if ($request->newPass1 == $request->newPass2) {
   if (strlen($request->newPass1) > 5){
    $user->password = bcrypt($request->newPass1);
    $user->save();
    return redirect('/')->with('success',
                    'Password successfully changed.');
   } else {
    return redirect('/change-password')->with('error',
'Your password is too short. Use more than 5 characters.');
   }
  } else {
   return redirect('/change-password')->with('error',
                'New password entries doesn\'t match.');
  }
 } else {
  return redirect('/change-password')->with('error',
                                  'Wrong old password.');
}
```

6.3.3 Hlavní administrátorské menu

Administrátorské menu je část webu přístupná výhradně moderátorům a administrátorům, obsahuje veškeré nástroje pro správu obsahu celého redakčního systému. Základní menu bylo navrženo s myšlenkou jednoduchosti a přehlednosti, lze ho vidět na ukázkovém snímku obrazovky níže.



Obrázek 9: Ukázkový snímek obrazovky administračního menu z pohledu administrátora

6.3.4 Správa uživatelů

Správa uživatelů je unikátní podstránka administrace přístupná výhradně administrátorovi webu (moderátor v administrátorské sekci vůbec tlačítko vedoucí k této správě nevidí, samozřejmě i při zobrazení stránky se kontrolují práva uživatele). Správu uživatelů zpracovává *UserController* a Blade šablona *admin_user_list*.

V seznamu uživatelů administrátor vidí kompletní výpis informací o uživateli z databáze – jméno, e-mail, přezdívku, práva (administrátor, moderátor, uživatel) a celkový počet komentářů. Dále má, u všech uživatelů, kteří nejsou administrátorem, tlačítko pro možnost změny uživatelských práv na moderátorské (i naopak) a tlačítko pro odstranění vybraného uživatele.

6.3.5 Presskit

Presskit představuje kompletní výpis informací, který může být důležitý pro veškerá herní média. Pro obecného (nového) návštěvníka webu však může být také užitečný, prakticky obsahuje veškeré informace o studiu (včetně her) na jedné stránce. Tato stránka je však optimalizována primárně s myšlenkou na desktopové uživatele, nikoliv mobilní (neočekává se, že by někdo měl zájem sledovat strukturovaný výpis studia s následným kontaktem a psaním formálního e-mailu přes mobilní zařízení).

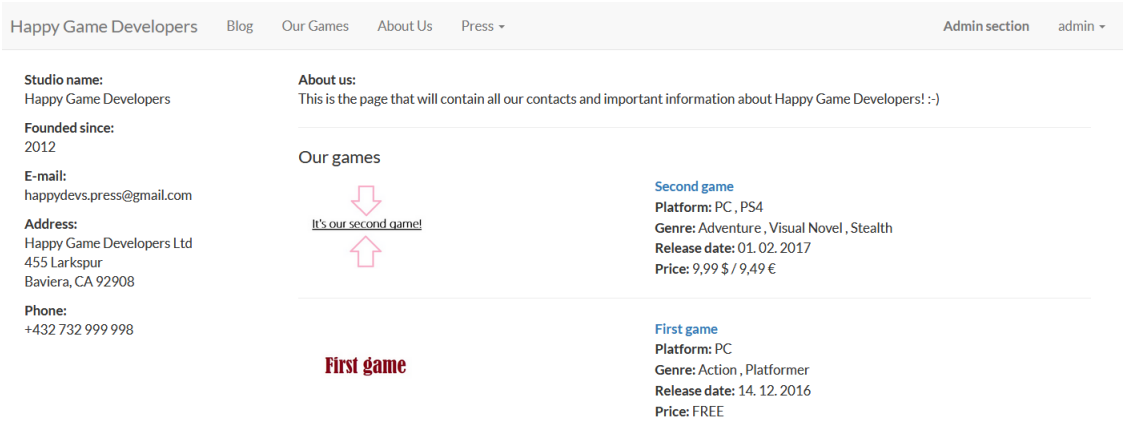
Presskit je rozdělen do dvou sloupců, kde levý je velmi úzký a obsahuje základní informace o vývojářském studiu. Tím se myslí název studia, datum (popř. jen rok) založení, hlavní kontaktní e-mail pro herní média, fyzická adresa kanceláře studia a telefonní kontakt do kanceláře.

