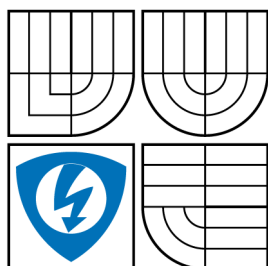


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

REDAKČNÍ SYSTÉM PRO RECENZI ČLÁNKŮ EDITORIAL SYSTEM FOR ARTICLE REVIEWS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

BC. ROMAN BUŇKA

VEDOUCÍ PRÁCE
SUPERVISOR

ING. RADIM BURGET

BRNO 2010

ZDE VLOŽIT LIST ZADÁNÍ

Z důvodu správného číslování stránek

ZDE VLOŽIT PRVNÍ LIST LICENČNÍ
SMOUVY

Z důvodu správného číslování stránek

ZDE VLOŽIT DRUHÝ LIST LICENČNÍ
SMOUVY

Z důvodu správného číslování stránek

ABSTRAKT

Cílem diplomové práce je analýza, návrh a implementace webové aplikace, která má sloužit autorům článků, recenzentům a správci systému k automatizaci recenzního řízení u vědeckých článků. V analýze recenzního řízení jsou stanoveny požadavky na aplikaci z pohledu jednotlivých uživatelských rolí. Návrh redakčního systému se zaměřuje na architekturu aplikace a návrh tříd. Vytvořená aplikace je představena společně s popisem její implementace.

KLÍČOVÁ SLOVA

Java, HTML, Wicket, Hibernate, Tomcat, MySQL

ABSTRACT

The objective of the thesis is an analysis, design and implementation of a web application which could be used by authors of articles, reviewers and the system administrator for automation of reviewing scientific articles. The reviewing analysis shows requirements required for the application from the viewpoint of users' roles. The editorial system design deals with the application architecture and a plan of classes. The application is shown together with a description of its implementation.

KEYWORDS

Java, HTML, Wicket, Hibernate, Tomcat, MySQL

BUŇKA, R. *Redakční systém pro recenzi článků*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2010. 52 s. Vedoucí diplomové práce Ing. Radim Burget.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Redakční systém pro recenzi článků“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

(podpis autora)

OBSAH

Úvod	11
1 Analýza recenzního řízení	12
1.1 Požadavky na recenzní řízení	12
1.1.1 Šéfredaktor	12
1.1.2 Autor	13
1.1.3 Recenzent	14
1.1.4 Čtenář	14
1.2 Časová posloupnost recenzního řízení	14
2 Návrh redakčního systému	16
2.1 Architektonický vzor MVC	16
2.1.1 Apache Wicket	16
2.2 Návrh perzistentních tříd	18
2.3 Objektově-relační mapování	20
2.3.1 Hibernate	20
2.3.2 Datový model	20
3 Implementace redakčního systému	22
3.1 Veřejná část	22
3.1.1 Úvodní stránka	22
3.1.2 Registrace do systému	24
3.1.3 Přihlašování do systému	26
3.2 Administrační rozhraní	26
3.2.1 Změna osobních údajů	27
3.2.2 Změna role	27
3.2.3 Uživatelské účty	28
3.3 Recenzní řízení	29
3.3.1 Recenzní řízení pro roli Autor	30
3.3.2 Odevzdání článku	30
3.3.3 Editace článku	31
3.4 Recenzní řízení pro roli Recenzent	31
3.4.1 Odevzdání recenze	33
3.4.2 Odsouhlasení recenze	33
3.5 Recenzní řízení pro roli Šéfredaktor	33
3.5.1 Vytvoření aktuality	34
3.5.2 Editace aktuality	34

3.5.3	Nastavení termínu	35
3.5.4	Nastavení e-mailu	36
3.5.5	Přidělení recenze	36
3.5.6	Recenze a opravy článků	37
3.5.7	Editace instrukcí	39
3.5.8	Nastavení upomínky	39
3.5.9	Vyhodnocení článků	40
3.5.10	Editace sborníku	41
3.5.11	Editace článku	41
3.6	Implementace sezení	43
3.7	Autorizační strategie	44
3.8	Testování	44
4	Zprovoznění aplikace	46
4.1	Konfigurace aplikace	46
4.2	Logování zpráv	47
4.3	Zabezpečení komunikace	48
4.4	Nasazení aplikace na server	48
5	Závěr	50
	Literatura	51
	Seznam symbolů, veličin a zkratk	52

SEZNAM OBRÁZKŮ

1.1	Diagram případů užití	13
1.2	Sekvenční diagram UML	15
2.1	MVC architektura v Apache Wicket	17
2.2	Diagram perzistentních tříd	19
2.3	E-R diagram	21
3.1	Úvodní stránka	23
3.2	Registrace uživatele	25
3.3	Přihlášení uživatele	26
3.4	Změna osobních údajů	27
3.5	Změna uživatelských rolí	28
3.6	Správa uživatelských účtů	29
3.7	Odevzdání článku	30
3.8	Editace článku	32
3.9	Odsouhlasení recenze	33
3.10	Editace aktuality	34
3.11	Nastavení termínu	35
3.12	Přidělení recenze	36
3.13	Seznam recenzentů	37
3.14	Recenze a opravy článků	38
3.15	Nastavení upomínky	39
3.16	Vyhodnocení článků	40
3.17	Editace sborníku	41
3.18	Editace článku	42
3.19	Výběr článku	43
3.20	Platnost stránky vypršela	43
3.21	Přístup na stránku zamítnut	44
4.1	AJAX ladící program	46
4.2	Chyba za běhu programu	47
4.3	Adresářová struktura aplikace	48

ÚVOD

Tato práce se zabývá analýzou, návrhem, implementací a zprovozněním webové aplikace v prostředí Java EE. Webová aplikace slouží autorům článků, recenzentům a správci systému k automatizaci recenzního řízení u vědeckých článků.

První kapitola se zabývá analýzou problému, popisuje případy užití pro jednotlivé uživatele systému, časovou posloupnost recenzního řízení. Na analýzu navazuje návrh aplikace, který popisuje architektonický vzor MVC tak, jak je implementovaná v rámci Apache Wicket, pomocí kterého je aplikace vytvořena. Je proveden návrh perzistentních tříd pro objektivě-relační mapování na tabulky databáze pomocí rámce Hibernate. Třetí kapitola se zabývá popisem implementace a představením vytvořené aplikace. Postupně je zde popsána veřejná část aplikace, následuje administrační rozhraní a největší část tvoří implementace recenzního řízení pro jednotlivé uživatelské role. V poslední části jsou uvedeny způsoby konfigurace aplikace a nasazení aplikace na server.

1 ANALÝZA RECENZNÍHO ŘÍZENÍ

Účelem analýzy je pochopit, jak se bude aplikace používat a co musí vykonávat. Cílem této analýzy je určit požadavky na webovou aplikaci, která má sloužit autorům článků, recenzentům a správci systému k automatizaci recenzního řízení u vědeckých článků. K dokumentování analýzy se používá jazyk UML – unifikovaný modelovací jazyk, který je standardizovaným modelovacím jazykem v oblasti softwarového inženýrství.

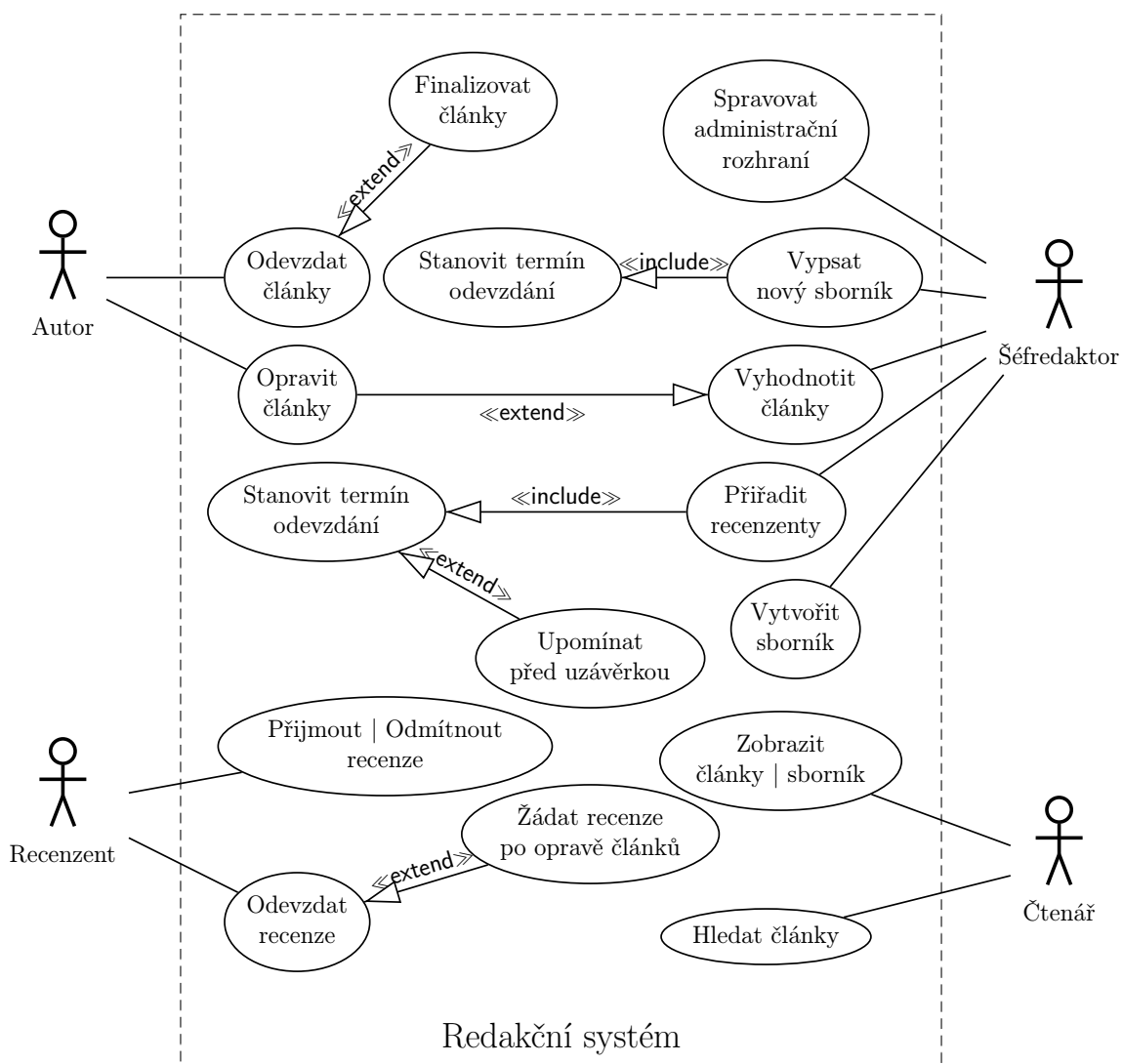
1.1 Požadavky na recenzní řízení

Základem pro analýzu jsou příklady užití, které jsou popisem toho, jak se bude aplikace používat. Případy užití věnují pozornost interakci systému s jeho uživateli. Diagram případů užití na obr.1.1 se skládá z herců, kteří komunikují ze systémem, z ohraničení systému a vztahů mezi jednotlivými případy užití. *Extent* označuje nepovinné rozšíření z jiného případu užití, *include* značí povinnou součást, kdy je případ užití zahrnut do jiného případu užití. V analyzované aplikaci patří mezi herce: šéfredaktor, autor, recenzent a čtenář. Recenzního řízení se mohou účastnit jen registrovaní uživatelé, kteří se musí přihlásit do systému. V další části textu jsou charakterizovány jednotlivé role v průběhu recenzního řízení.

1.1.1 Šéfredaktor

Šéfredaktor se přihlásí do systému uživatelským jménem a heslem administrátora. Stanoví datum zahájení nového recenzního řízení a napíše aktualitu s výzvou pro příjem článků pro nový sborník. Dále stanoví pro autory termín odevzdávání článků. Po uzávěrce nebo pokud jsou články označené jako finální, si přes webové rozhraní vyzvedne odevzdané články a ke každému článku přidělí ze seznamu recenzenta a stanoví termín odevzdání recenzí. Recenzentům se odešlou e-maily s přiděleným článkem v příloze. Poté si šéfredaktor v systému zobrazuje reakci recenzentů na zaslanoou žádost o recenzi článku. Zjistí tak, jestli recenzent přidělenou recenzi článku přijal nebo odmítl. Pokud recenzent odmítne, šéfredaktor přidělí článek jinému recenzentovi. Šéfredaktor může v systému nastavit, před uzávěrkou odevzdání, čas pro odeslání upomínek recenzentům. Po termínu odevzdání si šéfredaktor přes webové rozhraní vyzvedne odevzdané recenze. V případě, že recenzent žádá změny v článku, šéfredaktor stanoví termín odevzdání opraveného článku a autorovi se odešle recenze. Pokud recenzent označil, že požaduje recenzi opraveného článku, stanoví šéfredaktor termín odevzdání i pro recenzenta. Šéfredaktor může nastavit odeslání upomínek pro odevzdání oprav článků a jejich recenzí. Opravené články a jejich nové recenze

si po termínu odevzdání opět vyzvedne přes webové rozhraní. Na základě recenzí šéfredaktor články vyhodnotí, tzn. článek buď přijme nebo odmítne. Z článků, které přijal, se vytvoří nový sborník. Šéfredaktor má přehled o všech uživatelských účtech a může je mazat. Šéfredaktor také může vytvářet a editovat vytvořené aktuality, editovat odevzdané články a vytvořené sborníky, psát a editovat pokyny pro autory a recenzenty.



