

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Webový komunikátor pro technickou podporu

WebChat for Technical Support

2010

Jan Folwarczny

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Jan Folwarczny**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Webový komunikátor pro technickou podporu
WebChat for Technical Support**

Zásady pro vypracování:

Cílem práce je rozšířit webový komunikační systém pro příjem a zpracování emailových požadavků technické podpory firmy. Výsledkem práce musí být funkční systém, tak aby v něm bylo možno zpracovávat požadavky více operátory. Požadavky bude možno dále mezi operátory rozdělovat podle zadaných kritérií. Práce musí obsahovat především:

1. Analýzu řešení stávajícího systému a problému zpracování více požadavků více operátory.
2. Návrh řešení za použití volně dostupných technologií.
3. Implementace navrženého řešení.
4. Výsledky testování.
5. Uživatelskou a programátorskou dokumentaci.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Marian Mindek, Ph.D.**

Datum zadání: 20.11.2009

Datum odevzdání: 07.05.2010



doc. Dr.Ing. Eduard Sojka
vedoucí katedry



prof. Ing. Ivo Vondrák, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 7.5.2010

A handwritten signature in blue ink, reading "Jan Fojtík", is written over a horizontal line.

Rád bych na tomto místě poděkoval panu Ing. Mariánu Mindkovi, Ph.D. za vedení a rady pro vypracování bakalářské práce a panu Bc. Pavlovi Krkoškovi za poskytnutí informací a dat potřebných pro praktickou část práce.

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě dne 6.5.2010

BANAN S.R.O. www.banan.cz
IČ: 26867257, DIČ: CZ26867257
Slavíkova 1744/22
708 00 Ostrava, CZ ©
tel. +420 596 630 274

Abstrakt

Cílem této bakalářské práce je úprava a rozšíření systému pro emailovou komunikaci s technickou podporou firmy poskytující webhostingové služby.

Součástí práce je analýza současného stavu systému, zvláště částí, které se týkají požadovaných úprav, a popis jeho problematických míst. Práce zahrnuje rozbor důvodů nasazení takového systému, návrh a postup při implementaci úprav, řešení problematických částí a výčet provedených rozšíření uživatelského rozhraní. Obsahem práce je také popis použitých technologií, postup výsledného testování a jednoduchá uživatelská a programátorská příručka popisující použití provedených změn v systému.

Klíčová slova

systém, dotaz, technická podpora, emailová schránka, uživatelské rozhraní, tazatel, uživatel

Abstract

The aim of this thesis is a modification and extension of the system for email communication with technical support of company that provides web hosting services.

The thesis contains an analysis of the current state of the system, especially the part related to the requested modifications, and a description of the bottlenecks. The thesis includes the analysis due to the deployment of such a system, design and implementation process of adjustment, problem solving and a list of parts made by extending the user interface. The thesis contains also a description of used technologies, final testing process and simple user and programmer's manual describing the use of the changes in the system.

Keywords

system, question, technical support, mailbox, user interface, the questioner, user

Seznam použitých symbolů a zkratek

AJAX	- Asynchronous JavaScript and XML
API	- Application Programming Interface
ASP.NET	- Active Server Pages .NET
CSS	- Cascading Style Sheets
DBMS	- Database management system
DOM	- Document Object Model
FF	- Mozilla Firefox
GNU GPL	- GNU General Public License
HTML	- HyperText Markup Language
HTTP	- Hypertext Transfer Protocol
IE	- Internet Explorer
J2EE	- Java 2 Platform, Enterprise Edition
JSON	- JavaScript Object Notation
MIME	- Multipurpose Internet Mail Extensions
PHP	- PHP: Hypertext Preprocessor
POP3	- Post Office Protocol version 3
SQL	- Structured Query Language
W3C	- World Wide Web Consortium
XML	- Extensible Markup Language

Obsah

1. Úvod	9
2. Možnosti komunikace s technickou podporou	10
2.1. Rozbor prostředků komunikace	10
2.2. Důvody nasazení systému	10
3. Analýza stávajícího stavu	12
3.1. Popis možností stávajícího systému	12
3.1.1. Vyzvednutí a přidělení dotazu	13
3.1.2. Přeposlání dotazu	13
3.1.3. Pozastavení zpracování dotazu	13
3.1.4. Další možnosti	14
3.2. Nevýhody aktuálního systému a požadované úpravy	14
3.3. Rozbor implementace stávajícího řešení	15
3.3.1. Struktura scriptů	15
3.3.2. Aktivace a výběr aktivního uživatele	16
4. Použité technologie	20
4.1. PHP, Apache HTTP Server, MySQL, CRON	20
4.2. AJAX	21
4.3. jQuery	22
5. Návrh řešení a implementace	23
5.1. Roztřídění a aktualizace	23
5.2. Aktivace více uživatelů	25
5.3. Výběr uživatele pro přidělení dotazu	25
5.3.1. Návrhy řešení	25
5.3.2. Výsledné řešení	27
5.3.3. Přřazení dotazu při neexistenci žádného aktivního uživatele	28
5.4. Vyzvedávání dotazů	29
5.5. Pozastavení a přeposlání dotazů	30
5.6. Úpravy uživatelského rozhraní	31