V pravém (širokém) sloupci se nejprve vypíše celý obsah speciálního článku „About Us“ (jedná se o článek, se kterým se pracuje jako s blogovým, nicméně se na devblogu vůbec nezobrazuje, má svoji unikátní stránku a identifikátor 1),

kde má moderátor či administrátor příležitost vložit veškeré informace o studiu z historického pohledu, včetně odkazů na logo studia apod. Pod tímto článkem se poté zobrazuje kompletní výpis všech her studia ve velmi podobné formě, jakou již bylo možné sledovat u obecného výpisu her. Jediný praktický pozorovatelný rozdíl je v tom, že logo hry je zde mnohem menší, pro zdůraznění „hutnosti“ celého presskitu z hlediska množství informací pro nového návštěvníka.

Celý presskit zpracovává controller *PresskitController* a výstup je zpracován do šablony *presskit*. Samotné strukturální rozdělení prvků do dvou sloupců je řešeno pomocí bootstrapových tříd pro HTML prvek div – v tomto případě se jedná o aplikaci tříd `col-md-3` a `col-md-9`.

Obrovská výhoda presskitu je v tom, že správce redakčního webu ho nemusí nijak instalovat a ani do presskitu nemusí zadávat opakovaně specifická data (oproti velmi populární presskit šabloně od herního vývojáře Vlambeera). Presskit jednoduše pracuje se všemi daty uloženými v databázi.



Obrázek 10: Ukázkový snímek obrazovky presskitu (s ukázkovými údaji)

6.3.6 Žádosti o recenzentské kopie her

Žádosti o recenzentské kopie her jsou řešeny veřejným formulářem, v menu jsou dostupné jako podsekce části „Press“. Tuto část kódu zpracovává *PressRequestController* a šablona *review_copy_request_form*.

V samotném formuláři je po zájemci o recenzentskou kopii požadován kontaktní e-mail, URL on-line média, které zastupuje, a hra, kterou má zájem recenzovat (řešeno skrze HTML prvek select se seznamem všech her uložených v databázi). U formuláře je také prostor pro komentář k žádosti, tento prvek je však volitelný. Po kliku na odeslání žádosti se kontrolují jak přítomnost povinných prvků formuláře, tak i to, zdali formulář vyplnil opravdu uživatel nebo bot za účelu zatížení provozu celého systému. Toto ověření je provedeno skrytým input prvkem, od kterého se očekává, že po odeslání formuláře bude prázdný. Pokud není, požadavek je uznán jako strojový a není tak uložen do databáze.

Pro samotné sledování žádostí z pohledů moderátora či administrátora je určena speciální sekce v administrátorském menu. Pod tou se skrývá kompletní výpis nevyřízených žádostí – označení žádosti jako vyřízené se provádí jednoduchým tlačítkem „Archive“ přiloženým po pravé straně každé žádosti v jejich výpisu.

6.3.7 WYSIWYG editor TinyMCE a správa obrázků

Redakční systém jednoznačně vyžaduje kvalitní editor příspěvků – od moderátora nebo správce není korektní požadovat znalost HTML a CSS syntaxe za předpokladu, že opravdu budou pouze vykonávat správní funkce obsahu webu. Obyčejný text nestačí u větších příspěvků, v tomto redakčním systému se tím myslí články na devblogu a popisy her. Jejich autoři určitě budou chtít jednoduše použít odstavce, tabulky, obrázky či seznamy, a to co nejpříjemnější formou. Právě proto byl na velké textové formulářové prvky aplikován otevřený WYSIWYG editor TinyMCE.

Instalace, na kterou si stačilo vynaložit méně než pět minut času, ihned nabízí ve vybraném formuláři kvalitní správu textového obsahu, zvládá velmi přehledně a efektivně na vybrané úseky textu aplikovat změny velikosti a řezu písma. To však ale nestačí, proto k editoru byly aplikovány další pluginy (pouze však ty, které jsou dostupné ve verzi editoru dostupné zdarma), přesněji se jedná o pluginy link (hypertextové odkazy), autolink (automatické vytváření hypertextových odkazů), autoresize (omezování velikosti editoru, aby nepřekračoval hranice obrazovky při širokém vstupu, např. velkém obrázku), lists (číslované i odrážkové seznamy), media (vkládání HTML5 audia a videa), paste (konvertor textu z textových editorů jako např. Microsoft Word), preview (umožňuje ukázku článku nebo popisku před odesláním), table (tabulky) a image (správa obrázků).

