

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Nástroj pro správu požadavků



2016

Vedoucí práce: Mgr. Petr Krajča,
Ph.D.

Pavel Babák

Studijní obor: Aplikovaná informatika,
kombinovaná forma

Bibliografické údaje

Autor: Pavel Babák
Název práce: Nástroj pro správu požadavků
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2016
Studijní obor: Aplikovaná informatika, kombinovaná forma
Vedoucí práce: Mgr. Petr Krajča, Ph.D.
Počet stran: 44
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Pavel Babák
Title: Requirements management tool
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2016
Study field: Applied Computer Science, combined form
Supervisor: Mgr. Petr Krajča, Ph.D.
Page count: 44
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

V rámci bakalářské práce byl vytvořen systém pro evidenci a správu požadavků. Systém slouží pro komunikaci mezi zaměstnanci softwarové společnosti a zákazníkem při řešení zadaných požadavků na změnu softwaru. Text práce obsahuje úvod do problematiky, programátorskou dokumentaci a uživatelskou příručku.

Synopsis

The goal of this bachelor thesis was to implement an issue management system. The system tracks a list of defects and change requests and facilitates communication between the customers and the employees of a software company. The thesis contains an introduction to the topic, a development documentation and a user manual.

Klíčová slova: Nástroj pro správu požadavků

Keywords: Issue management system

Děkuji vedoucímu práce Mgr. Petrovi Krajčovi, Ph.D. za rady a připomínky. Dále bych chtěl poděkovat kolegům z práce za podporu a cenné rady při mém vstupu do světa webových technologií. Děkuji také všem blízkým lidem za podporu.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Úvod	8
2	Analýza problému	8
2.1	Řešení problému	8
2.2	Případy užití	9
2.2.1	Vytvoření požadavku	10
2.2.2	Vytvoření interního požadavku	11
2.2.3	Změna požadavku	12
2.2.4	Vytvoření uživatele	13
2.2.5	Vytvoření práva	14
2.2.6	Vytvoření záznamu číselníku	15
2.3	Životní cyklus požadavku	15
3	Programátorská dokumentace	18
3.1	Použité technologie	18
3.1.1	Hibernate	18
3.1.2	Apache Wicket	19
3.2	Popis databáze	19
3.3	Popis tříd	25
4	Uživatelská příručka	26
4.1	Instalace	26
4.2	Přístup k systému na studentském serveru	27
4.3	Základní ovládání aplikace	27
4.3.1	Přihlášení do systému	27
4.3.2	Odhlášení ze systému	27
4.3.3	Menu aplikace	28
4.3.4	Seznamy	28
4.3.5	Povinné údaje na formuláři	29
4.4	Administrace	30
4.4.1	Vytvoření uživatele	30
4.4.2	Vyhledání a editace uživatelů	30
4.4.3	Vytvoření práva	31
4.4.4	Editace práva	31
4.5	Číselníky	32
4.5.1	Vytvoření modulu	32
4.5.2	Editace modulu	32
4.5.3	Vytvoření priority	32
4.5.4	Editace priority	32
4.5.5	Vytvoření projektu	33
4.5.6	Editace projektu	33
4.5.7	Vytvoření stavu požadavku	33
4.5.8	Editace stavu požadavku	34

4.5.9	Vytvoření přechodu stavu požadavku	34
4.5.10	Editace přechodu stavu požadavku	34
4.5.11	Vytvoření typu požadavku	34
4.5.12	Editace typu požadavku	35
4.5.13	Vytvoření zákazníka	35
4.5.14	Editace zákazníka	35
4.6	Požadavky	36
4.6.1	Vytvoření požadavku	36
4.6.2	Vyhledání, editace požadavku	38
Závěr		42
A Obsah přiloženého DVD		43
Literatura		44

Seznam obrázků

1	Případ užití - Uživatel	9
2	Případy užití	10
3	Stavový diagram - životní cyklus požadavku	16
4	Databázové schéma - požadavek	20
5	Databázové schéma - požadavek a číselníky	21
6	Přihlášení do systému	28
7	Odhlášení ze systému	28
8	Platná položka v seznamu	28
9	Neplatná položka v seznamu	29
10	Upozornění na povinné údaje na formuláři	29
11	Vytvoření uživatele	30
12	Vyhledání uživatelů	31
13	Seznam práv	31
14	Editace práva	31
15	Editace modulu	32
16	Editace priority	33
17	Editace projektu	33
18	Editace stavu požadavku	34
19	Editace přechodu stavu požadavku	34
20	Editace typu požadavku	35
21	Editace zákazníka	35
22	Vytvoření požadavku	36
23	Výběr uživatelů	38
24	Vyhledání požadavků	38
25	Editace požadavku	39
26	Příspěvky požadavku	41

1 Úvod

Nástroj pro správu požadavků, který jsem v rámci této práce vytvořil, je určen pro použití v softwarové firmě, která vyvíjí a spravuje informační systém pro oblast pojišťovnictví. Zákazníkem je pojišťovna, která požaduje implementaci změn v informačním systému a přidávání nové funkcionality tak, aby informační systém odpovídal jejím aktuálním potřebám. Vytvořený nástroj by měl oběma stranám usnadnit komunikaci při řešení požadavků na změny v informačním systému a řešení požadavků na opravy dat na provozním serveru zákazníka. Na některých místech v textu se objevuje termín *Požadavkový systém*, což je jen alternativní název pro Nástroj pro správu požadavků.

Text bakalářské práce je rozdělen do tří základních kapitol. V první kapitole je popsána analýza problému a návrh jeho řešení. Součástí této kapitoly je popis hlavních případů užití a ukázka obvyklého životního cyklu požadavku.

Druhá kapitola je zaměřena na návrh databáze a popis tříd.

Obsahem třetí kapitoly je uživatelská příručka, která se skládá z pokynů pro instalaci a popisů všech formulářů v systému a jejich použití.

2 Analýza problému

Spolupráce osob podílejících se na vývoji a provozu informačního systému vyžaduje jednotný přístup k potřebným informacím. Informace o požadavcích zákazníka a jeho reklamaci je potřeba někde evidovat. Stejným způsobem je potřeba evidovat odpovědi (příspěvky) od zaměstnanců, kteří se na řešení požadavků podílejí. Je také důležité, aby se k procesu řešení konkrétního požadavku mohly kdykoliv připojit další osoby a aby bylo možné sledovat aktuální stav řešení požadavku. Pro tyto účely je zcela nevhodné, aby spolu kontaktní osoby zákazníka a zaměstnanci softwarové společnosti komunikovali prostřednictvím emailu. Osoby, které do procesu řešení požadavku vstupují až dodatečně, by tak mohly přijít o část informací a také by chyběla jednoznačná informace, v jakém stavu se požadavek aktuálně nachází.

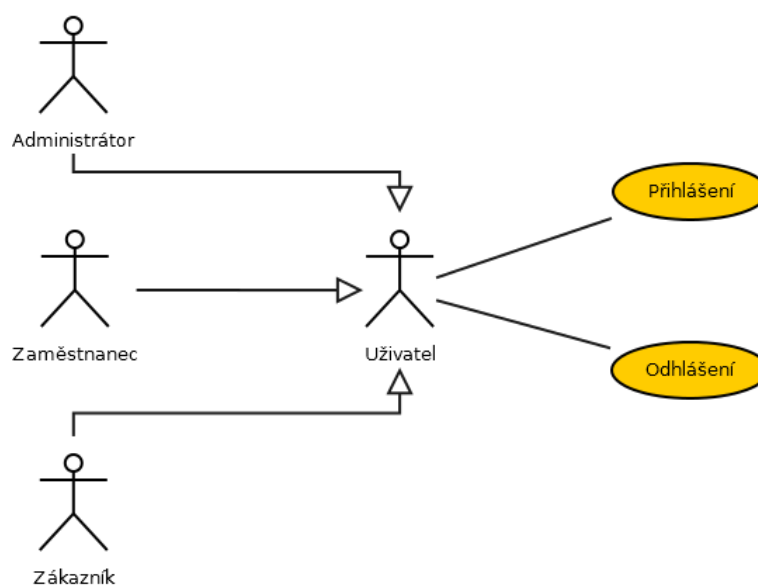
2.1 Řešení problému

Pro řešení vzniklé situace byl vytvořen jednotný informační systém. Tento systém je navržen pro komunikaci mezi softwarovou společností (dodavatelem softwarového produktu) a zákazníkem (společností, která je odběratelem softwarového produktu). Systém tedy není navržen pro komunikaci mezi jednotlivci bez začlenění ke konkrétní společnosti. Zákazníků může být v systému definováno víc, přičemž je zajištěno zamezení přístupu k požadavkům jiných zákazníků. Systém také umožňuje vytváření interních požadavků pouze pro potřeby softwarové společnosti. Komunikace mezi jednotlivými zaměstnanci se tímto způsobem začlenila do jednotné evidence informačního systému a už neprobíhá prostřednictvím emailu.

Systém o změnách na požadavku informuje uživatele prostřednictvím emailu, přičemž je přihlédnuto k tomu, že pro zákazníka jsou podstatné jen určité změny požadavku, o kterých chce být informován. Zaměstnanci jsou prostřednictvím emailu informováni o všech změnách na požadavku.

2.2 Případy užití

Uživatelé, kteří budou používat systém, se dělí do 3 základní rolí. Rozdělení uživatelů a jejich případy užití můžete vidět na Obrázcích 1 a 2.



Obrázek 1: Příklad užití - Uživatel

Zaměstnanec

Má přístup ke všem požadavkům a může modifikovat většinu jejich údajů (vyjma údajů, které jsou od vytvoření požadavků neměnné). Může k požadavku přiřazovat a odebírat ostatní zaměstnance, nemůže však přidávat a odebírat osoby patřící k zákazníkovi. Zaměstnanec může vytvářet interní požadavky, které slouží pro řešení interních úkolů. Do požadavků vytvořených zákazníkem může přidávat interní příspěvky (komentáře), které se zobrazí pouze ostatním zaměstnancům. Tato funkcionality slouží pro interní komunikaci mezi zaměstnanci při řešení požadavků zadaných zákazníkem.

Zákazník

Může vytvářet nové požadavky a v rámci požadavků může vytvářet příspěvky (komentáře). Může k požadavku přiřazovat a odebírat podílející se uživatele, kteří patří ke stejnému zákazníkovi (ke stejné společnosti) jako on sám.



Obrázek 2: Případy užití

Administrátor

Je správcem uživatelských účtů a všech číselníků. Může vytvářet nové uživatele, modifikovat všechny jejich údaje a ukončovat jejich platnost v případě, že již nemají mít oprávnění přihlásit se do systému. Může vytvářet nová práva a přiřazovat je uživatelům. Může také vytvářet a modifikovat záznamy číselníku, případně ukončit jejich platnost.