6. Testování	34
7. Uživatelská a programátorská dokumentace	35
7.1. Instalace systému	35
7.2. Nastavení skupin	36
7.3. Rozdělení upravených zdrojových kódů	37
8. Závěr	38
9. Literatura	39
10. Přílohy	40

1. Úvod

Tématem této bakalářské práce je úprava stávajícího systému pro emailovou komunikaci s technickou podporou firmy. Jedná se o systém, který umožňuje příjem zpráv zaslaných na emailovou adresu technické podpory a rozdělování těchto zpráv mezi zaměstnance firmy, kteří jsou do systému přihlášení. Ti pak mohou se zprávami dále pracovat. Mohou tazateli odpovědět, přeposlat zprávu jinému zaměstnanci či oddělení a provádět další operace. O vyzvedávání zpráv z emailové schránky se stará vždy jeden zaměstnanec firmy.

Cílem této bakalářské práce je upravení a rozšíření systému tak, aby možnost vyzvedávání zpráv byla dostupná pro více zaměstnanců. Rozdělování zaslaných zpráv by mělo být automatické a tato funkce by měla být dostupná i v interní části systému při manipulaci se zprávami v rámci jiných oddělení firmy.

V následující 2. kapitole s názvem „Možnosti komunikace s technickou podporou“ jsou rozebrány možné způsoby komunikace s technickou podporou a důvody nasazení systému pro centralizaci emailové komunikace. Navazující 3. kapitola, pojmenovaná „Analýza stávajícího stavu“, obsahuje rozbor používaného systému z hlediska jeho funkčnosti i implementace. Zaměřená je na části systému, které bylo nutno upravit a jsou zde rozebrány chyby, jenž se ve stávající implementaci mohou objevovat. Ve 4. kapitole „Použité technologie“, jsou krátce popsány jednotlivé technologie využívané v původním i upraveném systému. Pátá kapitola nazvaná „Návrh řešení a implementace“ obsahuje postup při úpravách systému, popis navržených a výsledných řešení a jejich implementaci. Nachází se zde také popis důvodů vzniku nových prvků uživatelského rozhraní. Kapitola 6. „Testování“ pak uvádí postupy, které byly při konečném testování použity a jaké chyby byly nalezeny. V 7. kapitole „Uživatelská a programátorská dokumentace“ se nachází popis změn, které vznikly díky úpravám, a postup při instalaci systému.

2. Možnosti komunikace s technickou podporou

Firma, která poskytuje služby v oblasti webhostingu a domén, mající téměř deset tisíc klientů potřebuje kvalitní technickou podporu. Ta musí být schopna reagovat jak na dotazy zákazníků, tak i na dotazy ostatních tazatelů. Způsob dotazování technické podpory by měl být pro tazatele co nejjednodušší. Mělo by být možné také komunikovat přes více prostředků jako jsou telefon, email, webchat a případně další média.

2.1. Rozbor prostředků komunikace

Jedna z možností komunikace je tedy emailová. Ta díky dostupným technickým prostředkům dovoluje jednoduše ukládat proběhlou komunikaci, dále ji zpracovávat a lehce s ní manipulovat. Je to velká výhoda například oproti telefonní komunikaci. U té by bylo velmi náročné proběhlou komunikaci už jen zaznamenávat. Rozhovor by se musel například nahrávat a zároveň ukládat. Takto nahrané soubory by byly velmi oběmné, což by znamenalo vysoké náklady na provoz i údržbu úložiště, ale také hlavně manipulace s těmito informacemi by byla velmi pomalá a neefektivní. V případě, kdy by se chtěl někdo ke zprávě vrátit, musel by vždy poslechnout všechny proběhle rozhovory a až podle nich reagovat. Což je v dnešní době nereálné.

Naproti tomu již zmíněná emailová komunikace je pro požadavky ukládání a jednoduché správy velmi vhodná. Samotný textový formát, ve kterém jsou emailové zprávy napsány jsou velice úsporné a dobře se s nimi manipuluje. V obsahu zprávy se dá jednoduše vyznat, takže přístup k informacím je velice rychlý. Proto je pro účely technické podpory velmi výhodné tento známý a často používaný způsob komunikace využívat.