Se samotným pluginem image však není tak jednoduchá instalace jako s ostatními pluginy vzhledem k tomu, že nahrávání obrázků na webový server vyžaduje jak podporu v backendu, tak i javascriptové (a AJAX/AJAJ) metody pro zpracování vstupu. PHP handler byl z velké části převzat ze samotné dokumentace WYSIWYG editoru TinyMCE, nicméně vyžadoval mnoho menších úprav pro samotné použití v kontextu PHP frameworku Laravel a také kontrolu duplicitních názvů souborů s obrázky. Tento handler byl zpracován formou metody do controlleru *ImageUploadController*, který se stará i o nahrávání loga hry (ty však ukládá do databáze, nikoliv přímo na souborový systém webu).

Po finálním zpracování funkčnosti pluginu image má správce (moderátor či administrátor) hned tři způsoby, jak může vybraný obrázek vložit do článku nebo popisu hry. Nejprimitivnější způsob je kopie obrázku (popř. přetažení) z jiné již existující webové stránky. V tom případě se obrázek neukládá na server, pouze se s odkazem vloží do textu. Druhým způsobem je nahrávání obrázku z vlastního počítače skrze systémový file explorer (ten je řešený skrze skrytý HTML formulář v kombinaci s JavaScriptem), v tomto případě se už obrázek nahraje přímo na webový server, do složky určené výhradně pro nahrané obrázky. Třetí možností je nahrání intuitivním způsobem, tzv. drag'n'dropem – správci textu stačí jednoduše přetáhnout soubor

s obrázkem přímo do textového editoru v prohlížeči pro jeho nahrání na webový server.

7 Testování

7.1 Administrátorský test – instalace na webhosting

Redakční systém pro web nezávislých herních vývojářů byl testován na serverech tří českých provozovatelů hostingových služeb – `http://www.webzdarma.cz`, `http://www.endora.cz` a `http://www.php5.cz`.

Průběh instalace všude proběhl totožně bez větších výhrad, na MySQL serveru byla založena databáze, do které se posléze importovaly všechny tabulky a vzorová (úvodní a uvítací) data. Poté byl upraven soubor .env s nastavením připojením k MySQL serveru a následně již nic nebránilo samotnému přesunu všech zdrojových souborů redakčního systému na souborový systém webu (vzhledem k velkému množství souborů PHP frameworku Laravel se jednalo o přenos na několik minut). Závěrem se stačilo přihlásit na vstupní administrátorský účet, do této fáze probíhalo vše bez problémů, jenže k tomu nemohlo dojít, protože všechny testované webové servery vykazovaly chybu 501: Server Error. Po delším zkoumání problému bylo zjištěno, že celý problém se skrývá v obsahu implicitního souboru .htaccess přiloženém ve složce public PHP frameworku Laravel. Pro zprovoznění redakčního systému na výše zmíněných serverech bylo potřeba odstranit následující část souboru .htaccess:

```
<IfModule mod_negotiation.c>
    Options -MultiViews
</IfModule>
```

Po této úpravě již nic nebránilo v přihlášení se k počátečnímu administrátorskému účtu, po delším vlastním zkoumání funkcionality však došlo k dalšímu kritickému problému, a to v případě serverů WebZdarma a Endora, které si do odpovědi na AJAX dotaz přiložili svoji reklamu a tím zneplatnili celý výstup PHP handleru, což znemožnilo nahrávání obrázků z WYSIWYG editoru na souborový server webu. Tento problém, i přes delší troubleshooting, nedošel k úspěšnému řešení – doporučované řešení selhávalo při aplikaci do frameworku. Je však důležité zmínit, že tyto zmíněné služby jsou dostupné zdarma a nepočítá se s nimi v rámci profesionálního užití s vyšší aktivitou ze strany uživatelů. Hostingovým službám dostupným zdarma se věnuje i kapitola Diskuze.

7.2 Uživatelský test

Uživatelský test proběhl na třech náhodně vybraných lidech, kteří byli schopni porozumět anglickému jazyku (alespoň A2 úroveň) a byli ochotni věnovat několik desítek minut tomuto testu. Každému uživateli byl předložen odkaz na již nainstalovaný redakční systém (viz minulá podsekce) a svěřen uživatelský účet s administrátorskými právy a obyčejný uživatelský účet. Následně dostali následující seznam úkolů:

1. Přihlašte se na administrátorský účet. Změňte si heslo na libovolné vlastní.

2. Přidejte na blog nový článek, vyzkoušejte různé odsazení odstavců, odrážky. Zkuste vytvořit tabulku a nahrát libovolný obrázek z Vašeho disku, jak přetažením souboru do editoru, tak i poklikáním na „Insert/edit image“.
3. Článek po odeslání zkontrolujte, zdali se v devblogu opravdu nachází. Poté k němu napište komentář.