Obr. 1.1: Diagram případů užití

1.1.2 Autor

Autor se musí nejprve registrovat do systému, kde si zvolí jedinečné přihlašovací jméno, zvolí heslo a vyplní osobní údaje. Po přihlášení do systému má možnost změnit si své osobní údaje, které vyplnil při registraci. Na základě uveřejněné výzvy

šéfredaktora může napsat jeden nebo více článků pro nový sborník. Články může odevzdávat jen přes webové rozhraní systému. Před uzávěrkou může odevzdaný článek smazat a odevzdat opravený. Autor má také možnost označit v systému odevzdaný článek jako finální, čímž potvrdí, že s prací na tomto článku skončil. Po vypršení termínu již autor nemůže přes webové rozhraní nic odevzdávat. Autorovi může být zaslána na jeho e-mail žádost o opravu článku s recenzí v příloze. Opravený článek odevzdá přes webové rozhraní systému.

1.1.3 Recenzent

Recenzent se musí nejprve registrovat do systému, kde si zvolí jedinečné přihlašovací jméno, zvolí heslo a vyplní osobní údaje, které může po úspěšně provedené registraci aktualizovat. Po přihlášení do systému má možnost změnit si své osobní údaje, které vyplnil při registraci. Na e-mailovou adresu recenzent dostává žádosti o vypracování recenzí s článkem v příloze. Po obdržení žádosti se recenzent přihlásí do systému kde označí, jestli přidělenou recenzí přijímá nebo odmítá. Pokud se blíží uzávěrka může dostat e-mailem upomínku. Po vypracování recenze se přihlásí do systému a přes webové rozhraní recenzi odevzdá. Může při tom žádat o opětovnou recenzi opraveného článku. Opravený článek je mu poté doručen na e-mail. Recenzi opraveného článku odevzdá přes webové rozhraní systému.

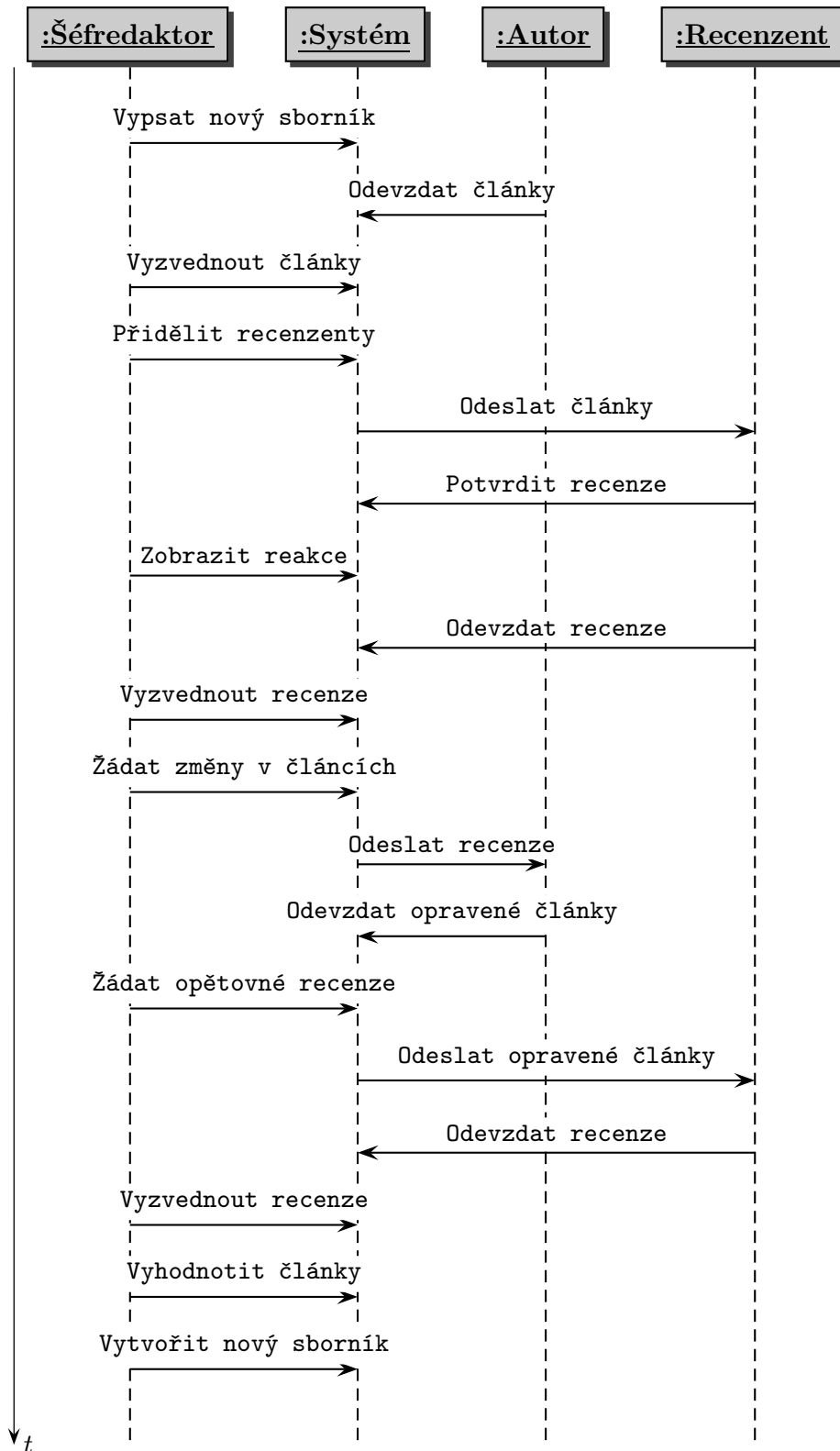
1.1.4 Čtenář

Čtenář je návštěvníkem webu, který si může stahovat sborníky a články, číst anotace k jednotlivým článkům, vyhledávat články, číst aktuality a zveřejněné informace.

1.2 Časová posloupnost recenzního řízení

Diagramy interakcí se zaměřují na události a zobrazují posloupnosti interakcí, tj. komunikaci mezi objekty v určitém pořadí. Objekty jsou zobrazeny v diagramu nahoře, v jedné řadě vedle sebe. Od každého objektu směřuje dolů přerušovaná čára, která vyjadřuje ubíhající čas. Šipky mezi časovými čarami objektů zobrazují zprávy zasílané mezi objekty.

Sekvenční diagram zachycující automatizaci recenzního řízení u vědeckých článků je znázorněn na obr.1.2, kde objekty jsou: šéfredaktor, systém, autor a recenzent. Pokud nebudou recenzenti žádat opravy článků, případně recenze těchto opravených článků, bude mít recenzní řízení kratší průběh než je vyobrazeno na diagramu. Šéfredaktor bude mít možnost přijmout opravený článek i bez nutnosti čekání na vyjádření recenzenta.



Obr. 1.2: Sekvenční diagram UML

2 NÁVRH REDAKČNÍHO SYSTÉMU

Cílem návrhu je vytvoření řešení na základě provedené analýzy. Návrh redakčního systému vychází z architektury MVC, která se skládá z pohledu, řadiče a modelu. Dále je proveden návrh perzistentních tříd pro objektivě-relační mapování na tabulky databáze pomocí rámce Hibernate.

2.1 Architektonický vzor MVC

V této části bude popsán způsob, jakým je MVC implementován v rámci Apache Wicket, pomocí kterého bude webová aplikace vyvíjena.

Implementace architektury MVC ve Wicketu je na obr. 2.1. Skládá se ze tří prvků, každý z nich má v MVC své vlastní poslání:

- Model reprezentuje doménový model a interakce s ním. Doménový model zahrnuje objekty jako jsou uživatelé, články apod. Doménový model obsahuje abstrakce vnějšího světa pro který byl systém vyvinutý.
- Pohled převádí data do podoby vhodné pro zobrazení uživateli. Stará se jak je komponenta zobrazena a dotazuje se modelu na dynamické části stránky.
- Řadič přijímá vstupní data od uživatele. Mohou to být např. hodnoty z textového pole nebo označený zaškrtačkový box, kliknutí uživatele na odkaz nebo na ovládací prvek. Řadič užívá uživatelův vstup k aktualizaci modelu, typicky zachází s navigací stránek a odesílání událostí k dalším komponentám.

U desktopových aplikací je řadič typicky zodpovědný pro odesílání zpráv k pohledu tehdy, když buď model detekuje změny nebo obdrží vstup. Ve webových aplikacích ale pohled zobrazí obsah komponenty jen na základě požadavku uživatele, není tedy potřeba, aby řadič informoval pohled. Je dostatečné když se aktualizují data modelu a pokud je potřeba komponentu zobrazit, pohled použije tyto aktualizovaná data. Jak dále ilustruje obr. 2.1, elementy řadič a pohled tvoří v jedné třídě ve Wicketu komponentu. Jsou tak zodpovědné jak za obsluhování vstupu, tak i za jejich zobrazení.

2.1.1 Apache Wicket

Wicket[8] patří mezi komponentně orientované rámce, které umožňují vytvářet webové aplikace skládáním jednotlivých komponent, které lze opakovaně využít. Vývoj webových aplikací se podobá programování ve Swingu. Stránky a komponenty ve Wicketu jsou reálné Java objekty.

Wicket se tak snaží překlenout mezeru mezi objektově-orientovaným programováním a skutečností, že samotný web je postavený na bezstavovém HTTP protokolu. Wicket poskytuje stavový programovací model založený jen na Javě a HTML.

Wicket odděluje HTML kód od Java kódu a nepřidává žádnou speciální syntaxi pro značkovací jazyk a umožňuje tak vytvořit design webové aplikace nezávisle. Světy HTML a Javy jsou paralelní a asociované pouze pomocí identifikátoru Wicket *id*, který je atributem v HTML a vlastností komponenty v Java třídě.

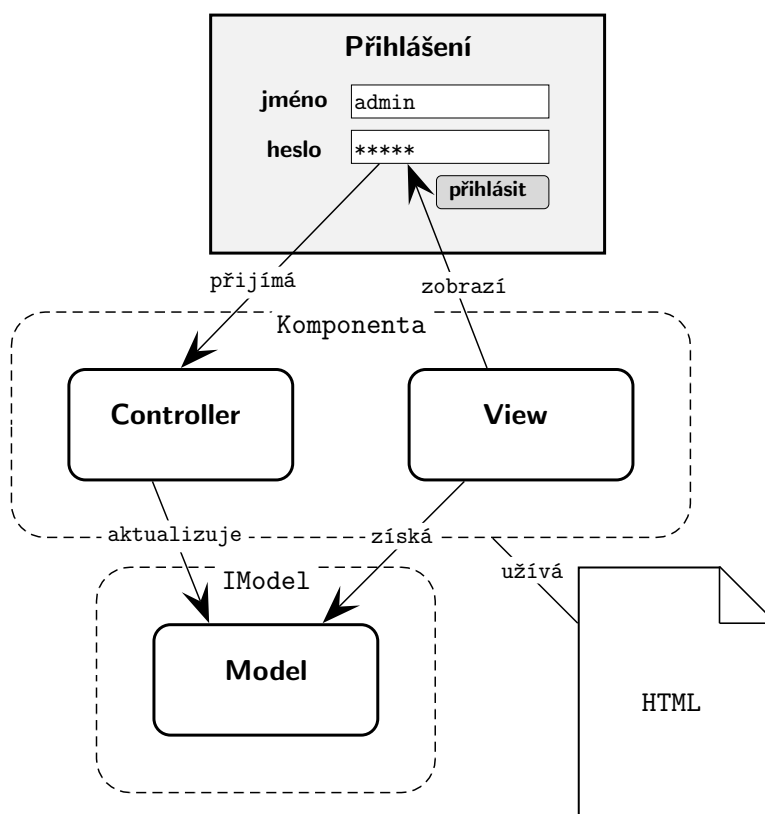
Bezpečnost aplikace je ve Wicketu řešena implicitně. URL neukazuje citlivé informace a všechny vazby komponent jsou v sezení relativní.

Wicket má automatickou podporu pro zpětné tlačítko v prohlížeči. Vrátí objekt stránky do stavu ve kterém byl, když byla stránka poprvé zobrazena.

Ve Wicketu lze jednoduše vytvářet znovupoužitelné komponenty. A to nejen použitím klíčového slova *extends*, ale i vytvořením Panelových komponent, které sdružují skupinu komponent jako znovu použitelnou jednotku.

Jednoduše a flexibilně lze ve Wicketu vytvářet validátory. Stejně tak je jednoduché i lokalizovat výpisy validačních chybových zpráv.

Wicket je velmi rozšířitelný, většina operací je přizpůsobitelných skrz návrhový vzor továrny nebo tovární metody.



Obr. 2.1: MVC architektura v Apache Wicket

Modely ve Wicketu mohou být velmi odlehčené ve smyslu nároků aplikace na paměťové prostředky. Když je model použitý, připojí požadovanou informaci z perzistentního úložiště. Když není model dlouho použitý, dočasná informace je smazána a tím se zredukuje velikost objektu a uvolní se paměť.