2.2. Důvody nasazení systému

Ve větší firmě s více zaměstnanci, kde každý zaměstnanec má na starosti svůj segment, není možné mít jednu pracovní pozici, která by dokázala vyřizovat a zpracovávat všechny možné typy dotazů, které mohou tazatelé mít. Pro člověka, který se potřebuje na firmu obrátit s nějakým dotazem tak nastává problém ve výběru konkrétního zaměstnance, který by byl schopen odpovědět na jeho dotaz.

Jedna možnost je veřejně poskytnout (například na webu firmy) o zaměstnancích informace, jakou funkci vykonávají a s jakými otázkami je se na ně možné obrátit. Tazatelé by si tak museli sami vybrat příslušného zaměstnance.

Komunikace s technickou podporou by pro koncového uživatele měla být co nejjednodušší. Je tedy dobré celý proces vyřizování dotazů zajistit tak, aby tazatel neměl žádné další starosti, všelijaké vyřizování a nebyl odkazován na žádné další postupy.

Další možnost tedy je zavedení systému, který by tazatele zbavoval těchto problémů a centralizoval by veškerou komunikaci mezi nimi a zaměstnanci. Pro vyřizování dotazů je tedy důležité, aby tazatel nemusel hledat kontakt na určitou osobu nebo oddělení firmy, které by bylo schopné na jeho dotaz odpovědět, vyhovět či jinak reagovat. Proto je dobré mít jednu emailovou adresu pro všechny druhy dotazů a proces, který každý dotaz přidělí kompetentním zaměstnanci.

Tato komunikace by pro tazatele vypadala, jako by komunikoval s jedním člověkem. Bylo by to pro něho mnohem jednodušší než zjišťování, na koho se má vůbec obrátit a následně třeba zjistit, že potřebnou informaci dostane někde jinde a celý proces znova opakovat.

Další z důvodů nasazení takového systému je kromě rozdělování dotazů také skladování historie komunikace včetně zaznamenání informací o akcích, které se s dotazy prováděly. Pak je jednoduché například vyhledat dotaz, který už byl zpracován, zjistit co a kdy se s ním dělá, a ná něho například navázat v dalším řešení dotazu.

3. Analýza stávajícího stavu

Prvním postupem bylo projít aktuálně nasazený systém a zjistit, co vůbec přesně umožňuje, jak pracuje a jak je napsán. Abych tuto analýzu mohl provést a nezasahoval do reálných dat, dostal jsem k dispozici kopii nainstalovanou na vlastní doméně a jednu testovací emailovou schránku.

3.1. Popis možností stávajícího systému

Systém primárně umožňuje vyzvedávání a správu dotazů zaslaných na email technické podpory. V systému je zaevidováno několik uživatelů (zaměstnanců firmy), kteří jsou rozděleni do skupin. Každá skupina reprezentuje jedno oddělení firmy (např. Technická podpora, Správa domén, Účetnictví, apod.).

Pro emailovou schránku, ze které se vyzvedávají zaslané zprávy je vybrán jeden uživatel, kterému se příchozí zprávy přiřklují. Tento uživatel pak má s přidělenými dotazy možnosti provádět několik operací. Může dotaz vyřídit sám, tak že tazateli odpoví. Může ho přeposlat jinému uživateli v rámci své skupiny nebo ho přeposlat jiné skupině. Takto se dá zpráva předávat mezi různá oddělení a zaměstnance firmy podle toho, pro koho je dotaz určen. Cílový uživatel pak tento dotaz zpracuje.

Systém je tedy navržen tak, že zákazníci firmy nebo jiní tazatelé nemusí vybírat konkrétního zaměstnance a jeho emailovou adresu, kterému svůj dotaz odešlou, ale pro zasílání dotazů existuje jen jedna emailová schránka, pomocí které se vyřizují všechny příchozí dotazy.

Uživatelé jsou rozděleni do dvou rolí – administrátoři a klasičtí uživatelé. Administrátoři jsou pevně určení a nedají se přes uživatelské rozhraní přidávat nebo odebírat. Mohou však vytvářet klasické uživatele.

Historie operací s dotazy je ukládána. Operace, které se se zprávou provádějí jsou zaznamenávány do logu. V něm jsou zaznamenány všechny operace, jako vyzvednutí dotazů, přiřazení dotazů uživatelům, uzavření dotazu apod.

Výpisy dotazů jsou rozděleny na skupiny:

- **Mé dotazy** – v této skupině jsou všechny dotazy přiřazené aktuálnímu uživateli, které ještě nebyly zpracovány.
- **Mé skupiny** – v této skupině jsou všechny nezpracované dotazy ostatních uživatelů skupin, do kterých patří i přihlášený uživatel. Jsou zde i dotazy, které nejsou přiděleny uživatelům, ale přímo jen skupině (vysvětleno v kapitole 3.1.3 Pozastavení zpracování dotazu).