4. Přidejte do redakčního systému alespoň jeden vlastní žánr, platformu a obchod (nemusí se jednat o reálná data).
5. Vytvořte novou hru, zařaďte ji do vlastní vytvořené platformy i žánru, zároveň i do předem volitelné platformy i žánru.
6. Přidejte ke hře video, alespoň jeden odkaz na obchod (opět se nemusí jednat o reálný použitelný odkaz) a nahrajte ke hře logo (libovolný obrázek splňující podmínky).
7. Odstraňte vlastní vytvořený žánr a sledujte, jak se projevila změna na stránce Vaší hry.
8. Přejděte na správu uživatelů – svému (druhému obdrženému) uživatelskému účtu přidejte práva moderátora. Odhlaste se z administrátorského účtu a přihlašte se s druhým (teď již moderátorským) účtem.
9. Smažte komentář, který jste vytvořili za administrátorský účet u svého článku. Poté upravte celý článek (libovolně) a sledujte změny. Poté článek odstraňte.
10. Změňte žánry a platformy u Vámi vytvořené hry za administrátorský účet.
11. Přejděte na formulář žádosti o recenzentskou kopii hry (dostupná i nepřihlášenému uživateli) a požádejte o hru, kterou jste sami vytvořili za administrátorský účet. Povinné údaje si vymyslete.
12. Z administrátorského prostředí si žádosti prohlédněte a poté žádost archivujte.
13. Odstraňte Vámi vytvořenou hru. Z moderátorského účtu odstraňte celý administrátorský účet.

U prvního úkolu se žádný z testerů nesetkal s problémem, bylo pouze vyčteno, že stránka po změnění hesla neinformuje uživatele o úspěšnosti nebo neúspěšnosti této operace. Vzhledem k této kritice byla pro každý interaktivní POST požadavek v redakčním systému doplněna tzv. flash message – krátká informativní zpráva zobrazující se pod hlavním menu stránky po proběhlé operaci. Flash mesasge může být zelená (úspěšně provedená operace), žlutá (upozornění) nebo červená (neúspěšně provedená operace). Po pravé straně každé flash message je vždy tlačítko pro schování této zprávy.

Vzhledem k tomu, že druhý úkol cílí na přechod do administrátorské sekce webu, dva testeři ze tří měli zpočátku problém si všimnout tohoto tlačítka (ačkoliv je v hlavním panelu a jako jediné vyznačené tučným písmem). Poté už však nedo-

šlo k žádným problémům v rámci přehledně vytvořeného výběru správních funkcí redakčního systému. U editoru článků byla vytknuta ikona určená pro přidání nebo editaci nového článku vzhledem k tomu, že znak plus je obvykle používán spíše pro přílohy než pro odeslání celkového obsahu. Proto byla tato ikonka zaměněna za jinou, funkčně i tématicky bližší vytváření a editaci článku na blog. Další výtkou bylo, že editor článků i při editaci obsahuje u potvrzovacího tlačítka text „Post“ – tento detail byl pochopitelně opraven, text tlačítka se tak dynamicky mění na základě toho, zdali se jedná o editaci článku nebo přidávání článku nového.

U třetího úkolu, týkající se kontroly napsaného článku přímo v blogové části redakčního systému, docházelo k problémům v krajních situacích, týkajících se obecně CSS problematiky a použití Bootstrapu v kombinaci s TinyMCE. Pokud se v závěru článku vloží tabulka, která se zarovná doprava, přesahuje pak obsahový prvek stránky určený komentářům a v designu vyvolává neplechu – jedná se o problémy, se kterými se pak zaručeně může setkat web designer, v tomto případě se problém dal vyřešit různými provizorními řešeními (např. vložení prázdného obsahu po levé straně tabulky). Další výtkou byla chybějící podpora středoevropských znaků vybraného fontu pro redakční systém – vzhledem k jeho cílové skupině mluvící primárně anglickým jazykem je redakční systém optimalizován pro globální použití, font si však samotný správce systému (resp. grafik a designer redakčního systému) může velmi jednoduše upravit v CSS.

Se čtvrtým úkolem neměl žádný z testerů problém, všichni dokázali bez problémů za okamžik přidat nový žánr, platformu i obchod.