Wicket má podporu pro všechny základní HTML rysy, které se používají při tvorbě webových aplikací. Komponenty Wicketu mohou změnit jakýkoliv atribut HTML tagu. HTML stránky, obrázky a zdrojové řetězce mohou být lokalizovány. Wicket má mnoho příkladů, které předvádí mnoho jeho vlastností.

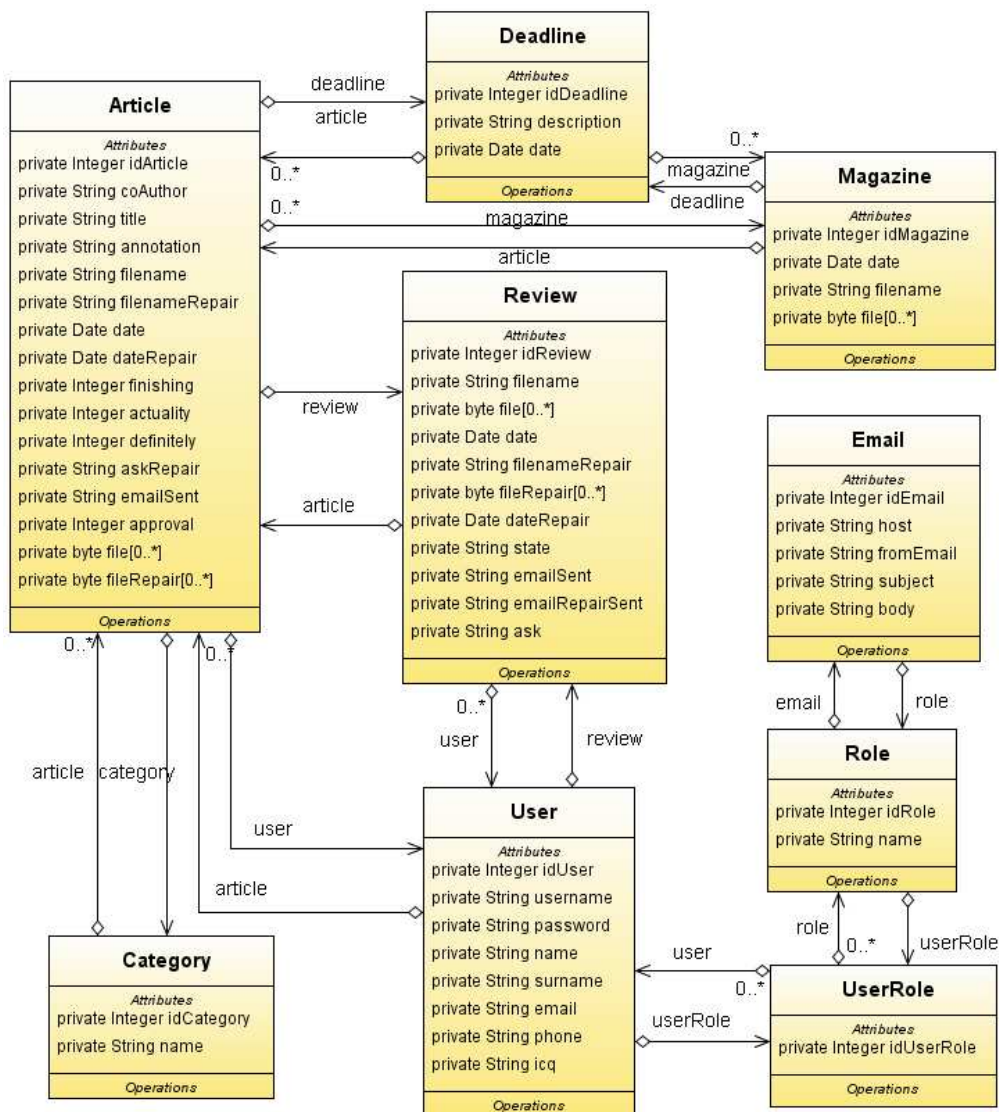
2.2 Návrh perzistentních tříd

Návrh datové vrstvy spočívá ve vytvoření perzistentních tříd (viz obr. 2.2), které se mapují na tabulky v databázi. V tomto mapování se atributy tříd stávají sloupci v tabulce, instance objektů odpovídají řádkům tabulky. Každá třída obsahuje atribut, který je určený pro primární klíč tabulky. V navrženém diagramu tříd jsou zobrazeny vzájemné vztahy mezi třídami a jejich násobnost. Pro tyto asociace se v tabulkách používají cizí klíče.

Třída *User* je určená pro uložení informací o každém registrovaném uživateli. Každý uživatel má jedinečné uživatelské jméno a e-mail. Mezi povinné údaje patří heslo, jméno a příjmení, mezi nepovinné telefon a ICQ. Ze vzájemných vztahů mezi třídami vyplývá, že jeden uživatel může napsat a recenzovat pro jeden sborník více článků. V případě, kdy jeden článek napíše více autorů, odevzdá ho do systému pouze jeden z nich. Uživatel může mít současně v systému více rolí.

Třída *Article* je určená pro ukládání informací o každém článku, aktualitě a pokynech, které může šéfredaktor autorům a recententům napsat. Každý článek má povinně název, anotaci, které jsou v případě publikování článku zveřejněny. Články a jejich opravy se odevzdávají ve formátu PDF. S každým binárním souborem se uloží jeho název a datum odevzdání. Také se ukládají informace, jestli byl článek a jeho oprava finalizována a jména případných spoluautorů článku. Atribut aktualita označuje, zda se jedná o článek či o aktualitu. Pro účely recenzního řízení se ukládá informace, jestli byl článek odeslán recenzentovi, jestli byl autor požádán článek opravit a zda byl nakonec schválen k publikování. Ze vzájemných vztahů mezi třídami vyplývá, že jeden uživatel může napsat více článků, jedna recenze se může vztahovat jen k jednomu článku a jeden sborník se bude skládat z více článků. Vztah mezi třídou *Deadline* značí, že se odevzdané články vztahují k datumu zahájení recenzního řízení. Pomocí tohoto vztahu se odliší, jestli článek patří k aktuálnímu recenznímu řízení nebo těm předešlým. Články se budou také třídit do kategorií.

Třída *Review* slouží k ukládání recenzí a recenzí opravených článků. S každým



Obr. 2.2: Diagram perzistentních tříd

souborem se uloží jeho název a datum odevzdání. Atribut stav označuje, jestli byl přidělený článek recenzentem přijat nebo odmítnut. Ukládají se také informace o tom, jestli byl recenzentovi odeslán e-mail s článkem v příloze, případně s opraveným článkem a také žádost recenzenta opravený článek recenzovat. Ze vzájemných vztahů mezi třídami vyplývá, že jeden uživatel může recenzovat více článků, a že jeden článek může mít jen jednu recenzi.

Třída *Deadline* je určená pro ukládání termínů stanovených pro recenzní řízení. Obsahuje atributy datum a popis. Pro účely recenzního řízení bude třeba ukládat termíny pro odevzdání článků a jejich oprav, recenzí a recenzí opravených článků. Dále termíny pro nastavení odesílání upomínek a to autorům pro odevzdání opravy článku, recenzentům pro odevzdání recenzí a recenzí opravených článků. Důležitým

termínem je také stanovení počátku recenzního řízení, které je určené pro odlišení od předešlého recenzního řízení.

Třída *Email* je určena pro ukládání informací potřebných pro odesílání e-mailů pro autory a také pro recenzenty. Jedná se o e-mailovou adresu, z které se e-maily budou odesílat, odchozí SMTP server, předmět a tělo zprávy. Vztah mezi třídou *Role* znamená, že pro každou roli lze provést jedno nastavení e-mailu.

Třída *Category* slouží k ukládání názvů kategorií článků. Ze vzájemného vztahu s třídou *Article* vyplývá, že jedna kategorie může obsahovat více článků.

Třída *Role* obsahuje seznam rolí pro registrované uživatele. Jedná se o roli Šéfredaktor, který bude zároveň administrátorem, dalšími rolemi jsou Autor a Recenzent. Vztah mezi třídou *UserRole* znamená, že jednu roli může mít více uživatelů.

Pro uložení uživatelských rolí registrovaných uživatelů slouží třída *UserRole*.

2.3 Objektově-relační mapování

Objektově relační mapování umožňuje použití databází v objektově orientovaném programování. Pro tyto účely se používá mapování perzistentních tříd na tabulky v databázi. Pomocí mapování lze uložit data z objektu do databáze anebo naopak data z nich získat. V aplikaci se bude pro tyto účely používat rámec Hibernate.

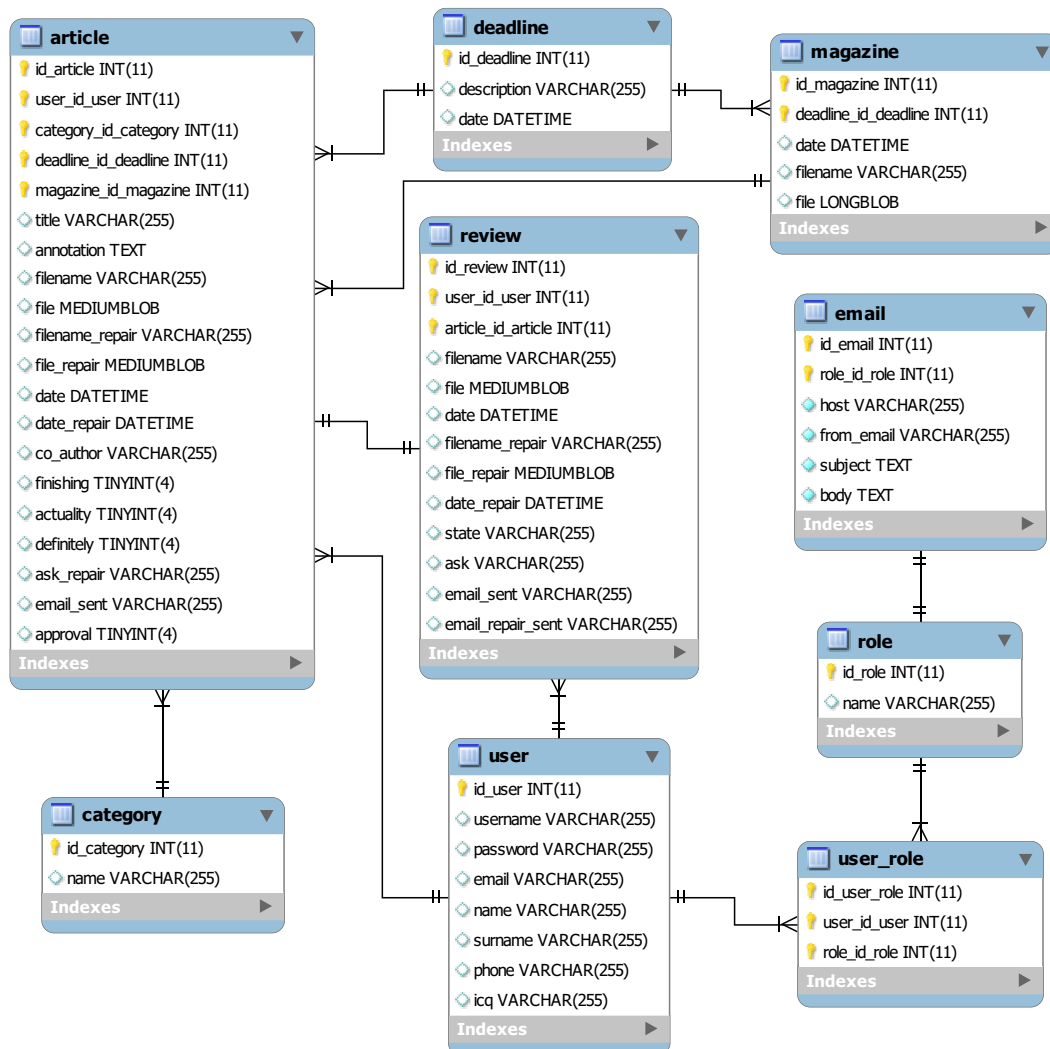
2.3.1 Hibernate

Hibernate[9] je v současnosti nejpoužívanější rámec pro objektově-relační mapování. Hibernate umožňuje vytvářet perzistentní třídy s využitím dědičnosti, asociací, polymorfismu, agregace, kompozice, kolekcí atd. Objekty se mapují buď pomocí anotací anebo mapovacích souborů ve formátu XML. Pro dotazování nad daty uloženými v databázi umožňuje Hibernate použít tři možnosti. Jedná se o Hibernate Query Language, který je navržený jako objektově-orientované rozšíření SQL, Hibernate Criteria Query nebo nativní SQL.

2.3.2 Datový model

Namapované perzistentní třídy na tabulky v databázi jsou zobrazeny pomocí E-R diagramu (viz obr. 2.3). Datový model se od diagramu tříd odlišuje v datových typech, které se odvíjí od použité databáze, v tomto případě se jedná o MySQL. Každá tabulka obsahuje primární klíč, který umožňuje identifikovat záznamy v tabulce a některé i cizí klíče, pro definování vazeb mezi tabulkami. Mezi záznamy dvou tabulek se mohou vyskytnout tři druhy vztahů: 1:1, 1:N, N:M. Znamenají, že jeden řádek v tabulce může souviset s jedním řádkem v druhé tabulce nebo jeden řádek s

více řádky v druhé tabulce. U vztahu N:M může souviset více řádků tabulky s více řádky v druhé tabulce, řeší se to pomocí spojovací tabulky.