2.2.1 Vytvoření požadavku

Vytvoření požadavku je jednou z hlavních funkcí této aplikace. Vytvořením požadavku klient popisuje svůj problém, návrh na vylepšení systému, apod. Zaměstnanci mají možnost na tento požadavek reagovat. Veškerá komunikace s klientem pak probíhá právě prostřednictvím tohoto požadavku.

Aktéři: Zákazník

Popis: Vytvoření požadavku

Vstupní podmínky: Uživatel musí být přihlášený a musí mít přiřazeno právo *Záložka požadavky*.

Výstupní podmínky: Vytvoření nového požadavku.

Běžná cesta:

1. Uživatel klikne na záložky Požadavky ⇒ Vytvoření.
2. Zobrazí se formulář pro vytvoření požadavku zákazníkem.
3. Uživatel vyplní údaje, vybere podílejší se uživatele a vybere soubory, které chce přiložit k příspěvku.

4. Uživatel potvrdí vložené údaje kliknutím na tlačítko Vytvořit.
5. Systém provede kontrolu zadaných údajů.
6. Systém k požadavku přidělí odpovědného zaměstnance dle nastavení záznamu v číselníku Zákazník pro daného zákazníka.
7. Systém uloží vytvořený požadavek do databáze.
8. Systém odešle notifikační email všem uživatelům připojeným k požadavku. Součástí emailu jsou i přílohy, které uživatel vybral při vytváření.
9. Uživateli je zobrazena informace o úspěšném provedení akce.

Výjimky: V1 - Uživatel nevyplní některé povinné položky nebo vybere přílohu, která překračuje povolený limit velikosti.

1. Systém upozorní uživatele na chyby v zadaných údajích.
2. Zpět na krok 3.

2.2.2 Vytvoření interního požadavku

Vytvoření interního požadavku je důležitou součástí této aplikace. Používají ji zaměstnanci pro zadávání návrhů na změny v jádře systému. Interní požadavky se také používají na zadání práce, která nebude zákazníkovi fakturována. Může se jednat např. o opravu dat na serveru zákazníka, která byla zapříčiněna chybou v systému. Veškerá komunikace mezi zaměstnanci pak probíhá právě prostřednictvím tohoto požadavku.

Aktéři: Zaměstnanec

Popis: Vytvoření interního požadavku.

Vstupní podmínky: Uživatel musí být přihlášený a musí mít přiřazeno právo *Záložka požadavky*.

Výstupní podmínky: Vytvoření nového interního požadavku.

Běžná cesta:

1. Uživatel klikne na záložky Požadavky \Rightarrow Vytvoření.
2. Zobrazí se formulář pro vytvoření požadavku zaměstnancem.
3. Uživatel vyplní údaje, vybere podílející se uživatele a vybere soubory, které chce přiložit k příspěvku.
4. Uživatel potvrdí vložené údaje kliknutím na tlačítko Vytvořit.
5. Systém provede kontrolu zadaných údajů.
6. Systém uloží vytvořený požadavek do databáze.

7. Systém odešle notifikační email všem uživatelům připojeným k požadavku. Součástí emailu jsou i přílohy, které uživatel vybral při vytváření.
8. Uživateli je zobrazena informace o úspěšném provedení akce.

Výjimky: V1 - Uživatel nevyplní některé povinné položky nebo vybere přílohu, která překračuje povolený limit velikosti.

1. Systém upozorní uživatele na chyby v zadaných údajích.
2. Zpět na krok 3.

2.2.3 Změna požadavku

Změna požadavku je tou nejdůležitější a nejčastěji využívanou částí aplikace. Probíhá zde veškerá komunikace mezi zákazníkem a zaměstnanci až po dokončení požadavku.

Aktéři: Zákazník, Zaměstnanec

Popis: Změna požadavku

Vstupní podmínky: Uživatel musí být přihlášený a musí mít přiřazeno právo *Záložka požadavky*.

Výstupní podmínky: Změna požadavku.

Běžná cesta:

1. Uživatel klikne na záložky Požadavky ⇒ Vyhledání, editace.
2. Zobrazí se formulář pro vyhledání požadavků.
3. Uživatel zadá výběrová kritéria a potvrdí vyhledání kliknutím na tlačítko Vyhledat.
4. Zobrazí se seznam požadavků.
5. Uživatel vybere požadavek, který chce editovat a klikne na ikonu pro editaci na příslušném řádku seznamu.
6. Zobrazí se formulář pro editaci požadavku a seznam veřejných příspěvků požadavku, který na stránce v prohlížeči nahradí formulář pro vyhledání a seznam vyhledaných požadavků.
7. Uživatel modifikuje údaje, změní výběr podílejících se operátorů, vloží text příspěvku, vybere přílohy příspěvku.
8. Uživatel v seznamu příspěvků vybere příspěvek, který chce zobrazit, a klikne na ikonu pro zobrazení na příslušném řádku seznamu.
9. Zobrazí se modální okno s nezkráceným textem příspěvku a seznam příloh příspěvku.

10. Uživatel má možnost stáhnout vybrané přílohy.
11. Uživatel ukončí zobrazení modálního okna kliknutím na tlačítko Zavřít.
12. Uživatel potvrdí provedené změny na požadavku kliknutím na tlačítko Uložit.
13. Systém provede kontrolu zadaných údajů.
14. Systém uloží změny požadavku do databáze.
15. Systém provede vyhodnocení provedených změn. Pokud na požadavku byla provedena změna, o které má být informován zákazník, odešle se všem zákazníkům připojeným do požadavku notifikační email, ve kterém je soupis provedených změn. Součástí emailu jsou i nově přidané přílohy. Zaměstnanci připojení k požadavku jsou emailem informováni o jakékoliv provedené změně. Pokud v požadavku nebyly provedeny žádné změny a uživatel pouze klikl na tlačítko Uložit, pak se žádný email neodesílá.
16. Uživateli je zobrazena informace o úspěšném provedení akce.

Alternativní cesta: Přihlášený uživatel je zaměstnanec.

1. Zaměstnanci se v seznamu příspěvků zobrazí i interní příspěvky požadavku.
2. Zaměstnanec má možnost modifikovat i údaje Projekt, Modul, Stav, Pracnost, Odpovědný, Odpovědný programátor a může také vytvořit interní příspěvek.

Výjimky: V1 - Uživatel při provádění změn vymaže údaj u některé z povinných položek nebo vybere přílohu, která překračuje povolený limit velikosti.

1. Systém upozorní uživatele na chyby v zadaných údajích.
2. Zpět na krok 7.

2.2.4 Vytvoření uživatele

Vytvoření uživatele je důležité pro autorizaci osob, které k systému přistupují. Administrátor při vytváření uživatele určí jeho budoucí roli v systému pomocí práv a nastavením příznaku *Zaměstnanec*.

Aktéři: Administrátor

Popis: Vytvoření uživatele a přiřazení práv

Vstupní podmínky: Uživatel musí být přihlášený a musí mít přiřazeno právo *Záložka administrace*.

Výstupní podmínky: Vytvoření nového uživatele.

Běžná cesta:

1. Uživatel klikne na záložky Administrace ⇒ Správa uživatelů ⇒ Vytvoření.
2. Zobrazí se formulář pro vytvoření uživatele.
3. Uživatel vyplní údaje a provede přiřazení práv.
4. Uživatel potvrdí zadané údaje tlačítkem Vytvořit.
5. Systém provede kontrolu zadaných údajů.
6. Systém uloží nového uživatele do databáze.
7. Uživateli je zobrazena informace o úspěšném provedení akce.

Výjimky: V1 - Uživatel nevyplní některé povinné položky, nebo zadá login, který není v rámci systému unikátní.

1. Systém upozorní uživatele na chyby v zadaných údajích.
2. Zpět na krok 3.

2.2.5 Vytvoření práva

Práva slouží k omezení přístupu k jednotlivým částem požadavkového systému. Administrátor má možnost přidávat nová práva a tato práva mohou být následně využita, až bude do požadavkového systému přidána nová funkcionalita.

Aktéři: Administrátor

Popis: Vytvoření nového práva

Vstupní podmínky: Uživatel musí být přihlášený a musí mít přiřazeno právo *Záložka administrace*.

Výstupní podmínky: Vytvoření nového práva.

Běžná cesta:

1. Uživatel klikne na záložky Administrace ⇒ Práva ⇒ Vytvoření.
2. Zobrazí se formulář pro vytvoření práva.
3. Uživatel vyplní údaje a potvrdí tlačítkem Vytvořit.
4. Systém provede kontrolu zadaných údajů.
5. Systém uloží nové právo do databáze.
6. Uživateli je zobrazena informace o úspěšném provedení akce.

Výjimky: V1 - Uživatel nevyplní některé povinné položky, nebo zadá kód práva, který není v rámci systému unikátní.

1. Systém upozorní uživatele na chyby v zadaných údajích.
2. Zpět na krok 3.

2.2.6 Vytvoření záznamu číselníku

Číselníky slouží k definování standardních hodnot používaných v rámci systému a umožňují tak zachovat data v konzistentní a jednotné podobě. Tyto hodnoty jsou nastaveny už při instalaci systému, ale administrátor má právo je měnit či přidávat.

Aktéři: Administrátor

Popis: Vytvoření záznamu číselníku

Vstupní podmínky: Uživatel musí být přihlášený a musí mít přiřazeno právo *Záložka číselníky*.

Výstupní podmínky: Vytvoření nového záznamu číselníku.

Běžná cesta:

1. Uživatel klikne na záložku Číselníky a následně vybere záložku s číselníkem, pro který chce vytvořit nový záznam.
2. Zobrazí se formulář pro vytvoření záznamu konkrétního číselníku (např. formulář Vytvoření zákazníka).
3. Uživatel vyplní údaje a potvrdí tlačítkem Vytvořit.
4. Systém provede kontrolu zadaných údajů.
5. Systém uloží nový záznam číselníku do databáze.
6. Uživateli je zobrazena informace o úspěšném provedení akce.