S pátým úkolem nebyl žádný kritický problém, všichni zvládli vytvořit novou hru, nicméně se k tomu našly určité výhrady. Jednou z kritik byla nepřipravenost formulářového prvku pro cenu hry, tento potenciální nedostatek je však již odůvodněn v začátcích kapitoly určené výsledkům – cena hry může být dána libovolně podle vývojáře na základě toho, že si tak může do jednoho prvku vložit více měn zároveň (někteří jsou zvyklí psát společně cenu pro více regionů, např. „9,99 € / $6,99“). Další možností je, kdyby studio hru nabízelo zdarma, v tom případě do tohoto formulářového prvku mohou napsat jednoduše „FREE“. Další výhrada byla mířená na menu výběru platforem a žánrů, které „podivně zešediví“ při použití. Po diskuzi s testerem a sledování problému se došlo k závěru, že tento problém je způsoben zpracováním formuláře prohlížečem a operačním systémem (tento test proběhl z operačního systému OS X) – tento HTML formulář si vizuálně zpracovává implicitně systém podle sebe, jinak tedy vypadal i na operačním systému Windows a Android (vlastnoručně testováno). Všude však byl plně funkční.

Šestá úloha proběhla bez větších problémů, jeden z testerů měl pouze výhrady k tomu, že se do formuláře pro nahrání odkazu automaticky nevkládá http prefix. Ten byl tedy do formuláře doplněn. Jiný tester prohlásil, že proti systému vyvolal rebelii a nahrál na server logo většího rozlišení, než je doporučované. Je třeba však poznamenat, že velikost loga je opravdu pouze doporučená pro základní nastavení vzhledových prvků redakčního systému. Správce (resp. webdesigner) si může vizuální styl měnit dle své libosti, což zahrnuje právě i způsob vykreslování loga hry.

U sedmé úlohy všichni testeři poznamenali, že po smazání žánru se žánr odstranil i přímo ze hry. Úspěšně tak bylo ověřeno rekurzivní mazání vazeb tohoto prvku redakčního systému.

Osmý úkol taktéž proběhl bez jakýchkoliv problémů, všichni již v okamžiku byli schopni přejít do administrátorské sekce a kliknout na odkaz vedoucí ke správě uživatelů. Tam každý hned dokázal změnit roli vybraného uživatelského účtu na moderátorský.

S žádným problémem se testeři nesetkali ani z hlediska moderování diskuze, všichni byli schopni odstranit vybraný komentář. Je dobré poznamenat, že šlo pozorovat rozdílnost přístupů k odstranění komentáře – někdo ho odstraňoval přímo ze stránky článku s komentářem, někdo přešel do administrátorské sekce a podsekce pro správu komentářů.

Editace žánrů i platforem nebyla problémem pro žádného testera – desátá úloha tak proběhla bez problémů.

U jedenácté úlohy, tedy vyplnění a odeslání požadavku na recenzentskou kopii hry, se pouze jeden z testerů měl problém zorientovat v hlavním menu, než si všiml, že se tato možnost skrývá jako podmenu sekce „Press“. Se samotným formulářem pak jeden z testerů zkoušel odeslat i prázdné povinné buňky formuláře (kontaktní e-mail) a ověřil, že taková operace je neplatná. Došlo k výtce na to, že stránka opět nevykázala žádnou zpětnou vazbu ohledně úspěšnosti nebo neúspěšnosti odeslání formuláře, tento problém však již byl vyřešen při zpracování zpětné vazby testerů na první úlohu – formulář tedy již používá flash messages.

U dvanácté úlohy se žádný tester nesetkal s problémy, jednoduše přešli do administrátorské sekce a vybrali podmenu určené vyřizováním žádostí. Tam si žádost prohlédli (pomyslně odeslali e-mail s reakcí na žádost) a poté žádost archivovali.

Poslední úloha byla už jen praktická kontrola racionálního uvažování samotných testerů, protože je samozřejmě absurdní žádat o smazání hlavního administrátorského účtu z pozice administrátora. Vykonání této úlohy je naprosto nemožné za předpokladu, že nedojde k překonání bezpečnostních prvků webu ilegální cestou. Všichni testeři tak pochopitelně přiznali neúspěch vykonat závěrečnou úlohu uživatelského testování správy redakčního systému pro web nezávislých herních vývojářů.

8 Diskuze

8.1 PHP Framework

PHP frameworky jsou záležitostí, která rozsáhle mění celý způsob tvorby webu z původní relativně primitivní práce s PHP (na backendu). To však platí jen do jisté míry, protože již architektura MVC (Model-View-Controller) byla často aplikována přímo v „čistém PHP“, nicméně se nejednalo o žádnou normu.