Obr. 2.3: E-R diagram

3 IMPLEMENTACE REDAKČNÍHO SYSTÉMU

V této kapitole je vytvořená aplikace představena a popsán způsob její implementace. Aplikaci lze rozdělit na část veřejnou a neveřejnou. V neveřejné části je implementováno základní administrační rozhraní pro změnu osobních údajů, nastavení rolí, správu uživatelů a větší část rozhraní je určena pro organizaci recenzního řízení.

3.1 Veřejná část

Tato část redakčního systému umožňuje všem návštěvníkům webu získat vytvořené sborníky a publikované články. Nabízí se zde také možnost registrace a přihlášení do systému.

3.1.1 Úvodní stránka

Stránka se skládá z levé navigace, z pravé navigace, hlavního panelu a horního menu. V levé navigaci si lze vybrat články podle kategorie, které se zobrazí v hlavním panelu. Pod tímto menu se nabízí nejnovější publikované sborníky ve formátu PDF a odkaz pro zobrazení všech sborníků v hlavním panelu. Podobně pravá navigace obsahuje aktuality a odkaz pro zobrazení všech aktualit. V hlavním panelu, umístěném uprostřed stránky, jsou všechny publikované články, které jsou seřazeny sestupně podle data. Vzhledem k tomu, že všechny články jsou ve formátu PDF se zobrazuje jen název článku, jeho autoři, datum publikování, anotace k článku a odkaz pro jeho stažení. V horním menu je pět položek, první je Úvod pro přesměrování na úvodní stránku. Položka Informace je určena pro informování o pravidlech recenzního řízení. Pro registrované uživatele je určeno Přihlášení a pro neregistrované možnost Registrace. Poslední položka Kontakt je určena pro kontakty na organizátora recenzního řízení. Nad horním menu je v pravé části formulářové okno pro možnost vyhledávání v člancích a jejich autorech. V levé části pak možnost přepínání mezi českou a anglickou verzí webu.

Při implementaci této části redakčního systému bylo třeba vzít do úvahy, že se mimo stránky s registrací a přihlášením uživatele, mění jen obsah hlavního panelu. Z toho důvodu byla vytvořena stránka *HomePage*, která obsahuje implementaci všech opakujících se prvků, tj. levé a pravé navigace, horního menu, vyhledávacího formuláře a odkazy s jazykovou lokalizací webu. Do HTML šablony úvodní stránky je vložen nepárový tag *wicket:child* a to v místě určeném pro měnící se obsah hlavního panelu. Ve všech ostatních stránkách, kromě stránky s registrací a přihlášením, je tak implementován pouze obsah hlavního panelu. Tyto stránky tak již nerozšiřují třídu Wicketu *WebPage*, ale vytvořenou třídu *HomePage*.

REDAKČNÍ SYSTÉM PRO RECENZI ČLÁNKŮ

ENGLISH | **ČEŠTINA**



[ÚVOD](#) | [INFORMACE](#) | [PŘIHLÁŠENÍ](#) | [REGISTRACE](#) | [KONTAKT](#)

KATEGORIE

- ELEKTROTECHNIKA
- FYZIKA
- MATEMATIKA
- PROGRAMOVÁNÍ
- TELEKOMUNIKACE
- OSTATNÍ

SBORNÍKY

1. číslo

Datum: 23.05.2010

[Ke stažení](#)

[Všechny sborníky](#)

ČLÁNKY

Felix millesimus

Autoři: Joe Rosales
Datum: 21.03.2010

Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria.

[Ke stažení](#)

Vobiscum centum

Autoři: Joe Rosales
Datum: 18.03.2010

Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria.

[Ke stažení](#)

Instrumentalis optimus

Autoři: John Viescas
Datum: 17.03.2010

Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria.

[Ke stažení](#)

Tertium diffusus

Autoři: John Viescas
Datum: 12.03.2010

Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria.

[Ke stažení](#)

Nominativus incola

Autoři: Mary Thompson
Datum: 11.03.2010

Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria.

[Ke stažení](#)

AKTUALITY

Datum: 15.05.2010
Coniunctivus neuter

Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria.

Datum: 11.05.2010
Indicativus neuter

Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria.

Datum: 01.05.2010
Abripere interior

Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria. Iustitia manentia femina sedecim clarior gloria.

[Všechny aktuality](#)

<< **12** >>

Copyright © 2010 Redakční systém pro recenzi článků. Všechna práva vyhrazena.

Obr. 3.1: Úvodní stránka

V HTML šablonách těchto stránek je vložený párový tag *wicket:extend* pro definování obsahu hlavního panelu. Vše co se nachází v kódu mimo tento párový tag nemá žádný vliv na vzhled zobrazené stránky.

Všechny výše uvedené části úvodní stránky používají komponentu Wicketu *ListView*, která zajistí, podle pomoci předaného seznamu, výpis aktualit, článků a sborníků. Data komponenta získává pomocí *LoadableDetachableModel*, který neserializuje objekt v modelu. Důsledkem toho se snižuje velikost stavu stránky v paměti, což hraje roli v těch případech, kdy stránku navštíví v jednom okamžiku velké množství uživatelů. Déle je použita komponenta *PageableView*, která zajišťuje stránkování. Výpisy jsou řazeny sestupně vždy podle nejnovějšího data.

Vyhledávání v článcích a autorech je řešeno dotazem nad databází. Výkonnější vyhledávání nabízí některé knihovny, např. projekt Apache Lucene[6]. Odkazy na úvodní stránce jsou optimalizované pro návštěvníky a vyhledávače.

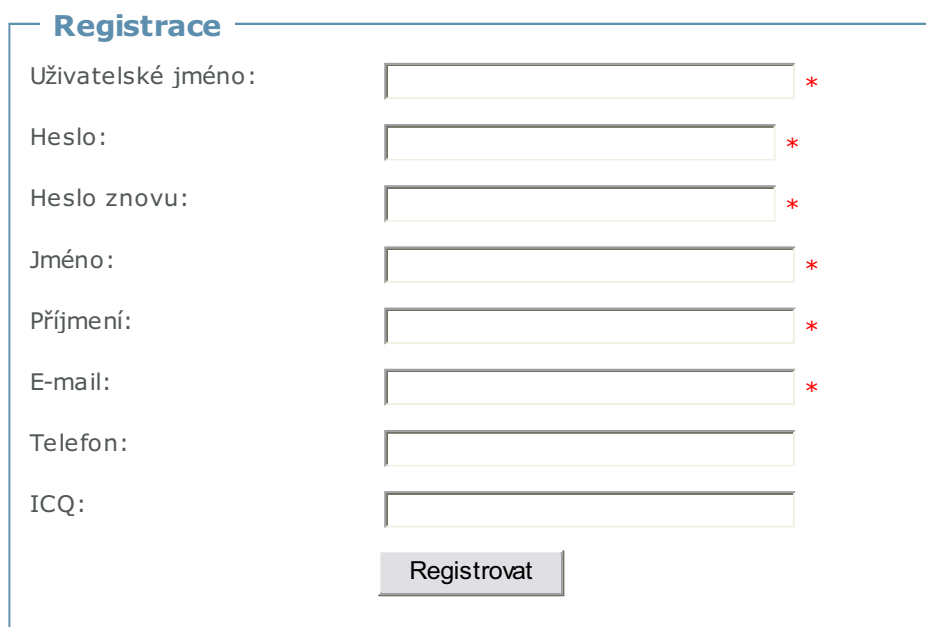
Lokalizace je implementována pomocí *ResourceModel*, které umožňuje připojit ke komponentě lokalizovaný text. Lokalizované zprávy jsou umístěny v souborech s příponou *properties*. Ty jsou pro každou lokalizovanou stránku dva a obsahují dvojice klíč, hodnota. Lokalizace stránek je určena podle nastavení v internetovém prohlížeči. Např. v prohlížeči Firefox lze zvolit jazyky pro zobrazování webových stránek a nastavit pořadí, v jakém se mají používat. Pro přepínání mezi českou a anglickou verzí webu je použita komponenta *Link* s metodou *onClick*, která mění hodnotu proměnné typu *Locale*, která je uložena v sezení. Ve vytvořené aplikaci je lokalizovaná jen úvodní stránka. V analýze recenzního řízení nebyl požadavek na to, aby se články odevzdávaly ve dvou jazykových verzích.

3.1.2 Registrace do systému

Pro registraci uživatelů do systému je určena třída *RegistrationPage*. Tato třída obsahuje šest formulářových komponent *TextField* pro vložení uživatelského jména, křestního jména a příjmení uživatele, e-mailu, telefonu a ICQ. Vložení prvních čtyř údajů je povinné a kontrola se provádí nastavením vlastnosti *Required*. Dalšími dvěma formulářovými komponentami jsou *PasswordTextField* pro zadání uživatelského hesla a pro jeho opětovné zadání z důvodu kontroly. Tyto údaje jsou také povinné a zobrazují se formou hvězdiček.

Z důvodu vhodnější formulace chybové zprávy, při nevyplnění některé položky uživatelem, byl vytvořen soubor *Registration.properties*. V tomto souboru je každá chybová zpráva identifikovaná cestou, která je určena hierarchií příslušné komponenty a nastavenou vlastností *Required*. Tímto způsobem je v tom souboru vytvořen klíč, ke kterému je hodnotou vhodně formulovaný textový řetězec.

V třídě se používá formulářová komponenta *Form*, která je v hierarchii komponent nadřazená všem osmi předchozím komponentám. Jako model používá objekt typu *User* pro uložení dat získaných z formuláře. Pro zobrazení chybové zprávy uživateli slouží komponenta *FeedbackPanel*. Tato třída dále obsahuje ve svých dvou vnitřních třídách vytvořené validátory pro kontrolu jedinečného uživatelského jména a e-mailu. V těchto vnitřních třídách je přepsaná metoda *onValidate*, která tyto údaje kontroluje oproti datům uložených v databázi. V případě e-mailu je ještě použitý *EmailAddressValidator*, který kontroluje, jestli má zadaný řetězec tvar e-mailové adresy. Pro ověření shodnosti zadaných hesel slouží validátor Wicketu *EqualPasswordInputValidator*.



Registrace

Uživatelské jméno: *

Heslo: *

Heslo znovu: *

Jméno: *

Příjmení: *

E-mail: *

Telefon:

ICQ:

Obr. 3.2: Registrace uživatele

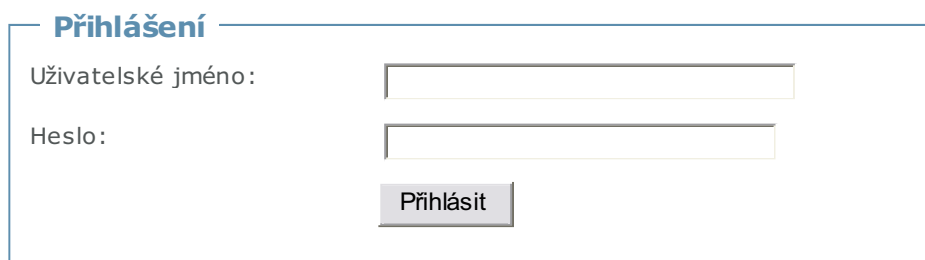
Pro ukládání hesel do databáze se používá *SHA-2*, což je hašovací funkce, která ze vstupního řetězce vytvoří otisk o velikosti 224 bitů. Hašovací funkce poskytuje třída *DigestUtils* z knihovny *commons-codec-1.4*. Před vytvořením haše se k získanému řetězci hesla přidává deset náhodných znaků. Tímto způsobem se zabezpečí situace, kdy uživatel zvolí slabé heslo.

Když uživatel správně vyplní formulářová pole a potvrdí odeslání údajů, volá se metoda *onSubmit* implementovaná v anonymní podtřídě komponenty *Form*. Ta tyto data uloží do databáze a nastaví uživateli roli Autor. Role může uživatel v administračním rozhraní změnit.

3.1.3 Přihlašování do systému

Pro přihlašování uživatelů do systému slouží třída *LoginPage*, která obsahuje formulářovou komponentu *TextField* pro zadání přihlašovacího jména uživatele a formulářovou komponentu *PasswordTextField* pro zadání hesla. Modelem formulářové komponenty *Form*, pro uložení vložených řetězců, je opět objekt typu *User*.

Poté co uživatel vyplní přihlašovací pole formuláře a potvrdí jejich odeslání, zavolá se metoda *onSubmit*. Nejprve se ověří, jestli uživatel zadal existující přihlašovací jméno a poté zda souhlasí heslo. K heslu zadanému do formulářového pole se nejprve přičtou stejné znaky, které byly přičteny i k uloženému heslu. Poté se pomocí funkce *SHA-2* vytvoří z řetězce haš, který se porovná s hašem hesla získaným z databáze. Pokud údaje nesouhlasí, vyvolá se autorizační vyjímka a vypíše se uživateli zpráva, že přihlášení nebylo úspěšné.



The image shows a login form with a blue border. At the top left, the title "Přihlášení" is written in blue. Below the title, there are two rows of labels and input fields. The first row has the label "Uživatelské jméno:" followed by a white rectangular input field. The second row has the label "Heslo:" followed by another white rectangular input field. Below these two input fields, centered horizontally, is a grey button with the text "Přihlásit" in white.

Obr. 3.3: Přihlášení uživatele

Pokud údaje souhlasí, získá se objekt typu *User*, který se použije pro vytvoření nového sezení. Nastaví se uživatelské role. V případě, že uživatel nebyl na stránku s přihlášením přeměrován ze stránky, na kterou neměl povolený přístup, je uživatel přeměrován do administračního rozhraní na stránku *MyAccountPage*, pomocí metody *setResponsePage*.