- **Ostatní skupiny** – všechny nezpracované dotazy patřící do skupin, kterých není přihlášený uživatel členem.
- **Zodpovězené** – dotazy, na které přihlášený uživatel odpověděl.
- **Koš** – dotazy, které přihlášený uživatel odstranil.

3.1.1. Vyzvednutí a přidělení dotazu

Při zaslání dotazu na email technické podpory se vyzvedne dotaz z mailové schránky a musí se přiřadit některému z uživatelů, který pak s dotazem dále pracuje. Při existenci několika skupin a v nich několika uživatelů se musí systém rozhodnout, komu vyzvednutý dotaz přiřadit.

V nynější podobě systému je to řešeno tak, že pro emailovou adresu je vybrána jedna skupina, která je nastavena jako výchozí pro příjem zpráv. Toto nastavení může měnit administrátor. Jelikož se zpráva musí přiřadit některému uživateli, je z každé skupiny vybrán jeden uživatel, kterému budou tyto zprávy v rámci skupiny přiřazeny. Proto každý uživatel skupiny, která sbírá dotazy ze schránky má možnost se aktivovat nebo deaktivovat.

Pokud se uživatel aktivuje, systém pak všechny příchozí zprávy přiřazuje právě tomuto uživateli. U ostatních skupin, které nejsou nastaveny pro příjem zpráv je administrátorem vybrán jeden uživatel, kterému budou přiřazovány přeposlané dotazy.

Po úspěšném vyzvednutí dotazu z emailové schránky se tazateli odešle zpráva o zpracování dotazu a je vygenerováno id dotazu – v systému se nazývá ticket. Poté, co uživatel na tento dotaz odpoví, bude do odpovědi ticket zařazen. Jestliže tazatel na tuto odpověď také reaguje, tak systém při vyzvedávání dotazu rozpozná ticket, vygenerovaný v minulém dotazu a zařadí ho do vlákna společně s minulým dotazem. Toto se může několikrát opakovat. Je pak možné vyhledat celou komunikaci.

3.1.2. Přeposlání dotazu

Uživatel, kterému přišla zpráva na níž není schopen odpovědět jí může přeposlat jinému uživateli. V rámci systému má možnost přeposlání buď konkrétnímu uživateli skupin, kterých je on sám také členem, nebo jí může předat nějaké skupině.

Oproti prvnímu případu, kdy přeposílá konkrétnímu uživateli, se u přeposlání na skupinu musí zase vybrat jeden její uživatel. V případě, kdy se dotaz přeposílá na skupinu, která není nastavena jako výchozí pro příjem zpráv ze schránky, tak se dotaz přidělí uživateli, který je napevno administrátorem nastaven jako příjemce v rámci skupiny. V opačném případě, kdy se přeposílá na skupinu, která je výchozí pro příjem, se vybere aktivní uživatel.

3.1.3. Pozastavení zpracování dotazu

Další důležitou funkcí systému, která také souvisí s přiřazováním zpráv uživatelům je pozastavování. V případě, kdy uživatel nebo celá jeho skupina není schopná okamžitě reagovat na dotaz (například při čekání na dokončení bankovní transakce nebo jiných událostech, které

nemohou uživatelé ovlivnit), je možné dotaz odložit o uživatelem zvolenu dobu. Tím se zruší přiřazení dotazu uživateli, kterému náležel. Zůstane ale přiřazen skupině.

Po uplynutí doby, na jakou byl dotaz pozastaven, se dotaz automaticky přidělí uživateli skupiny, které náleží. Stejně jako u přeposlání dotazů je to buď aktivní uživatel skupiny, která je výchozí pro příjem, nebo napevno aktivovaný uživatel v rámci některé z ostatních skupin.

3.1.4. Další možnosti

Kromě práce s dotazy obsahuje systém také několik dalších funkcí. Jedná z nich, která sice také trochu souvisí s dotazy je vytváření sbírky často kladených otázek. Do této sbírky se tak mohou přidávat celá vlákna již vyřešených dotazů.

Další část systému je interní komunikace mezi uživateli a skupinami. Do této části spadají interní zprávy a úkoly. Interní zprávy uživatel může, pokud na to má přidělené právo, posílat buď konkrétnímu uživateli nebo skupině. Jestliže pošle zprávu skupině, dostanou ji všichni její uživatelé. Uživatel pak v systému vidí své přijaté a odeslané zprávy.

Úkoly může uživatel zadávat jinému uživateli. Také pro tuto operaci musí mít přidělené právo. Uživatel, který dostal nějaký úkol ho může poslat ke schválení zadavateli úkolu. O schválení úkolu se stará jeho zadavatel. Každý uživatel vidí své nesplněné a splněné úkoly, a úkoly, které sám zadal.