Hlavními důvody pro použití takového frameworku jsou v dnešní době (opomeneme-li sekundární důvody jako např. trend profesionálních firem zaměřených na vývoj webových aplikací) zachování jednotnosti v přístupu k prvkům webu, bezpečnost (ta však např. z pohledu SQL Injection je zajištěna i bez frameworku při použití knihovny PDO, kterou snad již lze považovat za normu při tvorbě backendu) a pohodlí pro samotného programátora. To je zajištěno různými nadstandardy jednotlivých frameworků, mimo klasické šablonové systémy (např. Blade u Laravelu, Latté u Nette) se může jednat o objektově relační mapování databáze (ORM), migrace a seedování databází, předpřipravené formuláře či podpora příkazů pro jednodušší operaci se strukturou webu z příkazové řádky (CLI, v případě Laravelu se jedná o základní modul PHP Artisan).

Na druhou stranu přínos takových možností zároveň zvyšuje celou učební křivku programátora webových aplikací vzhledem k tomu, že se základy PHP si již nevystačí. Každý framework může mít jinou strukturu a vyžaduje určitý čas pro pochopení na určité úrovni.

Z použití frameworků pak vyplývají i další starosti programátora, které by jinak nemusely nastat – v případě PHP programátora byl aktuálně největším zlomem přechod z PHP verze 5.6 na verzi 7.0, kde se mění určité případy užití a zastaralé funkce se likvidují (deprecated). U frameworků k takovým rozsáhlým updatům může docházet velmi často (i v rámci měsíců) s tím, že programátor si nemůže být nikdy jistý, zdali druhý den nebude muset strávit celý učením předělané syntaxe frameworku a následně jejím předěláváním na implementovaném webu s frameworkem. Autoři frameworků na tenhle problém reagují určitou „zárukou“, která se obvykle pohybuje v rozsahu šesti měsíců až jednoho roku. Jsou však i výjimky, právě PHP framework Laravel, jak již bylo řečeno v kapitole Přehled literatury, tuto záruku nabízí po dobu tří let (*Laravel*, ©2017b).

Je zřejmé, že od webového programátora se neočekává perfektní znalost operačních systémů či přímo počítačů na úrovni strojového kódu. I přesto se však typický framework stává další „černou skříňkou“, která může stát i za neočekávanými problémy při implementaci. Za skvělý příklad lze považovat nečekanou událost z vývoje redakčního systému pro nezávislé herní vývojáře – v období testování verzí systému blízkých té finální došlo k tomu, že se vykreslovala neustále stará šablona, která již na souborovém systému webového serveru vůbec nebyla. I přesto, že došlo ke smazání cache webového prohlížeče, se pořád vykresloval starý a neaktuální výstup (v té době již nefunkční). Problém totiž nastal v tom, že compiler PHP frameworku

Laravel nedetekoval změnu ve složce s šablonami, ačkoliv došlo k jejímu kompletnímu přepsání. Tento problém lze však přiřadit operačnímu systému Windows, který pracuje s časovými razítky u souborů jinak než je zvykem na Linuxových serverech – řešením tohoto problému tedy byla editace šablony přímo na serveru, před ním se však skýtal zbytečně dlouhý čas zkoumání problému, který neměl vůbec nastat (není oficiálně zdokumentovaný).

ORM se na první pohled může zdát velkou výhodou – programátor nemusí tápat v SQL syntaxi, namísto toho jednoduše propojí svůj databázový dotaz přes volání relačních metod modelů. Vše je tak v PHP kódu mnohem jednodušší a přehlednější. PHP framework Laravel, ať už s použitím ORM nebo bez, se připojuje k databázi pomocí známé knihovny PDO. Na testovacích datech byla porovnávána rychlost vykonání databázového dotazu s použitím ORM i s použitím klasického SQL dotazu, nicméně po několika testovacích vzorcích v průměru prakticky nedocházelo k rozdílu v rychlosti. V praktickém použití s velkým množstvím dat však programátoři mohou dojít k jinému výsledku, ORM Eloquent se doporučuje při použití relací, Scope metod a pomocných funkcí u modelů, v jiných případech je vhodnější volit klasický databázový dotaz (Raman, 2015).

PHP frameworky jsou rozhodně užitečným a mocným nástrojem, nicméně je nelze idealizovat, ve všem můžeme najít chybu. Proto je vhodné před programováním se rozhodnout (obzvlášť u projektů menších rozměrů), zdali je použití velkého frameworku opravdu potřeba.

8.2 WYSIWYG editor

WYSIWYG editor představuje pro programátora redakčního systému obrovské ulehčení práce za cenu vložení externího kódu. V tomto redakčním systému byl pro články a popisy her implementovaný editor TinyMCE (verze zdarma, s méně dostupnými pluginy), který se honosí kvalitním ztvárněním a velmi jednoduchou instalací.