3.2 Administrační rozhraní

Administrační rozhraní obsahuje v horním menu položky Můj účet, Autor, Recenzent, Šéfredaktor a vpravo od nich se zobrazuje uživatelské jméno a odkaz pro odhlášení se ze systému. Každý přihlášený uživatel má přístup do sekce Můj účet, do sekce Autor a Recenzent podle nastavené role. Do sekce Šéfredaktor má přístup pouze jedna osoba, která řídí recenzní řízení a spravuje redakční systém. V levém menu lze změnit osobní údaje a role. Šéfredaktorovi se zobrazuje možnost spravovat uživatelské účty.

Pro zobrazení uživatelského jména slouží komponenta *Label* a pro odhlášení uživatele ze systému komponenta *Link*. Poté co uživatel klikne na odkaz, zavolá se metoda *onClick*. V ní se nejprve získá sezení, na které se zavolá metoda *invalidate*, která ho smaže. Pomocí metody *setResponsePage* se aplikace přesměruje na stránku s přihlášením do systému *LoginPage*.

3.2.1 Změna osobních údajů

Stránka *SettingUserPage* je určená ke změnám osobních údajů, které uživatel vyplnil při registraci. Všechna data se kromě hesla načtou z databáze do formulářových polí a změnit lze všechny položky mimo uživatelského jména. Před uložením změněných údajů je vyžadováno vložení aktuálního hesla.

ADMINISTRAČNÍ ROZHRAŇÍ

MŮJ ÚČET | AUTOR | RECENZENT | ŠÉFREDAKTOR

MENU

- OSOBNÍ ÚDAJE
- NASTAVENÍ ROLE

Informace o účtu

Uživatelské jméno:	author
Nové heslo:	<input type="password"/> *
Heslo znovu:	<input type="password"/> *
Jméno:	<input type="text" value="Mary"/> *
Příjmení:	<input type="text" value="Thompson"/> *
E-mail:	<input type="text" value="author@mail.cz"/> *
Telefon:	<input type="text" value="123456789"/>
ICQ:	<input type="text" value="123-456"/>
Současné heslo:	<input type="password"/> *

Copyright © 2010 Redakční systém pro recenzi článků. Všechna práva vyhrazena.

Obr. 3.4: Změna osobních údajů

3.2.2 Změna role

Stránka *SettingRolePage* umožňuje měnit uživatelské role autorům a recenzentům. Každý registrovaný uživatel má automaticky nastavenou roli Autor. Uživatel má možnost mít obě tyto role anebo nemít role žádné.

Změna rolí je implementována pomocí komponenty *RadioChoice*, která vytvoří přepínač s volbami ano/ne pro obě role. Pro roli šéfredaktor je komponenta nastavená jako neviditelná, pomocí přepsané metody *isVisible*. Šéfredaktor si nepotřebuje nastavovat role, má přístup na všechny stránky. V metodě *onSubmit* se nejprve zjišťuje, jestli si uživatel roli přidává, ruší anebo ponechal vše beze změn. Jako model pro vstupní data slouží statická vnitřní třída *Input*. Po změně rolí se uživateli zpřístupní odpovídající položky v horním menu. Stránky jsou opatřeny anotací, čímž se zabráňuje nepovolenému přístupu uživatele bez příslušné role.



Uživatelské role

Autor: ano ne

Recenzent: ano ne

Obr. 3.5: Změna uživatelských rolí

3.2.3 Uživatelské účty

Šéfredaktor je administrátorem systému a může si zobrazit seznam všech registrovaných uživatelů, se všemi jejich vloženými osobními údaji a uživatelskými rolemi. Šéfredaktorovi se také nabízí možnost vytvořené účty mazat a měnit uživateli heslo v případě, že ho zapomněl. Původní heslo již nelze výpočtem získat, protože funkce *SHA-2* je jednosměrná.

Registrovaní uživatelé se přehledně zobrazují v tabulce na stránce *UsersManagementPage*. Nejedná se o HTML tabulku, o čemž se lze přesvědčit při smazání cookies na této stránce, ale o využití komponenty Wicketu *DataTable*. Tato komponenta pracuje s objektem *SortableDataProvider* a s *PropertyColumn* objekty, které naplní buňky tabulky obsahem. Počet *PropertyColumn* objektů je daný počtem sloupců v tabulce, které chceme zobrazit. Podle nastaveného počtu řádků v tabulce se vytvoří navigátor, kterým se lze přepínat v tabulce, stejně jako u navigátoru stránek. *SortableDataProvider*, jak už jeho název napovídá, podporuje třídění položek v tabulce. Pro každý osobní údaj uživatele je implementovaná funkce *compare* z rozhraní *Comparator*.

Tabulka obsahuje tři speciální sloupce, pro které nelze použít *PropertyColumn*, protože může zobrazit jako obsah buňky jen text. Pro jakýkoliv jiný obsah buňky je třeba použít *AbstractColumn*, který je rodičovskou třídou *PropertyColumn*. Odkazy

na smazání každého uživatele jsou implementované v třídě panelu *DeleteUserPanel*, který obsahuje komponentu *Link*, která smaže řádek tabulky pomocí předaného objektu řádku tabulky.

Druhým speciálním sloupcem tabulky jsou modální okna, která zobrazí u každého uživatele jeho role. Implementace je v třídě panelu *ModalRolePanel*, který obsahuje komponentu *AjaxLink* s metodou *onClick*, která pomocí metody *show* zobrazí obsah. Pro nastavení hesla je vytvořené druhé modální okno, které obsahuje dvě formulářová okna, pro zadání hesla a jeho ověření. Implementace je v třídě *UserPasswordPanel*.

Počet řádků v tabulce je přednastavený na pět a lze ho měnit pomocí výběrové komponenty pod tabulkou. Nabízí se možnost zvyšování počtu řádků po pěti až do čtyřiceti. Implementace komponenty *DropDownChoice* je v panelu *SelectNumberPanel*. Zde je metoda *wantOnSelectionChangedNotifications* nastavena, aby vracela pravdivou hodnotu a volala přepsanou metodu *onSelectionChanged*, která změní počet zobrazených řádků tabulky podle výběru uživatele. Tento způsob implementace komponenty tak nepotřebuje mít potvrzující tlačítko výběru.

ID	Uživatelské jméno	Jméno	Příjmení	Heslo	Role	E-mail	Telefon	ICQ	
2	author	Mary	Thompson	změnit	zobrazit	author@mail.cz	123456789	123-456	Smazat
3	reviewer	Will	Thompson	změnit	zobrazit	reviewer@mail.cz	789123774	456-787	Smazat
4	author2	John	Viescas	změnit	zobrazit	author2@mail.cz	858723698	777-658	Smazat
5	author3	Joe	Rosales	změnit	zobrazit	author3@mail.cz	335478851	991-654	Smazat
6	reviewer2	Rachel	Patterson	změnit	zobrazit	reviewer2@mail.cz	552153125	811-654	Smazat
7	reviewer3	Bailey	Hallmark	změnit	zobrazit	reviewer3@mail.cz	441123147	788-985	Smazat

Vyberte počet řádků

Změna hesla

Heslo: *

Heslo znovu: *

[Zavřít okno](#)

Obr. 3.6: Správa uživatelských účtů

3.3 Recenzní řízení

V následující části jsou popsány všechny stránky, které byly vytvořeny pro jednotlivé uživatelské role, potřebné k organizaci recenzního řízení.

3.3.1 Recenzní řízení pro roli Autor

Autoři článků mají možnost odevzdávat a editovat články, případně odevzdávat na vyžádání šéfredaktora opravy svých článků. Na úvodní stránce *AutorPage* se autorům zobrazí pokyny, pokud je šéfredaktor vytvořil. Na této stránce se kontrolují nastavené termíny recenzního řízení oproti aktuálnímu datumu. Podle toho se autorům článků zpřístupní stránka pro odevzdávání článků nebo jejich oprav společně se stránkou pro editaci článků. Pokud je již po termínech odevzdávání, tak se zobrazí stránka s upozorněním, že je po uzávěrce. Internetové odkazy jsou implementovány pomocí komponenty *BookmarkablePageLink*.

3.3.2 Odevzdání článku

Před samotným odevzdáním souboru je povinné vepsat název článku do formuláře. Název je možné před ukončením uzávěrky upravit. Poté se vybere kategorie do které článek patří. Volitelně lze zatrhnout možnost pro finalizaci článku. Velikost souboru nesmí přesáhnout nastavenou hodnotu 5MB.

The image shows a web form for submitting an article, divided into five distinct sections:

- Název článku**: A text input field with a small icon on the right and a red asterisk below it, indicating it is required. A "Potvrdit" button is located below the field.
- Kategorie článku**: A dropdown menu with the text "Vyberte jeden" and a downward arrow. A "Vybrat" button is positioned below the dropdown.
- Soubor ve formátu PDF**: A section with the sub-label "Maximální velikost souboru: 5MB". It contains a file input field with a "Procházet..." button to its right, and an "Odevzdat" button below the field.
- Finalizace článku**: A section containing a single checkbox.
- Odevzdané články**: A section displaying a list of submitted files: "'soubor-2.pdf' [Smazat](#)" and "'soubor-1.pdf' [Smazat](#)".

Obr. 3.7: Odevzdání článku

Pokud autor nesplní některou z uvedených podmínek odevzdání, tak je upozorněn chybovou zprávou. Pokud je uložení souboru úspěšné, tak se zobrazí název souboru a odkaz pro jeho smazání. Autor může odevzdat více souborů, předpokládá se, že každý odezdá na jiné téma. Provádí se pouze kontrola, jestli autor neodevzdává soubor se stejným názvem a v tom případě je upozorněn na nutnost smazání původně odevzdaného souboru.

Implementace odevzdání článku je ve třídě *UploadArticlePage* a to v její vnitřní třídě *UploadForm*, která je potomkem třídy Wicketu *Form*. Stránka obsahuje i druhou vnitřní třídu *ListFiles* pro zobrazení a mazání odevzdaných souborů. Protože není dopředu známé kolik těchto souborů bude, je zde použit jeden z opakovačů Wicketu a to *ListView*. Jako parametr se mu předá seznam názvů odevzdaných souborů, které se předtím získaly z databáze. Dále se pomocí implementace metody *populateItem* vytvoří seznam komponent pro výsledné zobrazení na stránce. Opakovače tedy umožňují opakované zobrazení komponent. V tomto případě je to komponenta *Label* pro název souboru a *Link* k vytvoření odkazu pro smazání souboru.

Třída *UploadRepairArticlePage* umožňuje autorům odevzdávat opravené články. Na této stránce je třeba určit název původního souboru a to z důvodu možnosti odevzdání více oprav článků. Pro tyto účely je použita výběrová komponenta *DropDownChoice* se seznamem původně odevzdaných souborů autora. V případě že autor nemá vypracovat opravu článku, je nabídka prázdná a nemá možnost tak nic odevzdávat. Kategorie článku se zde již podruhé nevybírání. Autoři mají možnost finalizovat odevzdání anebo opravený článek do uzavěrky smazat.

3.3.3 Editace článku

Vzhledem k tomu, že se články odevzdávají ve formátu PDF, je možné editovat pouze ty údaje o článku, které jsou určeny k zveřejnění na úvodní stránce. Nejprve je třeba pomocí rozbalovací nabídky vybrat odevzdaný článek a po jejím potvrzení se načtou do tří textových polí údaje uložené v databázi a přiřazená kategorie. Tu lze změnit v další výběrové nabídce. Autor může změnit název a anotaci článku a případně jména spoluautorů článku. Stránka s editací je zpřístupněna pouze v termínu pro odevzdání článků anebo jejich oprav. Implementace stránky je vyřešena pomocí formulářových komponent *TextArea* a *DropDownChoice*.

3.4 Recenzní řízení pro roli Recenzent

Recenzenti v administračním rozhraní označí svůj souhlas nebo odmítnutí přidělené recenze, odevzdávají vypracované recenze, případně odevzdají recenzi opraveného

článku. Na úvodní stránce *ReviewerPage* se kontrolují šéfredaktorem nastavené termíny oproti aktuálnímu datumu. Podle toho se recenzentům zpřístupní stránka pro odsouhlasení a odevzdávání recenze nebo stránka pro odevzdání recenze opraveného článku. Pokud je již po termínech odevzdání, tak se pouze zobrazí stránka s upozorněním, že je po uzávěrce nebo, že již nelze vyjadřovat souhlas k přidělené recenzi. Internetové odkazy jsou implementovány pomocí komponenty *BookmarkablePage-Link*.

Odevzdaný článek

soubor-2.pdf ▾

Vybrat

Kategorie

Současná kategorie článku: Fyzika

Vyberte jeden ▾

Vybrat

Titul

Nominativus incola

*

Anotace

Iustitia
manentia femina sedecim clarior gloria. Iustitia
manentia femina
sedecim clarior gloria. Iustitia manentia femina
sedecim clarior
gloria. Iustitia manentia femina sedecim clarior
gloria. Iustitia
manentia femina sedecim clarior gloria. Iustitia
manentia femina
sedecim clarior gloria.