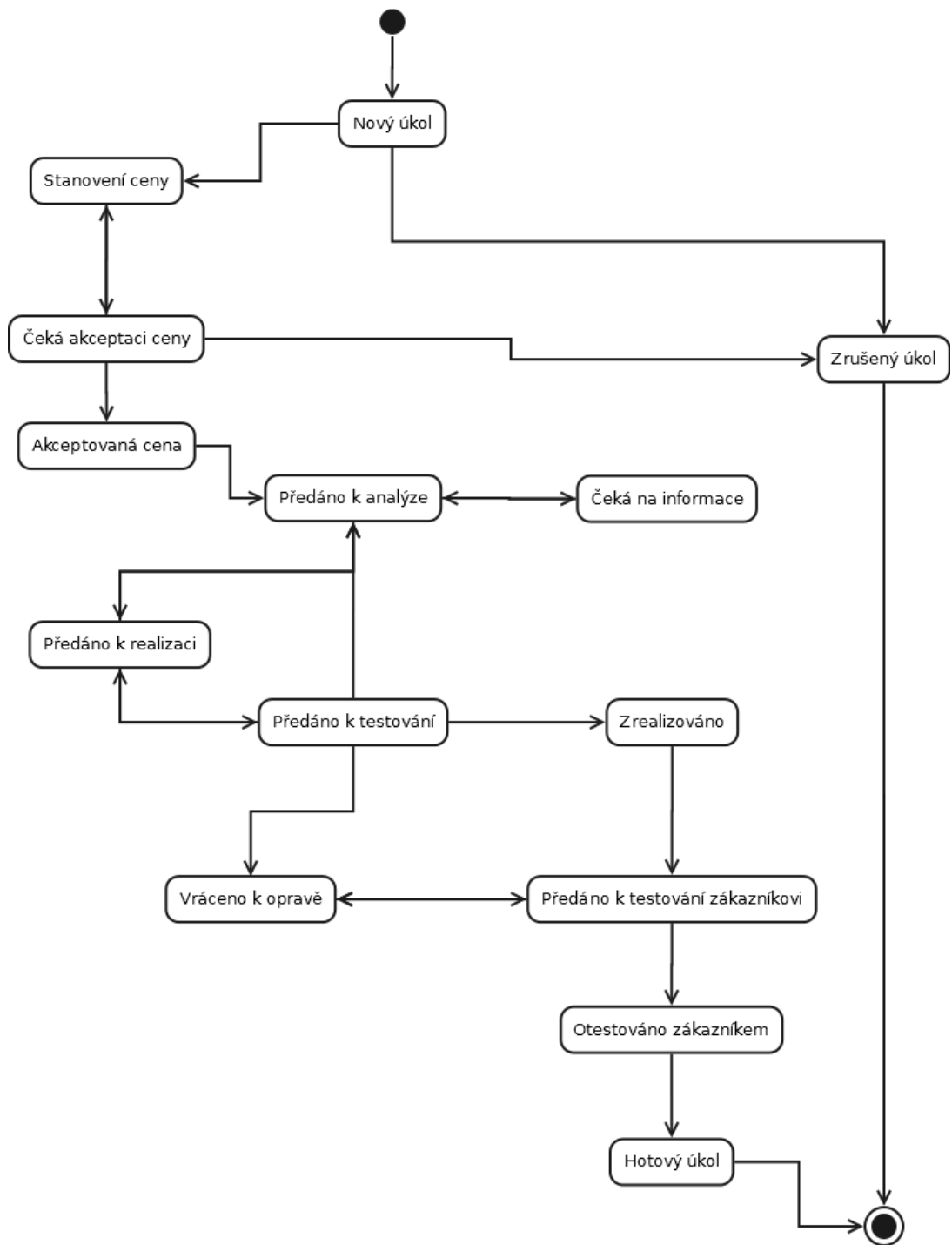
Výjimky: V1 - Uživatel nevyplní některé povinné položky, nebo zadá kód záznamu, který není v rámci daného číselníku unikátní.

1. Systém upozorní uživatele na chyby v zadaných údajích.
2. Zpět na krok 3.

2.3 Životní cyklus požadavku

Cyklus požadavku závisí vždy na zvoleném typu při vytváření. Jiný cyklus může být definován pro typ *Nové zadání* a jiný pro typ *Oprava dat*. Nové typy můžeme definovat v číselníku *Typ požadavku*, kde lze také provést přejmenování nebo zneplatnění záznamů číselníku. Totéž platí i pro definování stavů požadavků. V číselníku *Stav požadavku* můžeme přidat nové stavy a přejmenovat nebo zneplatnit stávající záznamy číselníku. Vztahy mezi jednotlivými stavy pro určitý konkrétní typ požadavku můžeme definovat v číselníku *Přechody stavů požadavků*. Pomocí těchto tří číselníků můžeme zcela změnit výchozí konfiguraci systému a přizpůsobit životní cyklus požadavku našim potřebám.

Stavový diagram pro nejčastěji se vyskytující typ požadavku *Nové zadání* můžete vidět na Obrázku 3. Pro větší přehlednost jsou v diagramu uvedeny jen důležité stavy a důležité přechody mezi stavy.



Obrázek 3: Stavový diagram - životní cyklus požadavku

Popis jednotlivých stavů:

- **Nový úkol** - Výchozí stav požadavku po zadání do systému.
- **Stanovení ceny** - V tomto stavu se určuje počet hodin potřebný k realizaci požadavku. Výsledkem je nastavení položky Pracnost a přepnutí požadavku do stavu Čeká akceptaci ceny.
- **Čeká akceptaci ceny** - Zákazník může akceptovat cenu, nebo vyjádřit pochybnosti kvůli příliš vysoké pracnosti. V takovém případě se může rozhodnout pro zrušení požadavku, nebo pro realizaci třeba jen části původního zadání. V případě, že dojde ke změně zadání, požadavek se poté vrací opět do stavu Stanovení ceny.
- **Zrušený úkol** - Do tohoto stavu se požadavek dostane, pokud se zákazník rozhodne, že požadavek nechce realizovat, ať už je důvodem příliš vysoká cena implementace, nebo chybné zadání požadavku do systému.
- **Akceptovaná cena** - Do tohoto stavu se požadavek dostane, pokud zákazník souhlasí s odhadnutou pracností.
- **Předáno k analýze** - Zadání je v tomto stavu analytiky zanalyzováno, výstupem analýzy jsou případy užití a následně je požadavek předán k realizaci programátorům.
- **Čeká na informace** - Pokud se během analýzy požadavku vyskytnou nejasnosti v zadání a je potřeba je upřesnit, přejde požadavek do tohoto stavu, kdy se čeká na další informace k zadání od zákazníka.
- **Předáno k realizaci** - V tomto stavu je požadavek zanalyzovaný a může začít jeho implementace programátory.
- **Předáno k testování** - Požadavek je implementovaný a je předán internímu testovacímu oddělení k otestování.
- **Zrealizováno** - Požadavek je otestovaný a je připraven k nasazení na testovací servery zákazníka.
- **Předáno k testování zákazníkovi** - Implementace byla nasazena na testovací server zákazníka a zákazník má možnost novou funkcionalitu vyzkoušet.
- **Vráceno k opravě** - Během testování požadavku byly odhaleny chyby, které je potřeba opravit. Může se jednat o chyby v analýze (zadání bylo špatně zanalyzováno, pochopeno) nebo v implementaci. Podle typu chyby se požadavek vrací do stavu *Předáno k analýze* nebo *Předáno k realizaci*. Pokud je oprava chyby nenáročná, může se požadavek obratem vrátit k testování.

- **Otestováno zákazníkem** - Zákazník požadavek na svém testovacím serveru otestoval a ten může být nasazen na provozní server.
- **Hotový úkol** - Požadavek je nasazený na provozní server a nejsou hlášeny žádné chyby.
- **Rozpracovaná analýza, Rozpracovaná realizace, Rozpracované testování** - Tyto stavy se v praxi příliš nevyužívají, protože přechod do stavu generuje odeslání emailu zaměstnancům a tato informace není u požadavků s nižší prioritou většinou podstatná. Používá se u požadavků s extrémní prioritou, u nichž je kladen důraz na rychlou realizaci požadavku a jsou ze strany vedoucích pracovníků sledované. Zaměstnanec přepnutím do některého z těchto stavů dává explicitně najevo, že práce na požadavku právě probíhá.
- **Čeká na informace k úkolu, Čeká na informace k zadání, Čeká na informace k realizaci, Čeká na informace k opravě** - Tyto stavy se v praxi příliš často nevyužívají.

3 Programátorská dokumentace

3.1 Použité technologie

Aplikace byla vytvořena v programovacím jazyku Java s využitím vývojového prostředí Eclipse. Pro snadné a rychlé ladění aplikace byl použit plugin RunJettyRun, který umožňuje vytvoření běhového prostředí přímo z Eclipse bez nutnosti nasazení projektu do webového kontejneru Apache Tomcat. Jako databáze byla vybrána MySQL a pro objektově relační mapování framework Hibernate. Pro řízení transakcí byl použit framework Spring. Webová aplikace byla vytvořena ve frameworku Apache Wicket.

3.1.1 Hibernate

Hibernate slouží pro mapování Javovských objektů na entity v relační databázi. Pro mapování byly použity anotace u domainových tříd. Každá domainová třída obsahuje anotaci samotné třídy, která slouží pro vygenerování tabulky do příslušného databázového schématu a dále anotace všech get metod pro vygenerování sloupců v tabulce. Pokud některá z metod není nebo nemá být provázána se sloupcem v databázi, musí být označena speciální anotací @Transient. To, jestli bude Hibernate modifikovat databázové schéma, určuje nastavení parametru `hbm2ddl.auto` v konfiguračním souboru `hibernate.cfg.xml`. Nejčastěji používané možnosti nastavení tohoto parametru jsou následující:

- **validate** - Validuje databázové schéma, ale neprovádí žádné jeho změny.

- **update** - Provede update databázového schématu. Vytvoří chybějící tabulky k domainovým třídám. U existujících tabulek provede přidání nebo odebrání sloupců dle provedených změn v odpovídající domainové třídě.

Hibernate také umožňuje snadné vyhledání dat v databázi za použití vyhledávacích kritérií (třída Criteria) nebo za použití HQL (The Hibernate Query Language) či nativního SQL pro složitější dotazy. [1]

3.1.2 Apache Wicket

Apache Wicket je framework pro tvorbu webových aplikací v programovacím jazyku Java. Jedná se o komponentně řízený framework (Component-based framework), kde se pracuje s připravenými nebo vlastními komponentami. Při vytváření vlastních komponent je možné využít dědičnost. S pomocí těchto komponent se vytváří webové stránky podobně, jako se vytváří grafická rozhraní standardní desktopové aplikace. Jednotlivé části webové stránky jsou pak definovány ve třech stejně pojmenovaných souborech, ovšem s odlišnými příponami java, html a xml. V souboru java je umístěn programový kód pro vytvoření komponent a obsluhu událostí na stránce. V souboru html je definován vzhled stránky a rozmístění komponent. V souboru xml jsou umístěny lokalizace textů a hlášení do daného jazyka. Souborů xml může být tedy definováno i víc, pokud provádíme lokalizaci do více jazyků.