Prvotní instalace v podobě nakopírování javascriptových knihoven a propojení s vybranými formulářovými prvky je opravdu jednoduchá a rychlá, programátor tak v několika minutách dojde k vizuálně příjemnému editoru umožňující základní zpracování textu na úrovni odstavců a řezu textu. Pro další funkce už pak přichází pluginy, které s obtížností implementace se liší plugin od pluginu. V této části stojí za poznamenaní plugin image.

Tento plugin je v dokumentaci TinyMCE popsán relativně stručně, avšak bez všech podstatných detailů pro použití i implementaci. Pro zpracování obrázků je potřeba si vytvořit vlastní PHP handler, který je v dokumentaci jen hrubě naznačen ukázkou, dále je potřeba si napsat vlastní frontendový vstup na soubory s tím, že TinyMCE část předpokládaných úloh (automatický upload) zvládá, část nikoliv (ruční upload přes file explorer). Výsledkem tak je vypozorovaná zkušenost, že každý programátor si udělá vlastní řešení (nebo přímo vlastní modul), který pak sdílí na internetu s ostatními. Tím vzniká velké množství možností, kde téměř každá z nich má své vady (nezdokumentované) a programátorovi nakonec nezbývá nic jiného než

s určitým kompromisem řešit plugin sám (což opět bez znalosti interní struktury WYSIWYG editoru je poněkud náročné).

Dalším problémem WYSIWYG editorů mohou být i nekompatibilní vstupy s okolím webu, jak k tomu např. došlo už při uživatelském testování, kdy vpravo odsazená tabulka na konci článku přesahovala do komentářové tabulky.

Vyplatí se tedy se všemi zápory WYSIWYG editor používat? Pokud programátor nechce strávit hodiny psaním vlastního, tak zaručeně. Jen je důležité neočekávat od editoru žádné zázračné a bezproblémové řešení, vady je potřeba respektovat a zvládat vhodně zpracovat pro svůj požadovaný výstup.

8.3 Freehostingové služby

Provozování webhostingových služeb rozhodně není administrativně levná záležitost, proto je poněkud absurdní očekávat, že by měl někdo zájem seriozní a potenciálně navštěvovaný webový projekt provozovat na webovém hostingu zdarma (a k tomu ještě na doméně třetí třídy).

Hostingové služby provozované zdarma mají obvykle velké omezení z hlediska množství dat (jak přenášených, tak ukládaných) a z hlediska rychlosti při připojení více uživatelů webu. Navíc, mnohé takové služby ke každému webu, ať už do hlavičky nebo patičky, přidávají vlastní reklamní proužek, který slouží k tomu, aby služba nebyla naprosto ztrátová.

Z toho však vyplývají i problémy poznané při samotném testování (sekce Testování, podsekce Administrátorský test), kdy se reklama vkládala i do odpovědí na AJAX požadavky, a tím znemožňovala takové požadavky na celém webu (v případě redakčního systému se jednalo o nahrávání obrázků na webový server). U některých hostingů jejich správci mají zájem takové problémy řešit (a na diskuzních fórech se tomu zběžně věnují), nicméně např. v tomto případě i po delší snaze nedošlo ke zprovoznění finální verze frameworku na takových webhostingových službách.

Použití redakčního systému pro nezávislé herní vývojáře na freehostingu lze považovat za zavržené. Taková implementace je už z praktického konceptu špatná, pokud má zájemce provozovat seriozní web na úrovni, který bude mít vyšší návštěvnost než několik uživatelů měsíčně. Každá správná webová prezentace vyžaduje vlastní doménu druhé třídy a kvalitní a spolehlivý hosting (ceny se u menších projektů pohybují v rámci stokorun měsíčně).

9 Závěr

Redakční systém pro nezávislé herní vývojáře splňuje vyjmenované základní požadavky vytyčené v cíli práce. Umožňuje základní správu článků, správu her vývojářského studia a nabízí funkcionalitu vhodnou pro komunikace s herními médii (redaktoři nebo např. populární veřejně prezentující se hráči). Další rozšíření by však byla jednoznačně přínosná. Mimo unikátní webový design, který se očekává před implementací pro web herního studia, by bylo jednoznačně vhodné systém rozšířit o podporu e-mailových služeb, verzování článků a vícejazyčné verze webu.