*

Spoluautor

Změnit

Obr. 3.8: Editace článku

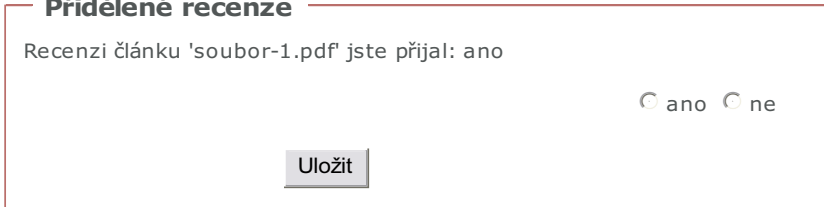
3.4.1 Odevzdání recenze

Odevzdání recenzí je implementováno ve třídě *UploadReviewPage*. Nejprve je třeba vybrat název článku ke kterému má být recenze odevzdána. V případě, že recenzent nemá přidělenou recenzi článku, je rozbalovací nabídka prázdná a nemá možnost nic odevzdávat. Recenzent má možnost označit v zaškrtačacím poli, že žádá opravený článek znovu recenzovat. Zaškrtačací pole je řešené komponentou *CheckBox*. Odevzdanou recenzi lze do uzávěrky smazat a znovu odevzdat.

Třída *UploadRepairReviewPage* slouží k odevzdání recenze opraveného článku. Stránka vypadá stejně jako stránka pro odevzdání první recenze, neobsahuje však již zaškrtačací pole pro žádost o recenzi opravy článku.

3.4.2 Odsouhlasení recenze

Recenzent by měl, po obdržené e-mailové žádosti, potvrdit svůj souhlas s přidělenou recenzí na stránce *AcknowledgeReviewPage*. Vzhledem k tomu, že recenzent může mít přidělených více recenzí, byl na této stránce implementován opakovač *List-View*. Jako parametr se mu předává seznam názvů odevzdaných článků získaných z databáze. V metodě *populateItem* tvoří seznam komponent *Label* pro název souboru a *RadioChoice* pro přepínač. Přepínač je implementován v samostatné třídě *AcknowledgeReviewPanel*. Přepínač umožňuje recenzetovi své rozhodnutí změnit. Jeho volba se zobrazuje šéfredaktorovi na stránce s přidělováním recenzí. Komponenta *Label* pak recenzetovi zobrazuje jeho uloženou volbu.



Přidělené recenze

Recenzi článku 'soubor-1.pdf' jste přijal: ano

ano ne

Uložit

Obr. 3.9: Odsouhlasení recenze

3.5 Recenzní řízení pro roli Šéfredaktor

Šéfredaktor organizuje celé recenzní řízení počínaje od nastavení termínů, přidělení recenzentů k článkům, vyhodnocení článků, žádání oprav až po vytvoření sborníku. Dále může vytvořit a editovat aktuality, nastavovat parametry pro automatické

odesílání e-mailů, upomínek, psát pokyny autorům a rezententům, editovat články a sborníky.

3.5.1 Vytvoření aktuality

Vytvoření aktualit je implementováno ve třídě *ActualityCreationPage* a obsahuje dvě textové pole *TextArea*, jejichž vyplnění je nastavené jako povinné. Vytvořená aktualita se uloží do databáze a je zobrazována na úvodní webové stránce jako první v pořadí. Po uplynutí jednoho měsíce od vydání se aktualita zobrazuje jen v seznamu všech aktualit.

3.5.2 Editace aktuality

Vytvořené aktuality může šéfredaktor editovat na stránce *ActualityEditPage*. Aktuality se mu nabízejí v rozbalovací nabídce a to podle datumu od nejnovější aktuality. Po potvrzení vybrané aktuality se načte z databáze její název a obsah. Vybranou aktualitu může také smazat. Výběr vytvořených aktualit je nabízen pomocí komponenty *DropDownChoice* a pro editaci se používá komponenta *TextArea*.

The image shows three distinct UI sections for managing news items:

- Vytvořené aktuality:** A section with a dropdown menu showing the date '2010/05/15 16:43:26' and a 'Vybrat' button.
- Editace aktuality:** A section with two text input fields. The first is labeled 'Název:' and contains 'Coniunctivus neuter'. The second is labeled 'Obsah:' and contains two lines of Latin text: 'Iustitia manentia femina sedecim clarior gloria.' and 'Iustitia manentia femina sedecim clarior gloria.'. There are small purple icons on the right side of both text areas. Below the fields is a 'Změnit' button.
- Smazání aktuality:** A section with a single 'Smazat' button.

Obr. 3.10: Editace aktuality

3.5.3 Nastavení termínu

Když šéfredaktor vypíše výzvu pro příjem příspěvků pro nový sborník, měl by nastavit termíny na stránce *SetDeadlinePage*. Nabízí se celkem pět termínů pro nastavení. Především je to datum a čas vypsání nového sborníku. Toto datum je důležité, protože je potřeba odlišit aktuální recenzní řízení od předešlých. Pokud se toto datum nenastaví, tak se předpokládá, že předchozí recenzní řízení ještě neskončilo. Datum se využívá při vyhledávání dat v databázi, např. aby se na stránce pro přidělení recenzentů šéfredaktorovi zobrazily jen nově odevzdávané články. Dále se jedná o dva termíny, které určí do kdy mohou autoři odevzdávat články a jejich opravy. Další dva termíny určují do kdy mohou recenzenti odevzdávat recenze a recenze opravených článků.

Vyberte termín

Autor

Datum

Hodina

23

Minuta

05

Nastavit termín

Změnit

Současné nastavení

Začátek recenzního řízení:	[2010-03-01 03:01:00]
Autor:	[2010-05-23 02:02:00]
Recenzent:	[2010-05-25 04:00:00]
Autor oprava:	[2010-05-25 02:02:00]
Recenzent oprava:	[2010-05-27 04:04:00]

Obr. 3.11: Nastavení termínu

Nastavení datumu se provádí pomocí komponenty Wicketu *DatePicker*, kde se kliknutím na ikonu otevře okno s kalendářem pro snadný výběr datum. Pro výběr hodiny a minuty byly zvoleny komponenty *DropDownChoice*. Rozbalovací nabídka tak nabízí jen rozsah možných hodnot. Tímto odpadají problémy, které by se musely v aplikaci řešit, pokud by je uživatel zadával sám do textového pole. Pro konverzi

řetězce na datum slouží metoda *stringToDate* v třídě *DateTimeUtil*. Všechny komponenty jsou nastavené jako povinné. Nastavené termíny se zobrazují na stránce.

3.5.4 Nastavení e-mailu

V průběhu recenzního řízení dochází k odesílání e-mailů recenzentů a autorům. Na stránce *SetEmailPage* se nastavuje pro obě role odchozí server, odchozí e-mail, předmět zprávy a tělo zprávy. Při výběru role se do textových polí načtou údaje uložené v databázi.

Na této stránce se také zobrazují aktuálně nastavené údaje. Implementace této stránky je řešena výběrovou komponentou *DropDownChoice* a komponentami textové pole *TextField* a textová oblast *TextArea*.

3.5.5 Přidělení recenze

Na stránce *AssignReviewerPage* přiděluje šéfredaktor odevzdané články recenzentům. Stránka je řešená tabulkou stejně jako stránka s uživatelskými účty. Z levé strany jsou první tři sloupce určeny pro identifikační číslo, jméno a příjmení autora. Následuje identifikační číslo článku, odkaz na detail, po kliknutí se otevře modální okno s názvem a anotací článku, další odkaz nabízí možnost stažení článku a informace, jestli autor odevzdání finalizoval či ne. Za sloupcem finalizace se nachází potvrzující tlačítko, které zde slouží pro přechod na stránku se seznamem recenzentů, který je řešen také pomocí tabulky.

IDA	Jméno	Příjmení	ID Č	Detail	Článek	Finalizace		ID R	Jméno	Příjmení	Odesláno	Přijato
2	Thompson	Mary	19	zobrazit	soubor-1.pdf	ano	<input type="button" value="Vybrat"/>	3	Will	Thompson	[ano]	ano
2	Thompson	Mary	20	zobrazit	soubor-2.pdf	ne	<input type="button" value="Vybrat"/>	0			[]	
4	Viescas	John	21	zobrazit	soubor-3.pdf	ano	<input type="button" value="Vybrat"/>	7	Bailey	Hallmark	[ano]	ano
4	Viescas	John	22	zobrazit	soubor-4.pdf	ne	<input type="button" value="Vybrat"/>	0			[]	
5	Rosales	Joe	23	zobrazit	soubor-5.pdf	ano	<input type="button" value="Vybrat"/>	0			[]	
5	Rosales	Joe	24	zobrazit	soubor-6.pdf	ne	<input type="button" value="Vybrat"/>	0			[]	

Odeslat přiděleným recenzentům e-mailů

Obr. 3.12: Přidělení recenze

Na stránce *ListReviewerPage* se zobrazují kromě hesla všechny údaje o recenzentech. Na konci každého řádku je potvrzovací tlačítko, které provede výběr a přesměrování zpět na stránku *AssignReviewerPage*. Zde se do příslušného řádku tabulky

načte identifikační číslo, jméno a příjmení vybraného recenzenta. Předposlední sloupec zobrazuje informaci o tom, jestli byl odeslán článek e-mailem a poslední sloupec, jestli recenzi přijímá nebo ne.

IDR	Jméno	Příjmení	E-mail	Telefon	ICQ	
3	Will	Thompson	reviewer@mail.cz	789123774	456-787	Přiřadit
6	Rachel	Patterson	reviewer2@mail.cz	552153125	811-654	Přiřadit
7	Bailey	Hallmark	reviewer3@mail.cz	441123147	788-985	Přiřadit

Obr. 3.13: Seznam recenzentů

Pod tabulkou je potvrzovací tlačítko pro odeslání e-mailu. Pokud je tlačítko potvrzeno, ikdyž už byly e-maily odeslány, tak se jen vypíše informační zpráva, že k odeslání nedošlo, protože nebyl přiřazen nový recenzent. To umožňuje e-maily odesílat postupně, aniž by se odeslal stejnému recenzentovi e-mail vícekrát. Všechny sloupce, jejichž obsahem je text, obsahují odkaz na řazení položek. Šéfredaktor si tak může např. seřadit články podle finalizace nebo zjistit, jestli byly odeslány všechny e-maily.

Implementace odkazu pro stažení článku je ve třídě *DownloadArticlePanel*. V tomto panelu je použita komponenta *Label* pro zobrazení názvu souboru a *Link* pro získání proudu bajtů. Potvrzovací tlačítko pro přechod na stránku s tabulkou recenzentů je implementované v třídě panelu *AssignReviewerPanel*. Pomocí metody *setResponsePage* dojde k přesměrování na stránku *ListReviewerPage* se seznamem recenzentů. Tabulka obsahuje textové položky a jedno potvrzovací tlačítko, které je implementované v třídě *ListReviewerPanel*. Tomuto panelu se předává objekt typu *User*, získaný z vybraného řádku tabulky. Po potvrzení výběru se tak uloží do databáze nově přidělená rezece a dojde k přesměrování zpět na stránku *AssignReviewerPage*, kde se vybraný recenzent zobrazí v řádku tabulky.

Implementace odesílání e-mailů s přílohou je ve třídě *EmailWithAttachment*, kde se nejprve získá z databáze soubor k odeslání a také nastavení e-mailu podle role uživatele. Poté se vytvoří objekt typu *Message*, v kterém se postupně nastaví všechny potřebné parametry pro odeslání e-mailu. Pokud je e-mail úspěšně odeslán, tak se tato informace uloží.

3.5.6 Recenze a opravy článků

Stránka *ArticleReviewsPage* navazuje v recenzním řízení na stránku s přidělením recenzentů. Stránka je řešená pomocí dvou tabulek zobrazených nad sebou. V první

tabulce jsou zleva první tři sloupce určeny pro identifikační číslo, jméno a příjmení recenzenta. Následuje odkaz pro stažení odevzdané recenze. Další tři sloupce slouží pro identifikační číslo, jméno a příjmení autora článku. Dále je potvrzovací tlačítko, kterým se žádá po autorovi oprava. Nelze žádat opravu dokud není odevzdaná recenze. Potvrzená žádost se zobrazí ve vedlejším sloupci a v posledním sloupci je zobrazena informace, jestli byl autorovi zaslán e-mail s recenzí v příloze.

V druhé tabulce jsou zleva první tři sloupce určeny pro identifikační číslo, jméno a příjmení autora. Další je odkaz pro stažení opraveného článku. Následují tři sloupce pro identifikační číslo, jméno a příjmení recenzenta. V dalším sloupci se zobrazuje, jestli recenzent žádá recenzovat opravený článek. Následuje informace o tom, jestli byl článek odeslán a v posledním sloupci odkaz pro stažení recenze opraveného článku. Pod oběma tabulkami jsou potvrzovací tlačítka pro odeslání e-mailu. Všechny sloupce, jejichž obsahem je text, obsahují také odkaz na řazení položek.