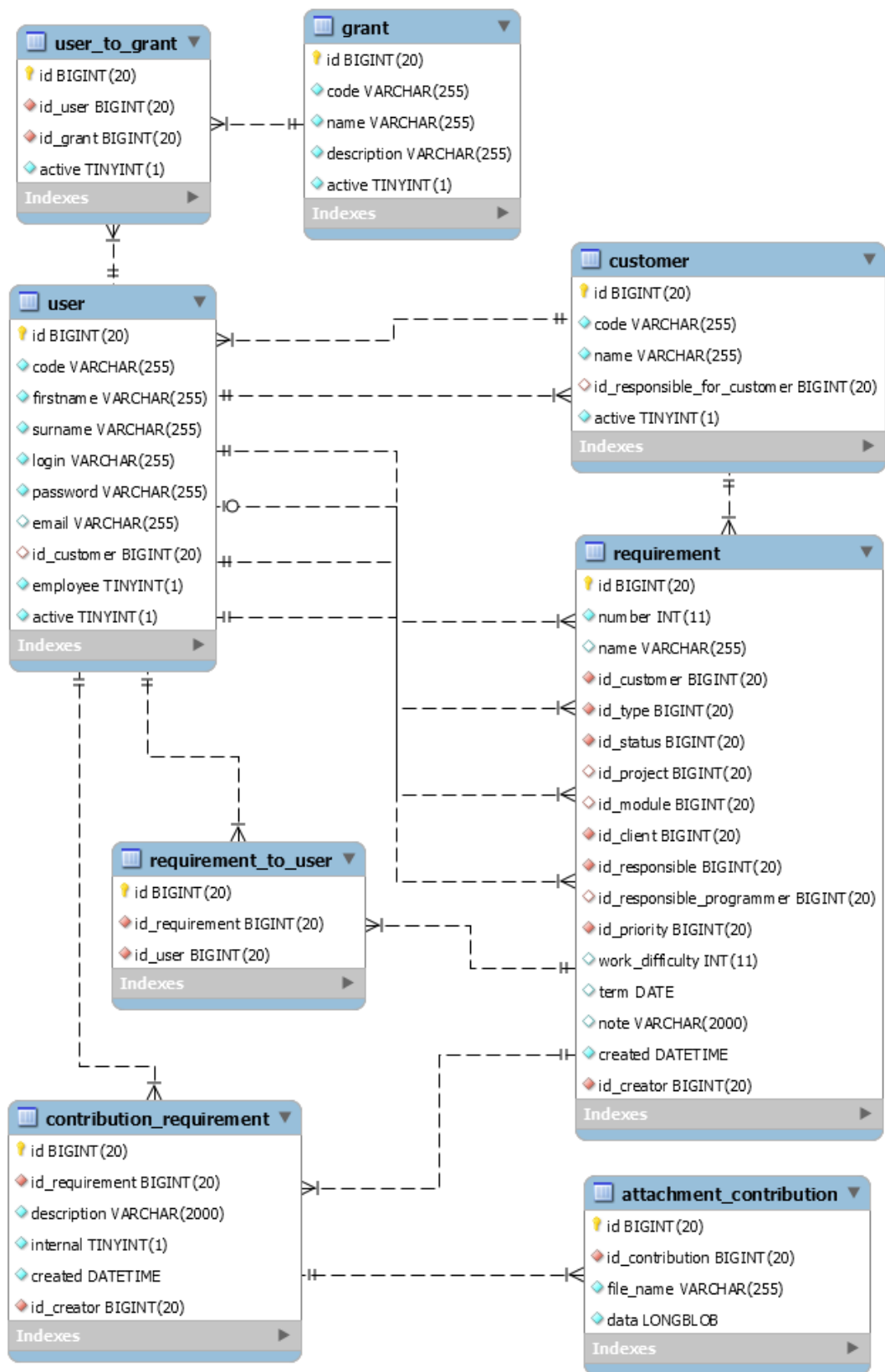
Wicket je server-side framework, tj. zdrojový kód je prováděn na straně serveru a prohlížeč dostane stránku k zobrazení.[2]

3.2 Popis databáze

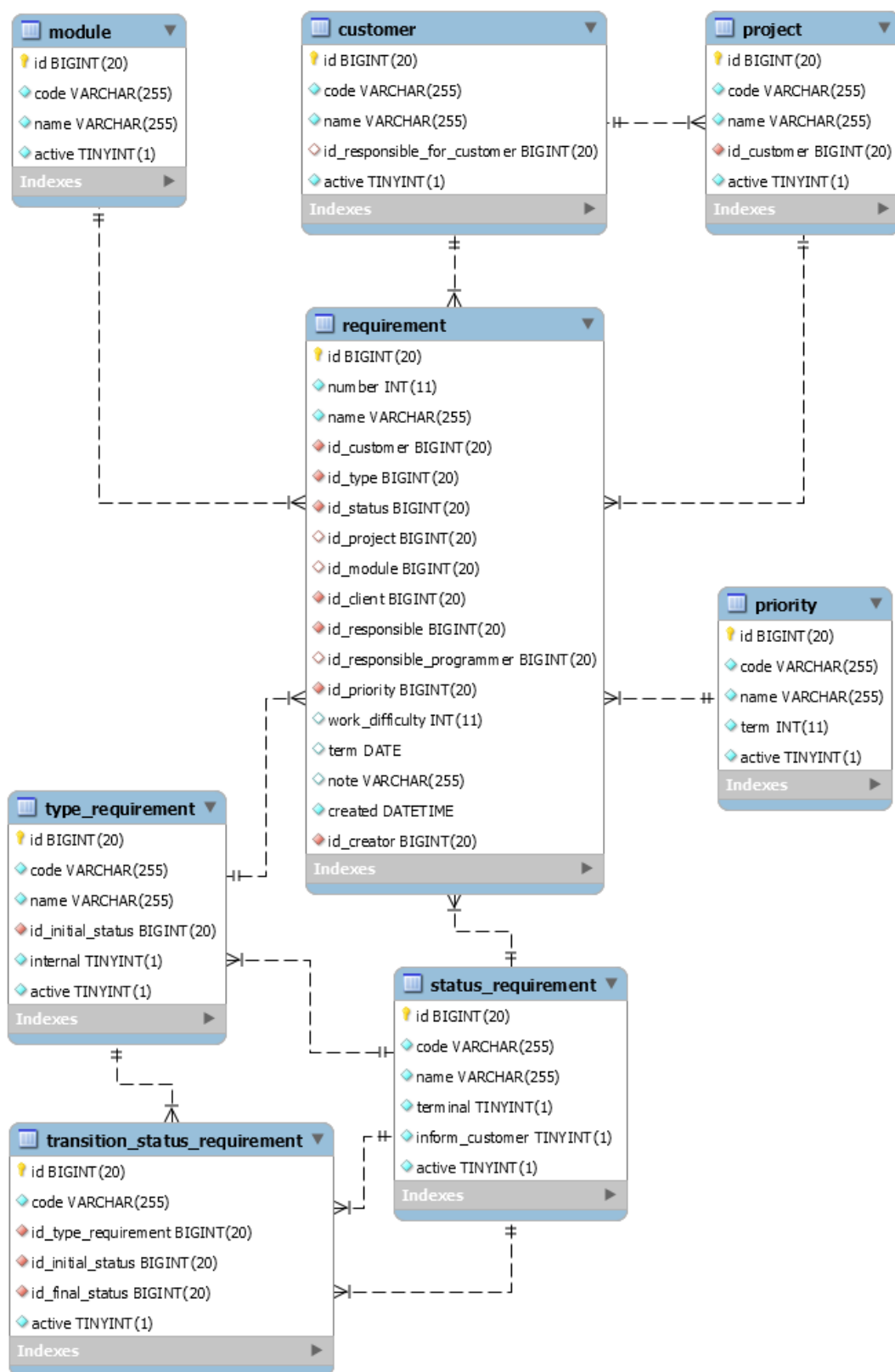
Tato kapitola popisuje strukturu databáze, její tabulky a všechny sloupce. Databázové schéma můžete vidět na Obrázku 4 a také na Obrázku 5, kde jsou všechny číselníky.

Tabulka **grant** obsahuje práva, která jsou použita pro omezení přístupu k určitým částem systému. Struktura tabulky vypadá následovně:

Atribut	Popis
id	primární klíč tabulky
code	identifikace práva (musí být unikátní)
name	název práva
description	popis práva
active	příznak určuje, zda se jedná o platnou položku



Obrázek 4: Databázové schéma - požadavek



Obrázek 5: Databázové schéma - požadavek a číselníky

Tabulka **user** obsahuje základní informace pro identifikaci uživatele a příznaky, v jaké roli uživatel může s aplikací pracovat. Struktura tabulky vypadá následovně:

Atribut	Popis
id	primární klíč tabulky
code	identifikace uživatele (generovaný údaj určený pro odlišení uživatelů se stejnými jmény)
firstname	křestní jméno
surname	příjmení
login	přihlašovací login uživatele
password	heslo uživatele
email	email uživatele, na který mu budou odesílány zprávy při změnách v požadavku
id_customer	zákazník, ke kterému je uživatel přiřazen
employee	příznak určuje, zda je uživatel zaměstnanec nebo zákazník
active	příznak určuje, zda se jedná o platnou položku

Tabulka **user_to_grant** je vazební tabulkou mezi uživatelem a právy. Struktura tabulky vypadá následovně:

Atribut	Popis
id	primární klíč tabulky
id_user	cizí klíč, odkaz do tabulky uživatelů
id_grant	cizí klíč, odkaz do tabulky práv
active	příznak určuje, zda se jedná o platnou položku

Tabulka **customer** obsahuje informace o zákaznících. Struktura tabulky vypadá následovně:

Atribut	Popis
id	primární klíč tabulky
code	identifikace zákazníka
name	jméno zákazníka
id_responsible_for_customer	uživatel, který je odpovědnou osobou pro daného zákazníka a jsou mu implicitně přidělovány nové požadavky vytvořené zákazníkem
active	příznak určuje, zda se jedná o platnou položku

Tabulka **project** obsahuje informace o projektech pro jednotlivé zákazníky. Struktura tabulky vypadá následovně:

Atribut	Popis
id	primární klíč tabulky
code	identifikace projektu
name	název projektu
id_customer	zákazník, ke kterému daný projekt náleží
active	příznak určuje, zda se jedná o platnou položku

Tabulka **module** obsahuje informace o modulech systému. Struktura tabulky vypadá následovně:

Atribut	Popis
id	primární klíč tabulky
code	identifikace modulu
name	název modulu
active	příznak určuje, zda se jedná o platnou položku

Tabulka **priority** obsahuje informace o prioritách požadavku. Struktura tabulky vypadá následovně:

Atribut	Popis
id	primární klíč tabulky
code	identifikace priority
name	název priority
term	počet dnů od aktuálního data, dle kterého bude přednastaven termín požadavku
active	příznak určuje, zda se jedná o platnou položku

Tabulka **status_requirement** obsahuje informace o stavech požadavku. Struktura tabulky vypadá následovně:

Atribut	Popis
id	primární klíč tabulky
code	identifikace stavu
name	název stavu
terminal	příznak určuje, zda se jedná o koncový stav požadavku, z něhož požadavek už nemůže přejít do dalšího stavu
inform_customer	příznak určuje, zda při přechodu požadavku do tohoto stavu má být informován zákazník, respektive uživatelé zákazníka, kteří jsou přiřazeni k danému požadavku
active	příznak určuje, zda se jedná o platnou položku

Tabulka **transition_status_requirement** obsahuje informace o přechodech stavu požadavku. Lze v ní definovat přechody z výchozího do koncového stavu pro konkrétní typ požadavku. Struktura tabulky vypadá následovně:

Atribut	Popis
id	primární klíč tabulky
code	identifikace přechodu stavu požadavku
id_type_requirement	typ požadavku, pro který je přechod definován
id_initial_status	výchozí stav požadavku
id_final_status	koncový stav požadavku
active	příznak určuje, zda se jedná o platnou položku

Tabulka **requirement** obsahuje informace o požadavcích. Struktura tabulky vypadá následovně:

Atribut	Popis
id	primární klíč tabulky
number	číslo požadavku. Kombinace zákazník a číslo požadavku musí být v rámci systému unikátní. Číslo požadavku je přidělováno automaticky na základě maximálního čísla požadavku pro daného zákazníka, které je zvýšeno o 1.
name	název požadavku
id_cusomer	zákazník
id_type	typ požadavku
id_status	stav požadavku
id_project	projekt
id_module	modul
id_client	uživatel, který je zadavatelem požadavku
id_responsible	uživatel (zaměstnanec), který je odpovědný za vyřešení požadavku
id_responsible_programmer	uživatel (zaměstnanec), který je odpovědný za realizaci požadavku
id_priority	priorita požadavku
work_difficulty	odhadovaný počet hodin, který je potřebný pro realizaci požadavku
term	termín, ve kterém zákazník požaduje vyřešení požadavku
note	poznámka
created	datum a čas, kdy byl požadavek vytvořen
id_creator	uživatel, který požadavek vytvořil

Tabulka **requirement_to_user** je vazební tabulkou mezi požadavkem a uživateli. Struktura tabulky vypadá následovně:

Atribut	Popis
id	primární klíč tabulky
id_requirement	cizí klíč, odkaz do tabulky požadavků
id_user	cizí klíč, odkaz do tabulky uživatelů

Tabulka **contribution_requirement** obsahuje texty příspěvků od uživatelů a další informace. Struktura tabulky vypadá následovně:

Atribut	Popis
id	primární klíč tabulky
id_requirement	cizí klíč, odkaz do tabulky požadavků
description	popis, samotný text příspěvku
internal	příznak označující, zda se jedná o interní příspěvek, který mohou vidět pouze zaměstnanci
created	datum a čas, kdy byl příspěvek vytvořen
id_creator	uživatel, který příspěvek vytvořil

Tabulka **attachment_contribution** je určena pro ukládání příloh (souborů) příspěvku. Data souboru jsou v tabulce uložena v atributu, který má datový typ BLOB (Binary Large Object). Struktura tabulky vypadá následovně:

Atribut	Popis
id	primární klíč tabulky
id_contribution	cizí klíč, odkaz do tabulky příspěvků
file_name	jméno uloženého souboru
data	data uloženého souboru

3.3 Popis tříd

Základní rozdělení tříd v projektu podle vrstev:

1. **Domain** - Domainové třídy obsahují anotace a jsou namapovány na databázové tabulky. Každá domainová třída tedy odpovídá některé z tabulek v databázi. Domainové třídy jsou umístěny v těchto packages:

```
cz.app.bl.administration.domain
```

```
cz.app.bl.codebook.domain
```

```
cz.app.bl.requirement.domain
```

2. **DAO (Data Access Object)** - Abstrahuje a zapouzdřuje veškerý přístup ke zdrojům dat. Spravuje spojení se zdrojem dat (s databází), aby bylo možné získávat a ukládat data. Třídy představující DAO vrstvu jsou umístěny v těchto packages:
`cz.app.bl.administration.dao`
`cz.app.bl.codebook.dao`
`cz.app.bl.requirement.dao`.
3. **Service** - Je výkonnou částí business logiky. Je to vrstva, která poskytuje jednotlivé služby pro volání z prezentační (webové) vrstvy. Pro práci s databází využívá metody z DAO vrstvy. Třídy servisní vrstvy jsou umístěny v těchto packages:
`cz.app.bl.administration.service`
`cz.app.bl.codebook.service`
`cz.app.bl.requirement.service`
4. **Prezentační (webová) vrstva** - Zajišťuje služby týkající se uživatelského rozhraní a ovládání aplikace. Třídy představující prezentační vrstvu jsou v projektu umístěny v package `cz.webapp`

4 Uživatelská příručka

4.1 Instalace

Při instalaci systému je nutné provést tyto kroky.

1. **Instalace Javy 1.7** - Instalace JDK 7 od Oracle.
2. **Instalace databáze MySQL** - Instalace MySQL 5.7.12 (nejlépe včetně programu MySQL WorkBench)
3. **Import databáze** - Import databáze prostřednictvím programu MySQL WorkBench nebo z příkazové řádky. Na přiloženém DVD je připraven dump databáze, který naimportujeme. Při importu bude vytvořeno schéma `requirements` včetně struktury tabulek a nalití dat. Pokud import provádíme z příkazové řádky, zadáme:

```
mysql -u root --p < DumpRequirements.sql
```

V operačním systému MS Windows musí být v systémové proměnné Path nastavena cesta k podadresáři `bin` MySQL Serveru, nebo příkaz musíme spustit z adresáře, ve kterém se nachází soubor `mysql.exe`. Obvyklé umístění adresáře je `C:\Program Files\MySQL\MySQL Server 5.7\bin`.

4. **Instalace Apache Tomcat 8** - V operačním systému MS Windows můžeme použít buď instalátor, nebo rozbalíme archiv zip, ve kterém se nachází Tomcat. V proměnném prostředí vytvoříme proměnnou `CATALINA_HOME` a jako její hodnotu nastavíme cestu k adresáři s Tomcat8.

5. **Spuštění Apache Tomcat** - Tomcat spustíme spuštěním dávkového souboru `startup.bat` nebo skriptu `startup.sh`, který se nachází v podadresáři `bin` adresáře Tomcatu.
6. **Nastavení konfiguračních souborů a úprava war souboru** - Na přiloženém DVD je v adresáři **webapp** soubor `SpravaPozadavku.war` a dále tyto konfigurační soubory:
 - `dao-context.xml`, `hibernate.cfg.xml`, `transaction-spring-config.xml`
V souborech nastavíme údaje pro připojení k databázi MySQL tak, aby odpovídalo naší instalaci. Upravíme údaje `url`, `username` a `password`. Nesmíme ale měnit název databázového schématu `requirements`.
 - `mail.properties` - V souboru jsou definovány údaje pro připojení k smtp serveru, ze kterého budou odesílány emaily uživatelům. Tento soubor není nutné měnit a je možné využít stávajícího nastavení.

Po provedení úprav zkopírujeme konfigurační soubory do adresáře `WEB-INF/classes`, který je zabalený uvnitř souboru `SpravaPozadavku.war` a přepíšeme tím původní konfigurační soubory.

7. **Nasazení balíku war na Apache Tomcat** - V prohlížeči se přihlásíme k Tomcat Web Application Manager a provedeme `deploy` upraveného balíku `war`.
8. **Přihlášení k systému**

4.2 Přístup k systému na studentském serveru

Aplikace je přichystaná k vyzkoušení na serveru `student.inf.upol.cz` na adrese <http://158.194.92.14:8081/SpravaPozadavku/login>. Postup přihlášení je stejný, jako když aplikaci sami instalujeme.

4.3 Základní ovládání aplikace

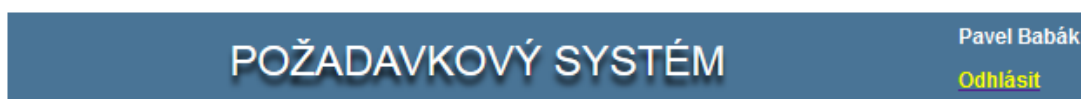
4.3.1 Přihlášení do systému

Se systémem mohou pracovat pouze přihlášení uživatelé. Pro přihlášení do systému slouží úvodní stránka, kde je okno pro přihlášení. Přihlašovací dialog můžete vidět na Obrázku 6. Pro vyzkoušení systému je přichystán uživatel s loginem **admin** a heslem **admin**, který má přiřazena veškerá práva.

4.3.2 Odhlášení ze systému

Pro odhlášení ze systému slouží hypertextový odkaz **Odhlásit**, který je umístěn na horní liště vpravo pod jménem přihlášeného uživatele. Hlavní lištu aplikace můžete vidět na Obrázku 7.

Obrázek 6: Přihlášení do systému



Obrázek 7: Odhlášení ze systému

4.3.3 Menu aplikace

Hlavní menu je tvořeno záložkami, které mají hierarchickou strukturu. Po výběru záložky se zobrazí buď podřízené záložky, nebo stránka pro vykonání určité činnosti, např. Vytvoření uživatele. V menu je možné se kdykoliv vrátit do nadřazené úrovně kliknutím myši na příslušnou záložku. Uživateli se zobrazí pouze záložky, pro která má přiřazena příslušná práva. Hlavní menu aplikace obsahuje 3 kořenové záložky:

Požadavky - Pro zobrazení záložky musí mít uživatel přiřazeno právo *Záložka požadavky*.

Administrace - Pro zobrazení záložky musí mít uživatel přiřazeno právo *Záložka administrace*.

Číselníky - Pro zobrazení záložky musí mít uživatel přiřazeno právo *Záložka číselníky*.



4.3.4 Seznamy

V seznamech jsou od sebe odlišeny platné a neplatné položky ikonou. Platnou položku můžete vidět na Obrázku 8 a neplatnou položku na Obrázku 9.

Vyhledání uživatelé							
	Kód	Jméno	Přímení	Login	Email	Zákazník	Zaměstnanec
	4	Pavel	Babák	pbabak	pavel.babak@ais.cz	AIS Software	Ano

Obrázek 8: Platná položka v seznamu

Formulář pro **editaci položky** se zobrazí při kliknutí na ikonu s tužkou na začátku řádku.

Vyhledání uživatelé							
	Kód	Jméno	Příjmení	Login	Email	Zákazník	Zaměstnanec
 	28	Lukáš	Kameník	Ikamenik	lukas.kamenik@ais.cz	AIS Software	Ano

Obrázek 9: Neplatná položka v seznamu

V seznamech lze využít **třídění**. Řádky v seznamu setřídíme vzestupně kliknutím na příslušný název sloupce, podle kterého chceme třídít. Sestupné třídění zapneme při opětovném kliknutí na název sloupce. Třídění úplně zrušíme při dalším kliknutí na název sloupce. Seznamy neumožňují třídění dle více sloupců současně. Zapnutím třídění na jiném sloupci zrušíme předchozí volbu třídění.

4.3.5 Povinné údaje na formuláři

Při zadávání údajů na formuláři je uživatel upozorněn na nevyplněné povinné údaje oranžovým podbarvením editačních polí a také zobrazením seznamu chyb v části formuláře *Zprávy o provedené akci*. Upozornění na nevyplněné povinné údaje můžete vidět na Obrázku 10.

Zprávy o provedené akci

- Chyba: pole 'Jméno' je povinné.
- Chyba: pole 'Příjmení' je povinné.
- Chyba: pole 'Login' je povinné.
- Chyba: pole 'Heslo' je povinné.
- Chyba: pole 'Zákazník' je povinné.
- Chyba: pole 'Práva přiřazená' je povinné.