Tento redakční systém je první (větší) autorův projekt s PHP frameworkem Laravel. I když došlo k mnoha hodinám učení, zpracovávání redakčního systému i opravám chyb a řešení návrhů nových funkcí, systém nelze považovat za perfektní. Jedná se však o plně dostačující nástroj pro nezávislé herní vývojáře, pro které je zájem tento redakční systém vydat s licencí pro volné použití i úpravy.

10 Reference

DRUPAL. *About*. Drupal [online]. Drupal, ©2016 [cit. 2016-11-23]. Dostupné z: https://www.drupal.org/about.

GALYONKIN, S. *38% of all Steam games were released in 2016*. In: Twitter [online]. 2016 [cit. 2017-03-21]. Dostupné z: https://twitter.com/Steam_Spy/status/804072335997358084.

HENICK, B. *HTML & CSS: the good parts*. Sebastopol, Calif.: O'Reilly, 2010. 352 s. ISBN 978-0-596-15760-9.

HOGAN, B. P. *HTML5 a CSS3 : výukový kurz webového vývojáře*. 1. vyd. Brno: Computer Press, 2011. 272 s. ISBN 978-80-251-3576-1.

JOHNSTON, M. *CMS or WCM - Which is Which?* CMS Critic [online]. 2011 [cit. 2016-11-23]. Dostupné z: https://www.cmscritic.com/cms-or-wcm-which-is-which/.

LARAVEL. *Laravel - The PHP Framework for Web Artisans*. Laravel - The PHP Framework for Web Artisans [online]. Taylor Otwell, ©[2017]a [cit. 2017-04-07]. Dostupné z: https://laravel.com/.

LARAVEL. *Release Notes: Support Policy*. Laravel - The PHP Framework for Web Artisans [online]. Taylor Otwell, ©[2017]b [cit. 2017-03-21]. Dostupné z: https://laravel.com/docs/5.4/releases/#support-policy.

LOCKE, J. *Top 6 reasons Drupal really sucks - Developer Edition*. Freelock [online]. 2011 [cit. 2017-04-04]. Dostupné z: https://www.freelock.com/blog/john-locke/2011-10/top-6-reasons-drupal-really-sucks-developer-edition.

MACINTYRE, P. *PHP: the good parts*. Sebastopol, CA: O'Reilly, 2010. 176 s. ISBN 978-0-596-80437-4.

NETTE. *Rychlý a pohodlný vývoj webových aplikací v PHP | Nette Framework*. Rychlý a pohodlný vývoj webových aplikací v PHP | Nette Framework [online]. Nette Foundation, ©2017 [cit. 2017-04-27]. Dostupné z: https://nette.org/.

PATAKI, D. *Is WordPress Code Really A Mess?*. Kinsta [online]. 2016 [cit. 2017-03-21]. Dostupné z: https://kinsta.com/blog/is-wordpress-code-really-a-mess/.

PHP-FUSION. *Main - Official home of PHP-Fusion*. Official home of PHP-Fusion [online]. PHP-Fusion Inc, ©2017 [cit. 2017-04-27]. Dostupné z: https://www.php-fusion.co.uk/home.php.

SANDU, B. *The Case Against Using Bootstrap To Design Websites*. Design your way [online]. ©[2017] [cit. 2017-03-21]. Dostupné z: http://www.designyourway.net/blog/inspiration/the-case-against-using-bootstrap-to-design-websites/.

RAMAN, S. *Laravel Eloquent vs Fluent query builder*. Sriraman - Passionate web developer [online]. 2015 [cit. 2017-04-16]. Dostupné z: https://blog.sriraman.in/laravel-eloquent-vs-fluent-query-builder/.

VERENS, K. *CMS design using PHP and jQuery: build and improve your in-house PHP CMS by enhancing it with jQuery*. Birmingham, U.K.: Packt, 2010. 340 s. ISBN 978-1-8495-1252-7.

WORDPRESS. *About*. WordPress [online]. WordPress, ©2016 [cit. 2016-11-23]. Dostupné z: https://wordpress.org/about/.

W3SCHOOLS. *HTML5 Video*. W3schools Online Web Tutorials [online]. Refsnes Data, ©2016 [cit. 2016-11-23]. Dostupné z: http://www.w3schools.com/HTML/html5_video.asp.

Přílohy

A Zdrojový kód

Na přiloženém disku se nachází zdrojový kód k redakčnímu systému pro web nezávislých herních vývojářů a stručný textový manuál pro instalaci (v českém i anglickém jazyce). U zdrojového kódu se pro jednoduchou instalaci bez problémů nachází i knihovny nezbytné pro zprovoznění redakčního systému, ty však pochopitelně nejsou dílem autora práce.