ID R	Jméno	Příjmení	Recenze	ID A	Jméno	Příjmení		Žádat opravu	Odesláno
3	Will	Thompson		2	Thompson	Mary	Změnit	ano	[ano]
0				2	Thompson	Mary	Změnit		[ne]
7	Bailey	Hallmark	recenze-2.pdf	4	Viescas	John	Změnit	ano	[ano]
0				4	Viescas	John	Změnit		[ne]
0				5	Rosales	Joe	Změnit		[ne]
0				5	Rosales	Joe	Změnit		[ne]

Odeslat autorům recenze k provedení oprav

Odeslat

ID A	Jméno	Příjmení	Oprava	ID R	Jméno	Příjmení	Poslat opravu	Odesláno	Recenze
2	Thompson	Mary	oprava-1.pdf	3	Will	Thompson	ano	[ano]	
2	Thompson	Mary		0				[]	
4	Viescas	John	oprava-2.pdf	7	Bailey	Hallmark	ne	[ne]	
4	Viescas	John		0				[]	
5	Rosales	Joe		0				[]	
5	Rosales	Joe		0				[]	

Odeslat recenzentům opravené články

Odeslat

Obr. 3.14: Recenze a opravy článků

Při implementaci této stránky byly vytvořeny tři panely. Pro odkaz na stažení recenzí, pro stažení opravených článků a pro uložení žádosti o opravu článku.

3.5.7 Editace instrukcí

Stránka *InstructionEditPage* umožňuje šéfredaktorovi upravovat obsah stránky Informace a Kontakt, které mají odkaz v horním menu na úvodní stránce a dále upravovat pokyny pro autory a recenzenty, které se zobrazují v administračním rozhraní. Implementace je řešena výběrovou komponentou *DropDownChoice* a dvěma textovými komponentami *TextArea*.

3.5.8 Nastavení upomínky

Šéfredaktor má možnost na stránce *SetReminderPage* nastavit tři termíny pro odeslání upomínek. Upomínky se mohou odesílat před odevzdáním recenzenzí, před odevzdáním oprav článků a před odevzdání recenzí opravených článků. Šéfredaktor může nastavené termíny vymazat, pokud nechce upomínky odesílat.

Vyberte upomínku

Recenzent upomínka

Vyberte datum

květen 2010

Po	Út	St	Čt	Pá	So	Ne
26	27	28	29	30	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

Vyberte hodinu

07

Vyberte minutu

07

Nastavit upomínku

Změnit

Současné nastavení upomínek

Recenzent:	[2010-05-23 13:45:00]	Smazat
Autor oprava:	[2010-04-19 23:40:00]	Smazat
Recenzent oprava:	[]	Smazat

Obr. 3.15: Nastavení upomínky

Implementace časovače je ve třídě *TimerForReminder*, která rozšiřuje třídu *TimerTask*. Vytvořeným objektem *Timer* se naplňuje úloha pomocí metody *schedule*, které se předá jako parametr termín pro odeslání upomínky a objekt třídy *TimerTask*. Vytvořené vlákno má pak jediný úkol a tím je odeslání e-mailu s upomínkou.

Implementace odesílání e-mailů bez přílohy je ve třídě *EmailWithoutAttachment*. Oproti třídě *EmailWithAttachment* je implementace odesílání e-mailů zjednodušená.

3.5.9 Vyhodnocení článků

Na stránce *AssortArticlePage* provádí šéfredaktor vyhodnocení všech odevzdaných článků. U každého řádku tabulky se zobrazuje identifikační číslo článku, odkaz na detail, který otevře modální okno s informacemi o názvu článku a jeho anotaci, identifikační číslo, jméno a příjmení autora.

První sloupec vlevo obsahuje zaškrťovací pole, pomocí kterých šéfredaktor vyznačí články určené k publikování. Nad tabulkou je jedno centrální zaškrťovací pole, kterým se označí nebo odznačí všechny řádky tabulky. Po potvrzení výběru lze vybrané články zobrazit kliknutím na odkaz pod tabulkou. Otevře se modální okno, kde se zobrazí stejná tabulka, ale jen s vybranými články. Pod odkazem se na této stránce ještě nachází potvrzovací tlačítka pro vytvoření nového sborníku. Pokud pro toto recenzní řízení byl již sborník vytvořen, tak se nový nevytvoří, ale jen se přepíše stávající sborník.

Vybrat/Odebrat všechny články

Vybrat	ID Č	Detail	ID A	Jméno	Příjmení
<input checked="" type="checkbox"/>	19	zobrazit	2	Mary	Thompson
<input checked="" type="checkbox"/>	20	zobrazit	2	Mary	Thompson
<input checked="" type="checkbox"/>	21	zobrazit	4	John	Vescas
<input checked="" type="checkbox"/>	22	zobrazit	4	John	Vescas
<input checked="" type="checkbox"/>	23	zobrazit	5	Joe	Rosales
<input checked="" type="checkbox"/>	24	zobrazit	5	Joe	Rosales

[ZOBRAZIT VYBRANÉ ČLÁNKY](#)

Vytvořit z článků sborník

Obr. 3.16: Vyhodnocení článků

Tabulka na této stránce je vyjíměčně vytvořena v HTML šabloně. Jednotlivé řádky tabulky se vypisují pomocí opakovače *ListView*. Pro zaškrťovací pole je použita komponenta *CheckGroup* a pro centrální zaškrťovací pole *CheckGroupSelector*.

Odkaz v tabulce pro otevření modálního okna je implementovaný v třídě *InfoModalPanel* pomocí komponenty *AjaxLink*. Další odkaz je stejným způsobem implementovaný pro zobrazení vybraných článků.

Nový sborník se vytvoří spojením všech vybraných článků. Pro tyto účely je v práci použita knihovna *iText*[15].

3.5.10 Editace sborníku

Stránka *MagazineEditPage* je určena k úpravě sborníků, které jsou zde seřazeny v tabulce. V první sloupci vlevo je číslo sborníku, následuje datum vydání, odkaz na stažení sborníku a další odkaz na očíslování sborníku. V dalším sloupci se nachází formulářové okno pro udevzdání sborníku. Šéfredaktor má tak možnost provést na vytvořeném sborníku dodatečné úpravy.

Číslo	Datum vydání	Soubor	Vložit			
1	23.5.10 16:12	23.05.2010 1.pdf	Očíslovat	<input type="text"/>	Procházet...	Smazat

Vyberte počet řádků

Obr. 3.17: Editace sborníku

V poslední sloupci je odkaz na smazání sborníku. Nesmaže se celý řádek tabulky, ale jen uložený soubor. Pokud je soubor smazaný, tak se pomocí metody *isVisible* nastaví jako neviditelné odkazy pro stažení, očíslování a smazání sborníku. Číslo zůstává a datum se aktualizuje při dalším odevzdání. Uložené datum slouží k tomu účelu, aby během jednoho recenzního řízení nevznikalo více sborníků. Číslo sborníků a datum vydání lze v tabulce třídit.

Vytvořené sborníky se vypisují pomocí opakovače *DataTable* od nejnovějšího čísla. Čtyři sloupce tabulky neobsahují prostý text a jsou proto implementovány pomocí Wicket panelů. Odkaz pro stažení sborníku ve třídě *DownloadMagazinePanel*, pro smazání pak ve třídě *DeleteMagazinePanel*. Třída *NumerateMagazinePanel* pro očíslování sborníku používá knihovnu *iText*[15]. V třídě *UploadMagazinePanel* je implementováno odevzdání sborníku. Počet zobrazených řádků v tabulce lze měnit pomocí výběrové komponenty pod tabulkou.

3.5.11 Editace článku

Šéfredaktor může na stránce *AdminArticleEditPage* editovat odevzdané články. Po kliknutí na potvrzovací tlačítko pro výběr článku dojde k přesměrování na stránku *ListArticlesPage* se seznamem odevzdaných článků.

Výběr článku

Změna kategorie

Současná kategorie článku: Matematika

Editace článku

Název: *

Obsah: *

Spoluautor:

Vložení článku

Smazání článku

Obr. 3.18: Editace článku

Ta obsahuje tabulku, kde je zleva identifikační číslo článku, odkaz na detail s názvem a anotací článku, číslo sborníku pro který byl článek odevzdán, informace jestli byl schválen k publikování, odkaz na stažení článku. Následuje identifikační číslo, jméno a příjmení autora článku. Vpravo je potvrzovací tlačítko pro vybrání článku. Textové položky tabulky lze seřadit.

Po jeho potvrzení se načtou uložená data do formulářových oken stránky *AdminArticleEditPage*. Ve výběrové nabídce lze změnit kategorii článku. Tři textová pole jsou určena pro název, anotaci a spoluautory článku. Šéfredaktor má také možnost článek v PDF souboru upravit a znovu vložit. Stránku uzavírá potvrzovací tlačítko pro smazání článku.

K implementaci stránky *AdminArticleEditPage* byly použity komponenty *TextArea* a *DropDownChoice*. Odevzdání článku je ve vnitřní třídě *UploadForm*. Tabulka ve třídě *ListArticlesPage* je vytvořena pomocí komponenty *DataTable*.

ID Č	Detail	ID S	Schváleno	Článek	ID A	Jméno	Příjmení	
19	zobrazit	1	ano	soubor-1.pdf	2	Mary	Thompson	<input type="button" value="Vybrat"/>
20	zobrazit	1	ano	soubor-2.pdf	2	Mary	Thompson	<input type="button" value="Vybrat"/>
21	zobrazit	1	ano	soubor-3.pdf	4	John	Viescas	<input type="button" value="Vybrat"/>
22	zobrazit	1	ano	soubor-4.pdf	4	John	Viescas	<input type="button" value="Vybrat"/>
23	zobrazit	1	ano	soubor-5.pdf	5	Joe	Rosales	<input type="button" value="Vybrat"/>
24	zobrazit	1	ano	soubor-6.pdf	5	Joe	Rosales	<input type="button" value="Vybrat"/>

Obr. 3.19: Výběr článku

3.6 Implementace sezení

Sezení je oblast paměti alokovaná na straně serveru, která je určená pro uchování údajů o uživateli. Na straně klienta se tyto údaje ukládají do souborů cookies, což jsou krátké textové zprávy. V internetové prohlídce je třeba mít cookies povolené.

Když uživatel poprvé přistupuje k webové aplikaci, server generuje náhodné číslo a použije ho k identifikaci sezení. Poté ho odešle zpět k prohlídce, kde se uloží do cookie. V případě Tomcatu se jmenuje *JSESSIONID*. V cookie je také uložen název serveru a cesta. Pokud cookie není smazaná a uživatel znovu přistupuje ke stránce aplikace, prohlídce zjistí, že má uloženou cookie, která je asociovaná s názvem serveru a cestou a odešle obsah cookie serveru. Server tak zjistí které sezení má použít.

V aplikaci je sezení implementováno v třídě *UserSession*, která rozšiřuje třídu Wicketu *WebSession*. Této třídě se předává požadavek na vytvoření sezení v podobě objektu typu *Request* z třídy aplikace *EditorialSystemApplication*. Třída *UserSession* obsahuje atribut typu *User*, který se nastaví předaným objektem ze stránky *LoginPage*.

Ve Wicketu se do sezení ukládá objekt stránky. Pokud uplyne platnost sezení, je smazané a zobrazí se informativní zpráva, že platnost stránky vypršela.

Page Expired

The page you requested has expired.

[Return to home page](#)

Obr. 3.20: Platnost stránky vypršela

3.7 Autorizační strategie

V aplikaci je implementovaná autorizační strategie, která zabraňuje nepovolenému přístupu na stránky.

Na jednotlivých stránkách administračního rozhraní a v sekci pro recenzní řízení jsou nastavená přístupová práva pomocí anotací. Pokud uživatel zadá přímo do internetového prohlížeče adresu stránky, která vyžaduje přihlášení do systému, zobrazí se mu stránka přístup zamítnut s odkazem na úvodní stránku.

Access Denied

You do not have access to the page you requested.

[Return to home page](#)

Obr. 3.21: Přístup na stránku zamítnut

Pokud se uživatel úspěšně přihlásí a jeho uživatelská role ho opravňuje k přístupu na požadovanou stránku, je automaticky přesměrován na tuto stránku. Na stránce s přihlášením je použita metoda *continueToOriginalDestination*, která umožňuje se vrátit k uloženému URL. Pokud by uživatel přistupoval na stránku s přihlášením přímo, zjistí se, že není uloženo URL a uživatel by byl po úspěšném přihlášení přesměrován na úvodní stránku v administračním rozhraní s nastavením účtů.

Implementace autorizační strategie je ve třídě *UserRolesAuthorize*, která implementuje rozhraní *IRoleCheckingStrategy*. Pokud nastane výše uvedená situace, tak se v metodě *hasAnyRole* nejprve získá objekt sezení ze třídy *UserSession*. V případě, že objekt je prázdný, tj. kdy uživatel není přihlášený, vrátí metoda nepravdivou hodnotu a zobrazí se stránka *AccessDeniedPage*. Při přihlášení uživatele se načtou z databáze jeho uživatelské role, které se nastaví v objektu třídy typu *User* a uloží se do sezení. Zavolá se opět metoda *hasAnyRole*, v tomto případě objekt typu *User* není prázdný a získá se z něj uživatelská role. Tato role se porovná s rolí, která je uvedena v anotaci na stránce. Pokud jsou role stejné, je uživateli na stránku povolen přístup.

3.8 Testování

K testování pomocí rámce JUnit[10] lze použít pomocnou třídu *WicketTester* pro testování Wicket aplikací. Jednotkové testy bez potřeby servletového kontejneru jsou jednodušší a rychlejší.