3.2. Nevýhody aktuálního systému a požadované úpravy

Hlavní nevýhodou tohoto systému je, že není víceuživatelský z pohledu zpracovávání dotazů. Je sice možné, aby do systému bylo přihlášeno více uživatelů, klidně i z jedné skupiny, ale může být aktivován pro příjem zpráv v rámci skupiny vždy jen jeden z nich. Problém tedy je hlavně ve vyzvedávání dotazů z mailboxu.

V aktuální verzi je pro každou emailovou schránku přiřazena jedná skupina. V této skupině se pak zpráva vyzvednutá ze schránky přiřadí právě jednomu aktivnímu uživateli. Ten ji následně může zpracovat sám nebo přeposlat. V případě, že je v systému přihlášeno více uživatelů, patřících do stejné skupiny, která je nastavena pro příjem zpráv, není možné, aby se také aktivovali. Vše tedy musí přijímat vždy jen jeden uživatel a sám vše rozdělovat.

Podobná situace je u přeposílání dotazů skupině, tam nastává stejný problém. Vždy dojde k přiřazení tomu jednomu aktivnímu uživateli ze skupiny.

K možnosti, kdy může být aktivní jen jeden uživatel skupiny se váže další nevýhoda. Pokud jeden uživatel patří do dvou nebo více skupin a aktivuje se, přebere ze všech svých skupin zodpovědnost za přebírání dotazů – deaktivuje minulé aktivní uživatele těchto skupin. Není potom možné, aby pro jednu skupinu byl aktivní minulý uživatel a nebyl deaktivován jiným uživatelem.

Požadované úpravy

Hlavní úpravou tedy je umožnit v systému aktivaci několika uživatelů najednou jak v rámci celého systému, tak v rámci skupiny. Mělo by být možné pro každou skupinu mít několik aktivovaných uživatelů, mezi které se budou příchozí a přeposlané dotazy rozdělovat. Toto rozdělování by mělo být automatické a nemělo by mít žádnou další manuální režii.

3.3. Rozbor implementace stávajícího řešení

Systém je vytvořen jako webová aplikace. Přístupná je jen zevnitř firmy, běžní návštěvníci internetu se k němu nedostanou. Je napsaná ve scriptovacím jazyce PHP. Data jsou ukládána v databázi MySQL s enginem MyISAM. Výběr zprávy z emailové schránky probíhá pomocí protokolu POP3.

Jelikož jsem neměl k dispozici žádnou dokumentaci ani jiné materiály popisující rozvržení a implementaci systému, musel jsem funkčnost zjišťovat přímo z kódu a databáze.

Z kódu šlo poznat, že už prošel několika úpravami, a že na něm nejpravděpodobněji pracovalo více programátorů. Byly zde použity také prvky starších verzí php (například práce s proměnnými předaných pomocí metody GET protokolu HTTP) a zkrácené značení úseků PHP kódu, které je v základním nastavení PHP vypnuto.

V případě, kdy by bylo toto nastavení zakázáno (v `php.ini` by nebyla nastavena hodnota `short_open_tag = on`) a spustily by se skripty se zkráceným zápisem, neprovedl by se tak zapsaný kód, ale jen by se vypsal na obrazovku. Proto jsem tento zápis v úpravách převedl na klasický plný. Rozdíly v zápisech jsou popsány níže.

<code><?php PHP kód ?></code>	- plný zápis otevření php kódu
<code><? PHP kód ?></code>	- zkrácený zápis
<code><?=\$promenna ?></code>	- zkrácený zápis pro výpis hodnoty za rovnítkem (může se jednat o obsah proměnné, výraz nebo literál)

3.3.1. Struktura scriptů

Celá funkčnost systému je rozložena do několika hlavních scriptů, ve kterých se postupně podle požadavků načítají (příkazy `include` nebo `require`) další vnořené skripty

Scripty, které řeší funkčnost a vykreslují uživatelské rozhraní:

<code>index.php</code>	- tento soubor slouží jen jako přihlašovací formulář, v případě úspěšného přihlášení přesměruje na <code>main.php</code>
<code>main.php</code>	- hlavní soubor, provádí procesy pro přihlašování, aktivaci a další funkce s uživateli, generuje HTML kód pro stránky, vkládá další skripty, které reprezentují jednotlivé podstránky