Nový uživatel

Jméno Příjmení Kód

Login Heslo Email

Zákazník Zaměstnanec Aktivní

Práva dostupná

- Záložka administrace
- Záložka číselníky
- Záložka požadavky

Práva přiřazená

Obrázek 10: Upozornění na povinné údaje na formuláři

4.4 Administrace

4.4.1 Vytvoření uživatele

Formulář pro vytvoření uživatele je přístupný po kliknutí na záložky Administrace ⇒ Správa uživatelů ⇒ Vytvoření. Formulář můžete vidět na Obrázku 11.

Nový uživatel musí mít zadaný login, který je v rámci systému jedinečný.

Pro příjem zpráv o změnách na požadavcích je vhodné vyplnit email uživatele. Údaj není nutné vyplňovat v případě, že uživatel nebude pracovat s požadavky.

Uživatel musí mít vyplněný údaj o zákazníkovi, který identifikuje, ke které společnosti patří. Tento údaj se pro jednotnost vyplňuje i pro zaměstnance softwarové firmy.

Příznak zaměstnanec slouží k odlišení zaměstnanců a zákazníků. Zaměstnanci mohou při práci s požadavky modifikovat většinu údajů na rozdíl od zákazníků, jejichž činnost na požadavcích je značně omezena.

Uživateli musí být přiřazeno alespoň jedno z dostupných práv. Pro přiřazení práv nebo zrušení přiřazení slouží tlačítka s šipkami mezi seznamem dostupných a přiřazených práv.

Nový uživatel					
Jméno	<input type="text"/>	Příjmení	<input type="text"/>	Kód	<input type="text"/>
Login	<input type="text"/>	Heslo	<input type="text"/>	Email	<input type="text"/>
Zákazník	Vyberte hodnotu	Zaměstnanec	<input type="checkbox"/>	Aktivní	<input checked="" type="checkbox"/>
Práva dostupná		Práva přiřazená			
<ul style="list-style-type: none">Záložka administraceZáložka číselníkyZáložka požadavky		<input type="text"/>			
		-> <-			
<input type="button" value="Vytvořit"/>					

Obrázek 11: Vytvoření uživatele

4.4.2 Vyhledání a editace uživatelů

Formulář pro vyhledání uživatelů je přístupný po kliknutí na záložky Administrace ⇒ Správa uživatelů ⇒ Vyhledání, editace. Formulář můžete vidět na Obrázku 12.

Uživatel zadá výběrová kritéria, podle kterých chce uživatele vyhledat, a výběr potvrdí tlačítkem Vyhledat. Systém zobrazí nalezené uživatele v seznamu. Po kliknutí na ikonu pro editaci položky v seznamu se zobrazí formulář pro editaci uživatele. Pro zadávání položek na formuláři platí stejná pravidla jako na

Uživatel (vyhledávací kritéria)

Kód Jméno Příjmení Pouze aktivní

Login Email Zákazník Vyberte hodnotu Zaměstnanec

Vyhledat

Obrázek 12: Vyhledání uživatelů

formuláři Vytvoření uživatele, je zde ale možné provést zneplatnění nebo zrušení zneplatnění uživatele změnou hodnoty položky *Aktivní*.

4.4.3 Vytvoření práva

Formulář pro vytvoření nového práva je přístupný po kliknutí na záložky Administrace ⇒ Práva.

Zadaný kód i název práva musí být v rámci systému jedinečný.

Vytvářet nová práva má však smysl pouze v součinnosti se změnami v implementaci, např. vytvořené právo bude v budoucnu použito pro omezení přístupu uživatelů k nově přidané funkcionalitě systému.

4.4.4 Editace práva

Formulář pro editaci práva je přístupný po kliknutí na záložky Administrace ⇒ Práva a kliknutí na ikonu pro editaci vybraného řádku v seznamu práv. Seznam práv můžete vidět na Obrázku 13 a formulář pro editaci práva na Obrázku 14.

Práva		
Kód	Název	Popis
bookmark_administration	Záložka administrace	Určeno pro omezení přístupu k administraci systému.
bookmark_codebook	Záložka číselníky	Určeno pro omezení přístupu k číselníkům systému.
bookmark_requirements	Záložka požadavky	Určeno pro omezení přístupu k požadavkům.

Obrázek 13: Seznam práv

Editace práva je omezena pouze na změnu názvu a změnu platnosti práva, kterou lze provést nastavením hodnoty položky *Aktivní*.

Zadaný název práva musí být v rámci systému jedinečný.

Editace práva

Kód Název Aktivní

Popis

Uložit

Obrázek 14: Editace práva

4.5 Číselníky

4.5.1 Vytvoření modulu

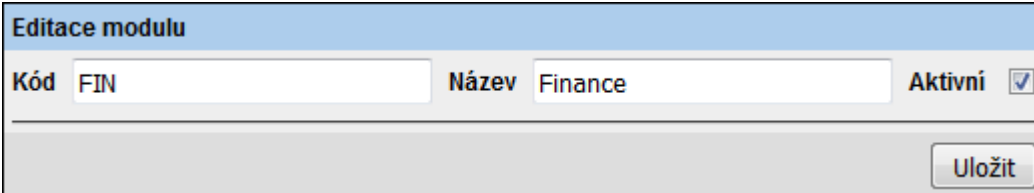
Číselník slouží pro rozdělení systému do dílčích částí (modulů). Při vytváření požadavku pak výběrem modulu můžeme identifikovat, které části systému se bude požadavek převážně týkat.

Formulář pro vytvoření nového modulu je přístupný po kliknutí na záložky Číselníky ⇒ Modul. Zadaný kód i název modulu musí být unikátní.

4.5.2 Editace modulu

Formulář pro editaci modulu je přístupný po kliknutí na záložky Číselníky ⇒ Modul a kliknutím na ikonu pro editaci vybraného řádku v seznamu modulů. Formulář můžete vidět na Obrázku 15.

Zadaný kód i název modulu musí být unikátní. Položka *Aktivní* určuje, zda se jedná o platný nebo neplatný záznam číselníku.



Editace modulu					
Kód	FIN	Název	Finance	Aktivní	<input checked="" type="checkbox"/>
					Uložit

Obrázek 15: Editace modulu

4.5.3 Vytvoření priority

Číselník slouží pro definování priorit požadavku.

Formulář pro vytvoření nové priority je přístupný po kliknutí na záložky Číselníky ⇒ Priorita. Zadaný kód priority musí být unikátní. Do položky *Termín* se zadává počet dnů pro splnění požadavku od jeho vytvoření a slouží pro automatické přednastavení termínu na požadavku.

4.5.4 Editace priority

Formulář pro editaci priority je přístupný po kliknutí na záložky Číselníky ⇒ Priorita a kliknutím na ikonu pro editaci vybraného řádku v seznamu priorit. Formulář můžete vidět na Obrázku 16.

Zadaný kód priority musí být unikátní. Položka *Aktivní* určuje, zda se jedná o platný nebo neplatný záznam číselníku.

Editace priority			
Kód	2	Název	Střední
		Aktivní	<input checked="" type="checkbox"/>
Termin	30		
			Uložit

Obrázek 16: Editace priority

4.5.5 Vytvoření projektu

Číselník slouží pro definování projektů. Projekty jsou definovány zvlášť pro každého zákazníka. V budoucnu by projekty měly sloužit i pro rozlišení typu fakturace zákazníkovi.

Formulář pro vytvoření nového projektu je přístupný po kliknutí na záložky Číselníky ⇒ Projekt. Zadaný kód projektu musí být unikátní. Je nutné vybrat zákazníka, pro kterého je projekt definován.

4.5.6 Editace projektu

Formulář pro editaci projektu je přístupný po kliknutí na záložky Číselníky ⇒ Projekt a kliknutím na ikonu pro editaci vybraného řádku v seznamu projektů. Formulář můžete vidět na Obrázku 17.

Zadaný kód projektu musí být unikátní. Položka *Aktivní* určuje, zda se jedná o platný nebo neplatný záznam číselníku.

Editace projektu			
Kód	1	Název	Sirael jádro
		Aktivní	<input checked="" type="checkbox"/>
Zákazník	AIS Software		
			Uložit

Obrázek 17: Editace projektu

4.5.7 Vytvoření stavu požadavku

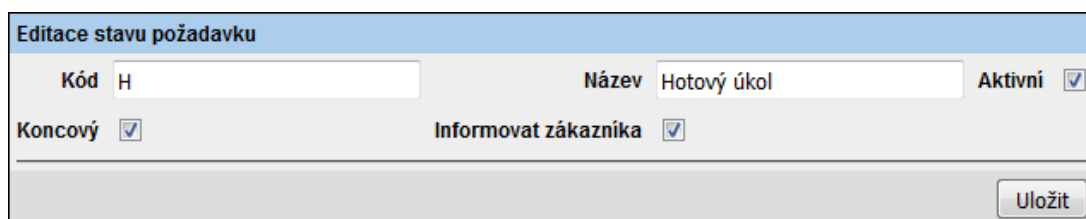
Číselník slouží pro definování stavů požadavků.

Formulář pro vytvoření nového stavu je přístupný po kliknutí na záložky Číselníky ⇒ Stav požadavku. Zadaný kód stavu musí být unikátní. Příznak *Koncový* určuje, zda se jedná o koncový stav požadavku. Požadavek je v tomto stavu ukončený. Příznak *Informovat zákazníka* určuje, zda se má být zákazník informován zasláním emailu při přechodu požadavku do tohoto stavu.

4.5.8 Editace stavu požadavku

Formulář pro editaci stavu požadavku je přístupný po kliknutí na záložky Číselníky ⇒ Stav požadavku a kliknutím na ikonu pro editaci vybraného řádku v seznamu stavů. Formulář můžete vidět na Obrázku 18.

Zadaný kód stavu musí být unikátní. Položka *Aktivní* určuje, zda se jedná o platný nebo neplatný záznam číselníku.



Obrázek 18: Editace stavu požadavku

4.5.9 Vytvoření přechodu stavu požadavku

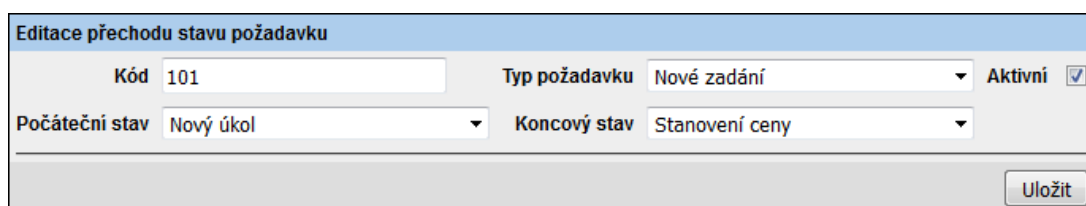
Číselník slouží pro definování přechodů mezi stavy požadavku. Pro každý typ požadavku jsou zvlášť definovány přechody mezi stavy.