Následuje malá ukázka jednotkového testu, ve kterém se testuje kliknutí na odkaz na stránce *HomePage* a následné přesměrování na stránku *InformationPage*.

```
// Stránka s odkazem
public class HomePage extends WebPage
{
    public HomePage()
    {
        add(new Link("toInformationPage")
        {
            public void onClick()
            {
                setResponsePage(new InformationPage());
            }
        });
    }
}

// Odkazovaná stránka
public class InformationPage extends WebPage
{
    public InformationPage()
    {
    }
}

// Testovací kód
private WicketTester tester = new WicketTester();

public void testLinkToInformationPage()
{
    // Testuje se zobrazení stránky s odkazem
    tester.startPage(HomePage.class);
    // Simuluje se kliknutí uživatele na komponentu
    tester.clickLink("toInformationPage");
    // Testuje se zobrazení odkazované stránky
    tester.assertRenderedPage(InformationPage.class);
}
}
```

4 ZPROVOZNĚNÍ APLIKACE

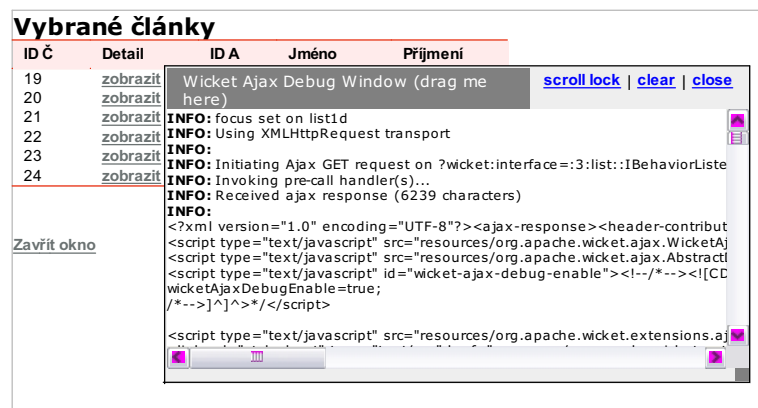
Tato kapitola se zabývá možnostmi konfigurace vytvořené aplikace, zabezpečení hesel při síťové komunikaci, způsobu logování zpráv a nasazení aplikace na server.

4.1 Konfigurace aplikace

Webová aplikace vytvořená pomocí rámce Wicket může běžet ve dvou režimech. Ve vývojovém režimu s maximální podporou pro vývojáře a v režimu nasazení pro maximální výkon a bezpečnost. Standardně se Wicket aplikace spouští ve vývojovém módu. Konzole Tomcatu po jeho spuštění vypisuje, v které režimu aplikace běží. Možností, jak lze tyto režimy přepínat, je nastavení systémové vlastnosti z příkazového řádku: `java -Dwicket.configuration=deployment`. Další možností je nastavení filtru v souboru `web.xml`, tzv. popisovači nasazení:

```
<filter>
  ... filter-name ... filter-class ...
  <init-param>
    <param-name>configuration</param-name>
    <param-value>deployment</param-value>
  </init-param>
</filter>
```

Ve vývojovém režimu jsou všechny zdroje sledovány a znovu načteny pokud u nich dojde k modifikaci. Wicket dále kontroluje, jestli jsou všechny komponenty, přidané k hierarchii Java komponent, zobrazené v HTML. Wicket ponechává všechny Wicket tagy v zobrazovaném HTML, který je odesíláný ke klientovi. Pokud nastane neočekávaná výjimka, zobrazí se stránka s kompletním výpisem běžících vláken¹.



Obr. 4.1: AJAX ladící program

¹stack trace

Stránky, které obsahují AJAX komponenty, zobrazují AJAX ladící program. Wicket ponechává komentáře a prázdná místa ze souborů JavaScriptu. Wicket nekomprimuje statické textové zdroje jako jsou soubory JavaScriptu a CSS.

V režimu nasazení je sledování modifikace zdrojů vypnuto. Wicket ignoruje komponenty přidané k hierarchii Java komponent, které nejsou v HTML. Všechny tagy Wicketu jsou odebrány ze zobrazovaného HTML, který je odeslán ke klientovi. Při neočekávané vyjímce se zobrazí chybová stránka pouze s odkazem na domovskou stránku. AJAX ladící program je vypnutý. Prázdná místa a komentáře jsou odebrány ze souborů JavaScriptu. Statické zdroje textu, např. soubory CSS a JavaScriptu jsou komprimované pomocí *gzip*, pokud je podporovaný prohlížečem klienta.

Unexpected RuntimeException

```
WicketMessage: Exception in rendering component: [Component id = start]
Root cause:
java.lang.NullPointerException
    at cz.vutbr.feec.utko.xbunka01.web.SetDeadlinePage$2.load(SetDeadlinePage.java:106)
    at org.apache.wicket.model.LoadableDetachableModel.getObject(LoadableDetachableModel.java:120)
    at org.apache.wicket.Component.getDefaultModelObject(Component.java:1657)
    at org.apache.wicket.Component.getDefaultModelObjectAsString(Component.java:1676)
    at org.apache.wicket.markup.html.basic.Label.onComponentTagBody(Label.java:114)
    at org.apache.wicket.Component.renderComponent(Component.java:2619)
    at org.apache.wicket.markup.html.WebComponent.onRender(WebComponent.java:62)
    at org.apache.wicket.Component.render(Component.java:2450)
    at org.apache.wicket.MarkupContainer.renderNext(MarkupContainer.java:1414)
    at org.apache.wicket.MarkupContainer.renderComponentTagBody(MarkupContainer.java:1577)
    at org.apache.wicket.MarkupContainer.onComponentTagBody(MarkupContainer.java:1501)
    at org.apache.wicket.Component.renderComponent(Component.java:2619)
    at org.apache.wicket.MarkupContainer.onRender(MarkupContainer.java:1512)
    at org.apache.wicket.Component.render(Component.java:2450)
    at org.apache.wicket.MarkupContainer.autoAdd(MarkupContainer.java:229)
    at org.apache.wicket.markup.resolver.MarkupInheritanceResolver.resolve(MarkupInheritanceResolver.java:66)
```

Obr. 4.2: Chyba za běhu programu

4.2 Logování zpráv

Ve vytvořené aplikaci mohou nastat neočekávané vyjímky, které jsou ošetřené bloky *try/catch*. Vyjímka může nastat např. při odesílání e-mailu, upomínky, vytvoření sborníku atd. Pro účely zaznamenání především těchto chybových zpráv do souboru je použita knihovna *Apache log4j*. Konfigurace knihovny je v souboru *log4j.properties*. Soubor pro logování byl pojmenován *log4j.log* a ukládá se do adresáře Tomcatu *bin*. Logování pak probíhá tím způsobem, že se v třídě vytvoří objekt třídy *Logger*, který si o třídě uchovává informace. Pro logování zpráv, při výskytu uvedených vyjímek v aplikaci, je použita úroveň *warn*, protože neomezují její funkčnost.

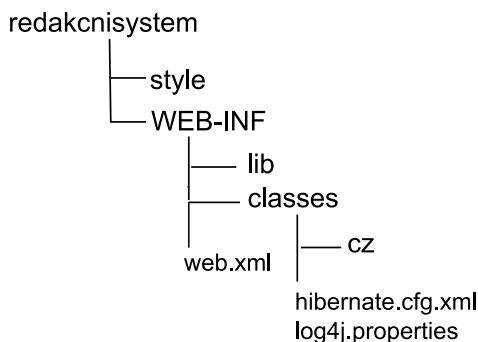
4.3 Zabezpečení komunikace

V popisu implementace již bylo uvedeno, že se hesla ukládají do databáze ve formě haše s přídatkem soli. Při síťové komunikaci mezi uživatelem a server se však přenáší data v nezašifrované podobě. Uživatelské jméno a heslo tak lze zjistit pomocí síťového analyzátoru. K bezpečné komunikaci lze použít nadstavbu HTTP protokolu HTTPS. Data přenášená sítí jsou pak šifrována. Servletový kontejner Tomcat, na které byla aplikace vyvíjena, podporuje šifrování dat pomocí protokolu SSL. Instalace a konfigurace SSL podpory pro Tomcat je uvedena na stránkách projektu[12].

4.4 Nasazení aplikace na server

Pro zprovoznění aplikace je zapotřebí servletový kontejner, aplikace byla vyvíjena na Apache Tomcat 6.0.26[7] a databáze byla použita MySQL 5.1.43[13].

Adresářová struktura aplikace se skládá z adresáře WEB-INF a style, který obsahuje soubory CSS. WEB-INF obsahuje podadresář lib s knihovnamy potřebnými pro běh aplikace, popisovač nasazení soubor *web.xml* a složku classes. V adresáři classes se nachází soubor *hibernate.cfg.xml*, *log4j.properties* a další podadresáře, které obsahují přeložené Java třídy, HTML a properties soubory.



Obr. 4.3: Adresářová struktura aplikace

Postup pro nasazení aplikace na server:

1. Vytvořit strukturu tabulek v databázi podle souboru *localhost.sql*. Kódování je nastaveno na UTF-8, porovnání UTF-8_czech_ci. Tabulky se naplní počátečními daty, které lze mimo administrátorský účet smazat.
2. V konfigurační souboru databáze *my.ini* nastavit proměnnou *set-variable = max_allowed_packet=5M*. Případně zvolit jinou hodnotu pro maximálně povolenou velikost paketu.

3. Rozbalit soubor redakcnisystem.zip a v souboru hibernate.cfg.xml opravit přihlašovací jméno a heslo pro přístup k databázi.
4. Vytvořit archiv WAR pomocí příkazu *jar -cvf redakcnisystem.war **. Příkaz znamená, že vše co je v tomto adresáři se zkomprimuje do souboru pojmenovaném redakcnisystem.war
5. Zkopírovat redakcnisystem.war do adresáře Tomcatu webapps.
6. Restartovat MySQL a Tomcat.

Aplikace se spustí zadáním adresy <http://localhost:8080/redakcnisystem/>

5 ZÁVĚR

Tato prakticky zaměřená práce se zabývala tvorbou webové aplikace na platformě Java EE pomocí rámce Apache Wicket a rámce Hibernate pro objektivě-relační mapování. Cílem bylo vytvořit aplikaci, která bude automatizovat recenzní řízení pro vědecké články.

V první části byla provedena analýza případů úžití pro jednotlivé uživatelské role pomocí diagramu UML. Analyzována byla také časová posloupnost pomocí sekvenčního diagramu.

Návrh aplikace vycházel ze způsobu implementace architektury MVC rámcem Apache Wicket. Vývoj aplikací pomocí tohoto komponentně založeného rámce se podobá programování desktopových aplikací. Dále byl proveden návrh perzistentních tříd, které se mapují na tabulky v databázi, v digramu tříd. Pro toto mapování byl použitý, dnes zřejmě nepoužívanější nástroj pro ORM, rámeček Hibernate.

Popis implementace je nejobsáhlejší částí práce. Aplikace je postupně představována a ke každé stránce je přidán popis její implementace. Nejprve se věnuje implementaci veřejné části aplikace a popisuje způsob přihlášení a registrace do systému. Následuje administrační rozhraní, kde lze měnit osobní údaje a uživatelské role. Největší část se poté věnuje tomu, jak je implementované recenzní řízení pro jednotlivé uživatelské role.

V poslední části jsou informace o tom, jak lze vytvořenou aplikaci konfigurovat mezi vývojovým režimem a režimem nasazení pro maximální výkon a bezpečnost. Na závěr je uveden postup, jakým způsobem aplikaci nainstalovat na servletový kontejner Tomcat.

LITERATURA

- [1] BAUER, C., KING, G. *Java Persistence with Hibernate*. Manning Publications Co., 2007. 841 s. ISBN 1-932394-88-5.
- [2] DASHORST, M., HILLENIOUS, E. *Wicket In Action*. Manning Publications Co., 2009. 364 s. ISBN 1-932394-98-2.
- [3] KANISOVÁ, H., MULLER, M. *UML srozumitelně*. Computer Press, 2004. 158 s. ISBN 80-251-0231-9.
- [4] PAGE-JONES, M. *Základy objektově orientovaného návrhu v UML*. Grada Publishing spol. s r.o., 2001. 367 s. ISBN 80-247-0210-X.
- [5] TONG, K., K., I. *Enjoying Web Development with Wicket*. TipTec Development, 2007. 410 s. ISBN 978-99937-929-0-1.
- [6] *Apache Lucene* [online]. Dostupné z: <http://lucene.apache.org>
- [7] *Apache Tomcat* [online]. Dostupné z: <http://tomcat.apache.org>
- [8] *Apache Wicket* [online]. Dostupné z: <http://wicket.apache.org>
- [9] *SSL Configuration* [online]. Dostupné z:
<http://tomcat.apache.org/tomcat-4.0-doc/ssl-howto.html>
- [10] *Apache log4j* [online]. Dostupné z: <http://logging.apache.org>
- [11] *Hibernate* [online]. Dostupné z: <http://www.hibernate.org>
- [12] *JUnit* [online]. Dostupné z: <http://www.junit.org>
- [13] *MySQL* [online]. Dostupné z: <http://www.mysql.com>
- [14] *Wicket-Library* [online]. Dostupné z: <http://www.wicket-library.com>
- [15] *iText PDF* [online]. Dostupné z: <http://itextpdf.com/>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

AJAX	Asynchronous JavaScript and XML
E-R diagram	Entity-Relationship diagram
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
Java EE	Java Enterprise Edition
MVC	Model-View-Controller
ORM	Objektivě-Relační Mapování
PDF	Portable Document Format
POJO	Plain Old Java Object
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
SQL	Structured Query Language
UML	Unified Modeling Language
URL	Uniform Resource Locator
WAR	Web Application Archive
XML	Extensible Markup Language