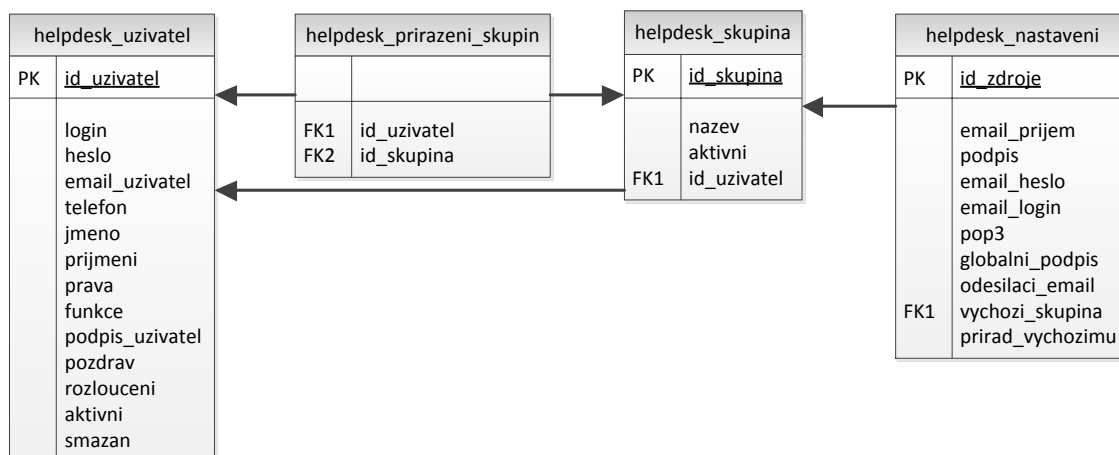
Scripty provádějící automatické úkony:

- precti_mail.php - script, který pomocí POP3 protokolu vyzvedává příchozí emaily ze schránky a přiřazuje je aktivnímu uživateli výchozí skupiny pro příjem
- kontrola_pozastavene.php - kontroluje pozastavené dotazy, v případě, že čas pozastavení již vypršel, přidělí dotaz aktivnímu uživateli skupiny, ke které je dotaz přidělený
- pravidelne_ukoly.php - kontroluje a odesílá vytvořené pravidelné úkony

Všechny 3 scripty jsou nastaveny v CRONu a pravidelně spouštěny. Tím se zaručí automatické zpracování, kdy například v případě vyzvednutí emailu ze schránky se tazateli oznámení pošle ihned. Jestliže by se dotazy vyzvedávali až při načítání stránky aktivním uživatelem, mohlo by dojít k situaci, že tazatel zašle dotaz a ten bude uživatelem vyzvednut až třeba po několika hodinách. Tazatel by po tento čas nebyl nijak informován a nevěděl by, zda jeho zpráva došla.

3.3.2. Aktivace a výběr aktivního uživatele

V aktuální verzi je pro každou emailovou schránku vytvořen záznam v tabulce helpdesk_nastaveni. Atribut vychozi_skupina odkazuje na primární klíč id_skupina v tabulce helpdesk_skupina. Tedy každé emailové schránce je přiřazena jedna skupina. Celé propojení uživatelů skupin a schránek je zobrazen na obrázku Obrázek 3.1.



Obrázek 3.1 Část datového modelu popisující přiřazení uživatelů do skupin a výběr aktivního uživatele

V tabulce helpdesk_skupina jsou atributy aktivni a id_uzivatel. Hodnota id_uzivatel obsahuje vždy aktivního uživatele pro danou skupinu. Ten je vybrán buď administrátorem na pevně nebo je to jeden uživatel skupiny, který se sám aktivuje. Pro tuto možnost je zde atribut aktivni. Ten nabývá pouze dvou hodnot – 0 nebo 1. Pokud je

nastavena hodnota 0, znamená to, že aktivní uživatel je nastaven napevno administrátorem a uživatelé skupiny nemají možnost se sami aktivovat. Pokud atribut nabývá hodnoty 1, mají uživatelé skupiny možnost sami se aktivovat a deaktivovat a do atributu `id_uzivatel` se po aktivaci ukládá identifikátor aktivovaného uživatele. Jestliže není nikdo aktivován, tak `id_uzivatel=0`.

Je zde ale problém, že hodnota `aktivni=1` lze administrátorem zadat pouze pro jednu skupinu. Je to zřejmě proto, že systém byl dříve navržen jen pro jednu emailovou schránku, takže potřeba možnosti aktivace uživatele nebyla u ostatních skupin potřeba. Tento nedostatek by bylo dobré také potřeba vyřešit.

V tabulce `helpdesk_uzivatel` je také atribut `aktivni`, určující zda je uživatel aktivovan nebo ne. Při možnostech, kdy může být aktivní vždy jen jeden uživatel a to v rámci celého systému, protože možnost aktivovat se je dostupná vždy jen v jedné skupině, je tento atribut zbytečný. Informace o tom, který uživatel je aktivní se dá totiž zjistit z tabulky `helpdesk_skupina`, z hodnoty atributu `id_uzivatel` u záznamu, který má nastaveno `aktivni=1`.