Formulář pro vytvoření přechodu je přístupný po kliknutí na záložky Číselníky ⇒ Přechody stavů požadavků. Zadaný kód přechodu musí být unikátní.

4.5.10 Editace přechodu stavu požadavku

Formulář pro editaci stavu požadavku je přístupný po kliknutí na záložky Číselníky ⇒ Přechody stavů požadavků a kliknutím na ikonu pro editaci vybraného řádku v seznamu přechodů. Formulář můžete vidět na Obrázku 19.

Zadaný kód přechodu musí být unikátní. Položka *Aktivní* určuje, zda se jedná o platný nebo neplatný záznam číselníku.



Obrázek 19: Editace přechodu stavu požadavku

4.5.11 Vytvoření typu požadavku

Číselník slouží pro definování typů požadavků.

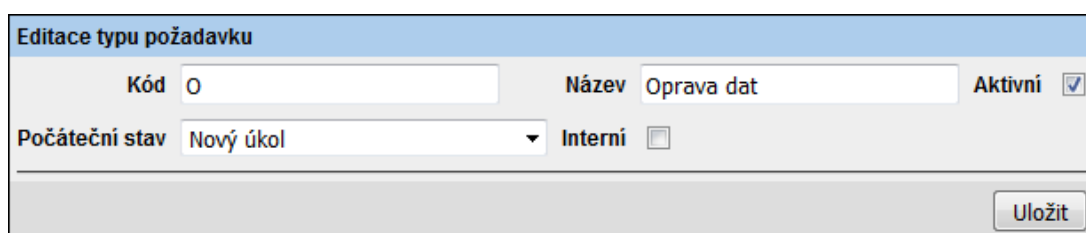
Formulář pro vytvoření typu požadavku je přístupný po kliknutí na záložky Číselníky ⇒ Typ požadavku. Zadaný kód typu požadavku musí být unikátní.

Položka *Počáteční stav* slouží pro nastavení stavu vytvořeného požadavku na základě vybraného typu. Položka *Interní* určuje, zda se jedná o typ určený jen pro interní požadavky nebo typ určený jen pro požadavky zákazníka.

4.5.12 Editace typu požadavku

Formulář pro editaci typu požadavku je přístupný po kliknutí na záložky Číselníky ⇒ Typ požadavku a kliknutím na ikonu pro editaci vybraného řádku v seznamu typů. Formulář můžete vidět na Obrázku 20.

Zadaný kód typu musí být unikátní. Položka *Aktivní* určuje, zda se jedná o platný nebo neplatný záznam číselníku.



Editace typu požadavku			
Kód	<input type="text" value="O"/>	Název	<input type="text" value="Oprava dat"/> Aktivní <input checked="" type="checkbox"/>
Počáteční stav	<input type="text" value="Nový úkol"/>	Interní	<input type="checkbox"/>
<input type="button" value="Uložit"/>			

Obrázek 20: Editace typu požadavku

4.5.13 Vytvoření zákazníka

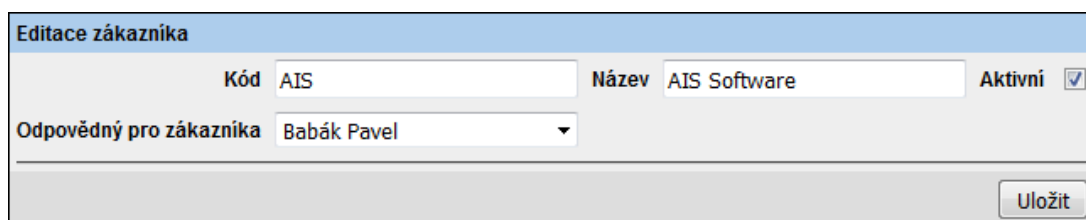
Číselník slouží pro definování zákazníků.

Formulář pro vytvoření zákazníka je přístupný po kliknutí na záložky Číselníky ⇒ Zákazník a kliknutím na ikonu pro editaci vybraného řádku v seznamu zákazníků. Formulář můžete vidět na Obrázku 21.

4.5.14 Editace zákazníka

Formulář pro editaci zákazníka je přístupný po kliknutí na záložky Číselníky ⇒ Zákazník a kliknutím na ikonu pro editaci vybraného řádku v seznamu zákazníků. Formulář můžete vidět na Obrázku 21.

Zadaný kód a název zákazníka musí být unikátní. Položka *Aktivní* určuje, zda se jedná o platný nebo neplatný záznam číselníku.



Editace zákazníka			
Kód	<input type="text" value="AIS"/>	Název	<input type="text" value="AIS Software"/> Aktivní <input checked="" type="checkbox"/>
Odpovědný pro zákazníka	<input type="text" value="Babák Pavel"/>		
<input type="button" value="Uložit"/>			

Obrázek 21: Editace zákazníka

4.6 Požadavky

4.6.1 Vytvoření požadavku

Formulář pro vytvoření požadavku je přístupný po kliknutí na záložky Požadavky ⇒ Vytvoření. Formulář, který se zobrazí zaměstnanci, můžete vidět na Obrázku 22. Na formuláři určeném pro zákazníka některé z položek nejsou, protože zákazník tyto údaje nezadává. Konkrétně se jedná o údaje *Projekt*, *Modul*, *Odpovědný* a *Odpovědný programátor*. V ostatních položkách jsou formuláře totožné.

The screenshot shows a web form titled "Nový požadavek". It contains several input fields and dropdown menus. The "Zákazník" field is pre-filled with "AIS Software". Other fields like "Projekt", "Modul", "Typ", "Priorita", "Termín", "Odpovědný", and "Odpovědný programátor" are currently empty or show "Vyberte hodnotu". There is a "Přidat / odebrat podílejší se" button, a "Příspěvek" text area, and three "Přílohy" sections, each with a "Procházet..." button and "Soubor nevybrán." text. A "Poznámka" text area is at the bottom. A "Vytvořit" button is located in the bottom right corner.

Obrázek 22: Vytvoření požadavku

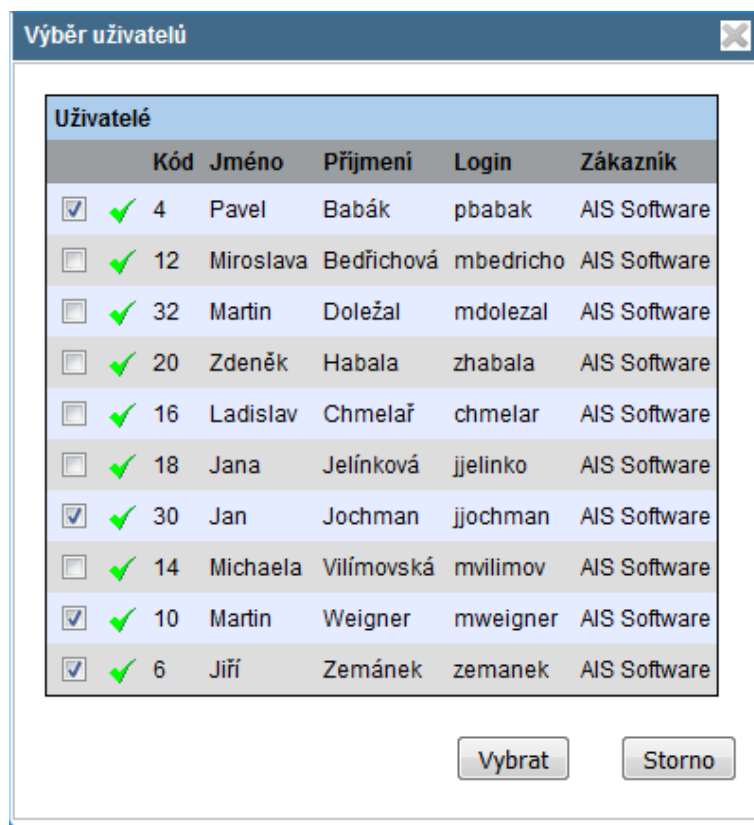
Jednotlivé položky formuláře pro zaměstnance:

- **Název** - Název požadavku
- **Zákazník** - Zákazník se předvyplní dle nastavení zákazníka pro přihlášeného uživatele a tento údaj není možné modifikovat.
- **Projekt** - Projekt, ke kterému požadavek patří.
- **Modul** - Umožňuje zadat část systému, které se bude požadavek převážně týkat.
- **Typ** - Typ požadavku. Specifikuje, zda požadavek představuje Nové zadání, Reklamaci, apod.

- **Priorita** - Priorita požadavku. Při nastavení priority dojde k automatickému přednastavení termínu.
- **Termín** - Termín požadavku určuje, do jakého data požaduje *Zadavatel* splnění požadavku. Termín zadáme buď zápisem ve formátu DD.MM.RRRR (DD je číslo dne v měsíci, MM je číslo měsíce, RRRR je rok, přičemž jednotlivé části data jsou od sebe odděleny tečkou), nebo výběrem z kalendáře. Kalendář otevřeme kliknutím na ikonu kalendáře, která je umístěna vpravo od editačního pole pro zadání termínu.
- **Zadavatel** - Zadavatel požadavku je automaticky přednastaven na uživatele, který požadavek vytváří, je možné ho ale změnit na jiného uživatele, který má stejného zákazníka jako přihlášený uživatel. Toho lze využít, pokud uživatel vytváří požadavek za jiného uživatele.
- **Odpovědný** - Jedná se o zaměstnance, který je odpovědný za vyřešení požadavku. Je to hlavní osoba na požadavku.
- **Odpovědný programátor** - Jedná se o zaměstnance, který je odpovědný za realizaci / implementaci požadavku.
- **Podílející se** - Seznam uživatelů, kteří jsou zapojeni do řešení požadavku. Může se jednat o zaměstnance i zákazníky. Každý uživatel však může do tohoto seznamu přidávat nebo odebírat pouze uživatele, kteří patří ke stejnému zákazníkovi jako on sám. Jinými slovy uživatel může přidávat nebo odebírat pouze své kolegy.
- **Příspěvek** - První příspěvek požadavku zpravidla blíže specifikuje celé zadání požadavku a zadání příspěvku je povinné.
- **Přílohy** - Výběr příloh, které se vztahují k prvnímu příspěvku požadavku. Výběr přílohy se spustí po kliknutí na tlačítko Procházet.
- **Poznámka** - Zapisují se sem informace, které mají být pro ostatní uživatele na první pohled viditelné. Na rozdíl od vytvořeného příspěvku je poznámka editovatelná a informace je tedy možné kdykoliv aktualizovat.

Změna podílejících se uživatelů se spustí kliknutím na tlačítko *Přidat / odebrat podílející se*. Následně se zobrazí modální okno pro výběr uživatelů, které můžete vidět na Obrázku 23. Pomocí zatržitek v levé části seznamu lze přidat nebo odebrat podílející se uživatele. Výběr uživatelů potvrdíme kliknutím na tlačítko Vybrat, nebo zrušíme výběr kliknutím na tlačítko Storno. Po provedení výběru se aktualizuje seznam Podílejících se na formuláři Vytvoření požadavku.

Kliknutím na tlačítko Vytvořit potvrdíme vytvoření požadavku. Pokud jsou údaje na formuláři vyplněny správně, dojde k uložení požadavku do databáze a odeslání notifikačního emailu všem uživatelům připojeným do požadavku. Pokud se rozhodneme požadavek nevytvářet, můžeme formulář zrušit kliknutím na libovolnou záložku v hlavním menu, nebo použitím tlačítka Zpět v prohlížeči.



Obrázek 23: Výběr uživatelů

4.6.2 Vyhledání, editace požadavku

Formulář pro vyhledání požadavků je přístupný po kliknutí na záložky Požadavky ⇒ Vyhledání, editace. Formulář můžete vidět na Obrázku 24.

Uživatel zadá výběrová kritéria, podle kterých chce požadavky vyhledat a výběr potvrdí tlačítkem Vyhledat. Systém zobrazí nalezené požadavky v seznamu. Po kliknutí na ikonu pro editaci položky v seznamu se zobrazí formulář pro editaci požadavku, který nahradí formulář pro vyhledání. Formulář, který se zobrazí zaměstnanci, můžete vidět na Obrázku 25. Zákazník může vytvářet pouze veřejné příspěvky, a proto se mu volba *Interní příspěvek* nezobrazuje. Zákazník také na rozdíl od zaměstnance nemůže modifikovat tyto údaje: *Projekt*, *Modul*,

Požadavek (vyhledávací kritéria)

Zákazník: Vyberte hodnotu | Id: | Id (od): | Id (do):

Název: | Projekt: Vyberte hodnotu | Modul: Vyberte hodnotu

Typ: Vyberte hodnotu | Stav: Vyberte hodnotu | Priorita: Vyberte hodnotu

Zadavatel: Vyberte hodnotu | Odpovědný: Vyberte hodnotu | Odpovědný programátor: Vyberte hodnotu | Podílejší se: Vyberte hodnotu

Termín (od): | Termín (do): | Vytvořen (od): | Vytvořen (do):

Vyhledat

Obrázek 24: Vyhledání požadavků

Stav, Pracnost, Odpovědný a Odpovědný programátor. V ostatních položkách jsou formuláře totožné.

Editace požadavku

Název Změna vytváření hlášení

Zákazník AIS Software Id 1 Typ Interní úkol

Projekt Sirael jádro Modul Pojistné události Stav Předáno k analýze

Priorita Střední Termín 20.6.2016 Pracnost 20

Zadavatel Babák Pavel Odpovědný Jochman Jan Odpovědný programátor Zemánek Jiří

Přidat / odebrat podílejší se

Podílejší se Weigner Martin

Interní příspěvek

Nový příspěvek

Nové přílohy Procházet... Soubor nevybrán.
Procházet... Soubor nevybrán.
Procházet... Soubor nevybrán.

Poznámka

Uložit

Obrázek 25: Editace požadavku

Jednotlivé položky formuláře:

- **Název** - Název požadavku
- **Zákazník** - Zákazník požadavku. Údaj není modifikovatelný.
- **Id** - Id je vygenerováno systémem při vytvoření požadavku a nelze jej modifikovat. Kombinace údajů *Zákazník* a *Id* tvoří jednoznačnou identifikaci požadavku.
- **Typ** - Typ požadavku. Údaj není modifikovatelný.
- **Projekt** - Projekt, ke kterému požadavek patří.
- **Modul** - Umožňuje zadat část systému, které se bude požadavek převážně týkat.
- **Stav** - Stav požadavku
- **Priorita** - Priorita požadavku. Při nastavení priority dojde k automatickému přednastavení termínu.

- **Termín** - Termín požadavku určuje, do jakého data požaduje *Zadavatel* splnění požadavku. Termín zadáme buď zápisem ve formátu DD.MM.RRRR (DD je číslo dne v měsíci, MM je číslo měsíce, RRRR je rok, přičemž jednotlivé části data jsou od sebe odděleny tečkou), nebo výběrem z kalendáře. Kalendář otevřeme kliknutím na ikonu kalendáře, která je umístěna vpravo od editačního pole pro zadání termínu.
- **Pracnost** - Představuje odhadovaný počet hodin potřebný ke splnění (dokončení) požadavku. Jedná se o odhadovaný součet hodin práce všech zaměstnanců, kteří se na požadavku budou podílet. Tento údaj zpravidla zadává *Odpovědný* při nacenění požadavku. Zákazník poté musí odsouhlasit odhadnutý počet hodin, aby mohly začít práce na realizaci. U interních požadavků, tj. požadavků vytvářených zaměstnanci pro interní firemní potřeby, má tento údaj pouze informativní charakter, protože neslouží k fakturaci.
- **Zadavatel** - Zadavatel požadavku. Údaj není modifikovatelný.
- **Odpovědný** - Jedná se o zaměstnance, který je odpovědný za vyřešení požadavku. Je to hlavní osoba na požadavku.
- **Odpovědný programátor** - Jedná se o zaměstnance, který je odpovědný za realizaci / implementaci požadavku.
- **Podílející se** - Seznam uživatelů, kteří jsou zapojeni do řešení požadavku. Může se jednat o zaměstnance i zákazníky. Každý uživatel však může do tohoto seznamu přidávat nebo odebírat pouze uživatele, kteří patří ke stejnému zákazníkovi jako on sám. Jinými slovy uživatel může přidávat nebo odebírat pouze své kolegy.
- **Interní příspěvek** - Tato volba slouží k vytvoření interního příspěvku, který uvidí pouze zaměstnanci.
- **Nový příspěvek** - Text nového příspěvku. Pokud uživatel nezadá žádný text, nový příspěvek se nevytváří.
- **Přílohy** - Přílohy, které se vztahují k příspěvku požadavku. Výběr přílohy se spustí po kliknutí na tlačítko Procházet. Přílohy se uloží pouze v případě, že je zadán i text příspěvku. Není tedy možné ukládat přílohy bez doprovodného textu.
- **Poznámka** - Zapisují se sem informace, které mají být pro ostatní uživatele na první pohled viditelné. Na rozdíl od vytvořeného příspěvku je poznámka editovatelná a tyto informace je tedy možné kdykoliv aktualizovat.

Změna podílejících se uživatelů se provádí stejným způsobem jako na formuláři pro Vytvoření požadavku.

Napravo od formuláře se zobrazuje seznam příspěvků požadavku, který můžete vidět na Obrázku 26. Číslo udává pořadí příspěvku dle času jeho vytvoření. Pro číslování interních příspěvků se k číslu posledního veřejného příspěvku doplní tečka a pořadí interního příspěvku od posledního veřejného příspěvku (např.: číslo 3.2 označuje interní příspěvek, který je druhým interním příspěvkem, který byl vytvořen po 3 (posledním) veřejném příspěvku). Číslo příspěvku je praktický údaj, protože se můžeme v popisu příspěvku odkázat na jiný příspěvek prostřednictvím jeho čísla. V seznamu je popis jednotlivých příspěvků zkrácen. Pro zobrazení celého popisu příspěvku slouží modální okno, které se spustí kliknutím na ikonu s lupou na příslušném řádku seznamu. V modálním okně se kromě nezkráceného popisu příspěvku zobrazí i seznam příloh. Tyto přílohy je možné stáhnout a uložit na lokální disk.

Příspěvky požadavku						
	Číslo	Zkrácený popis	Vytvořen	Vytvořil	Interní	Příloh
	1	Popis požadovaných změn je v příloze.	09.05.2016 22:39:29	Babák Pavel	Ne	1
	2	@PB: Posílám připomínky k zadání.	09.05.2016 22:50:52	Jochman Jan	Ne	1

Obrázek 26: Příspěvky požadavku

Kliknutím na tlačítko Uložit na formuláři pro editaci požadavku potvrdíme provedené změny. Pokud jsou údaje na formuláři vyplněny správně, dojde k uložení změn do databáze a odeslání notifikačního emailu uživatelům připojeným do požadavku. Pokud se rozhodneme editaci požadavku ukončit bez uložení změn, můžeme formulář zrušit kliknutím na libovolnou záložku v hlavním menu, nebo použitím tlačítka Zpět v prohlížeči.

Závěr

Cílem práce bylo vyvinout webovou aplikaci, která bude umožňovat zákazníkům zadávat požadavky na změnu softwaru a zaměstnancům softwarové společnosti tyto požadavky zpracovávat a reagovat na ně. Zákazníci i zaměstnanci mohou v průběhu řešení požadavku zadávat své příspěvky (komentáře) a k těmto příspěvkům přidávat i přílohy. Aplikace pak o těchto příspěvcích informuje prostřednictvím emailu ostatní uživatele. Při odesílání emailů o změnách na požadavku je dále zohledněno, že zákazníka zajímají jen určité změny na požadavku, zatímco zaměstnanci jsou prostřednictvím emailu informováni o všech změnách. Aplikace také umožňuje uživatelům sledovat aktuální stav požadavku, procházet příspěvky a stahovat si přílohy příspěvků.

Vhodným rozšířením aplikace by bylo přidání výkazu práce zaměstnanců, ve kterém by zaměstnanci identifikovali svoji činnost ke konkrétnímu požadavku. Tyto výkazy práce by pak sloužily pro fakturaci práce zákazníkovi a vytvářely by také přehled o činnostech jednotlivých zaměstnanců.

A Obsah přiloženého DVD

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletní projekt z Eclipse v ZIP archivu.

sql/

Sql soubor (dump) pro vytvoření databázového schématu.

webapp/

War soubor pro nasazení aplikace na Tomcat a konfigurační soubory.

install/

Instalace Javy, Apache Tomcat a MySQL pro různé platformy.

readme.txt

Instrukce pro instalaci a spuštění webové aplikace.

Literatura

- [1] BAUER, Christian; KING, Gavin. *Hibernate in Action*. Manning Publications, 2004, 400 stran, ISBN 9781932394153.
- [2] DASHORST, Paul, Martijn; HILLENIOUS, Eelco. *Wicket in Action*. Manning Publications, 2008, 392 stran, ISBN 9781932394986. Dostupné z: <https://www.manning.com/books/wicket-in-action>
- [3] SPELL, Brett. *Java - Programujeme profesionálně*. Computer Press, Praha, 2002, ISBN 80-7226-667-5.
- [4] Hibernate Tutorial [online]. Dostupné z: <http://www.tutorialspoint.com/hibernate/>
- [5] Hibernate Tutorial - Hibernate Community Documentation [online]. Dostupné z: <https://docs.jboss.org/hibernate/orm/3.3/reference/en/html/tutorial.html>
- [6] Apache Wicket User Guide - Reference Documentation [online]. Dostupné z: <https://ci.apache.org/projects/wicket/guide/6.x/guide/single.html>
- [7] HTML and CSS Tutorial [online]. Dostupné z: <http://www.w3schools.com/>