Výběr uživatele při vyzvedávání zpráv z emailové schránky

Při vyzvedávání zpráv z emailové schránky je výběr uživatele závislý na tom, která skupina je nastavena jako výchozí. Z této skupiny je pak vybrán jeden aktivní uživatel pomocí následujícího SQL příkazu.

```
SELECT helpdesk_uzivatel.id_uzivatel
FROM helpdesk_uzivatel, helpdesk_prirazeni_skupin
WHERE helpdesk_uzivatel.id_uzivatel =
      helpdesk_prirazeni_skupin.id_uzivatel
      AND aktivni = 1
      AND id_skupina = $vychozi_skupina
```

Existuje zde ale možnost nastavit pro emailovou schránku, aby se zpráva přidávala vždy výchozímu uživateli skupiny. Toto je nastaveno podle atributu `pridat_vychozimu` (viz schéma na obrázku Obrázek 3.1). Pokud v záznamu má atribut hodnotu "on", tak se při vyzvedávání pošty nevybere uživatel podle výše uvedeného dotazu, ale použije se následující.

```
SELECT id_uzivatel
FROM helpdesk_skupina
WHERE id_skupina = $vychozi_skupina
```

Zde nastává logická chyba. Pokud je u skupiny, která je pro tuto schránku nastavena jako aktivní (což znamená, že se zde uživatelé mohou sami aktivovat), tak při aktivaci uživatele se změní hodnota atributu `id_uzivatel` v tabulce `helpdesk_skupina` na identifikátor právě aktivovaného uživatele. V tomto případě je tedy jedno, zda bude hodnota nastavena na "on"

nebo ne. Buď se vybere uživatel podle atributu `aktivni` z tabulky `helpdesk_uzivatel`, nebo se vybere aktivní uživatel rovnou z tabulky `helpdesk_skupina`, u obou případů se ale bude jedna o stejného uživatele.

Pokud bude u skupiny, která je nastavena jako výchozí pro příjem, vybrán jeden uživatel, napevno nastaven administrátorem, a nebyl by nastaven atributu `pridat_vychozimu` na "on", vybíral by se uživatel pomocí výše uvedeného prvního SQL dotazu. Hledal by se tedy aktivní uživatel výchozí skupiny, který nemusí existovat, protože hodnotu `aktivni` v tabulce `helpdesk_uzivatel` mohou uživatelé měnit jen pokud je nastavena hodnota `aktivni=1` v tabulce `helpdesk_skupina` u jejich skupiny. Protože by se žádný takový uživatel nemusel najít přiřazovaly by se zprávy jen skupině a uživateli s `id_uzivatel=0` (což je uživatel se jménem „nikdo“, kterému se dočasně přiřazovaly zprávy v případě, že zrovna neexistoval žádný aktivní uživatel skupiny). Ty by se uživateli „nikdo“ hromadily do té doby, než by se jiný uživatel této skupiny aktivoval (při aktivaci uživatele dochází k přiřazení ještě nepřijízených zpráv jeho skupin). Aktivace by musela být u skupiny povolena.

Tyto případy, ale nemusejí vždy nastat. Pokud jsou ale hodnoty v nastavení zadány špatně, mohou problémy vznikat. Je tedy dobré tyto případy ošetřit.

Výběr uživatele při pozastavení a přeposílání dotazu

Dotazy, které jsou pozastavené (mají hodnotu atributu `pozastaveno=1`) se ve skriptu `kontrola_pozastavené.php` kontrolují, zda jejich čas pozastavení již vypršel. Tento čas je uložen v logu dotazu. Vazby tabulek dotazu a logu jsou na obrázku Obrázek 3.2.

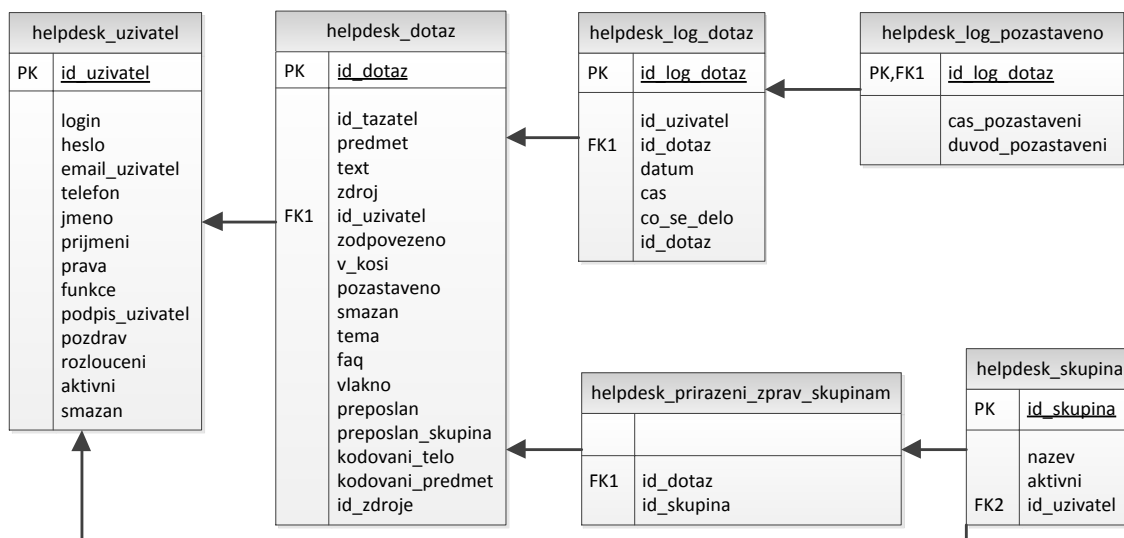
V tomto skriptu se tedy načtou všechny pozastavené dotazy a jejich časy pozastavení. Pokud čas pozastavení již vypršel, vybere se uživatel pomocí níže uvedeného dotazu a tomuto uživateli se dotaz přidělí.

```
SELECT id_uzivatel
FROM helpdesk_uzivatel
WHERE aktivni = 1
```

Po přidělení se u dotazu změní atribut `pozastaveno` na hodnotu 2. Tím se označí, že dotaz již není ve stavu pozastavení, ale je již přidělen. Uživatelé ho v systému vidí barevně odlišený. Pokud není žádný uživatel jako aktivní, dotaz se přiřadí uživateli „nikdo“ a při aktivaci jiného uživatele se vyzvedne.

Tento výběr uživatele ale neřeší přiřazení skupin. Problém zde může nastat, pokud se během pozastavení změní možnost nastavování aktivity na jinou skupinu (v tabulce `helpdesk_skupina` bude hodnota `aktivni` nastavena pro jinou skupinu) a v ní se aktivuje jiný uživatel. Tak by se zpráva přiřadila uživateli jiné skupiny, než do jaké je ona sama přiřazena. Přiřazení dotazu skupině je zobrazeno na obrázku Obrázek 3.2 a přiřazení uživatele do skupiny na obrázku Obrázek 3.1.

Je to také jen krajní možnost, kdy může nastat chyba, protože se nepředpokládá, že by se nastavení měnilo tak často, ale je to i tak nedostatek, který by bylo dobré opravit.



Obrázek 3.2 Část datového modelu, kde je zobrazeno zařazení dotazu do skupin, přiřazení uživatele dotazu a čas pozastavení

U přeposílání se dá vybrat buď přímo uživatel, který patří do jedné ze skupin přihlášeného uživatele nebo skupina. Při přeposlání dotazu skupině se vybere uživatel, který je zapsán v tabulce helpdesk_skupina. Podle výše popsaných událostí je zde uložen uživatel na pevně zadán administrátorem, uživatel, který se sám aktivoval nebo uživatel „nikdo“, pokud v systému není nikdo aktivní. SQL příkaz pro výběr je zde jednoduchý a vypadá takto:

```

SELECT id_uzivatel
FROM helpdesk_skupina
WHERE id_skupina = $skupina
  
```

Pokud se tedy dotaz přepoše jiné skupině, tak se v databázi kromě přidání záznamů do logu změní u přeposílaného dotazu v tabulce helpdesk_dotaz atribut id_uzivatel na uživatele vybraného výše uvedeným dotazem, preposlan_skupina na 1 a preposlan na 0. V tabulce helpdesk_prirazeni_zprav_skupinam se také provede změna přiřazení dotazu nové skupině. Na obrázku Obrázek 3.2 jsou zobrazeny tyto tabulky a jejich vazby.

U této akce by tedy nemělo docházet k žádnému problému a bude po úpravách, při možnosti aktivace více uživatelů najednou, třeba vyřešit jen výběr konkrétního uživatele.

4. Použité technologie

Aktuální i upravená verze systému je postavena na běžně rozšířených volně dostupných technologiích. Tato kapitola obsahuje ke každé z nich krátký popis použití či problémů spojených s touto prací.

4.1. PHP, Apache HTTP Server, MySQL, CRON

PHP a Apache HTTP Server

Je scriptovací jazyk vytvořen pro snadné vytváření dynamických webových stránek. Je vyvíjen jako procedurální jazyk. Od 5. verze obsahuje sice možnost vytváření objektů se všemi rysy objektového programování, ale jeho jádro a knihovny jsou stále procedurální, nejsou přehledně členěny do balíčků a nevyhazují výjimky. Oproti jiným jazykům jako J2EE a ASP.NET je mnohem volnější v syntaxi, dodržování struktury kódu apod. Zvyšuje to jednoduchost, ale také to může být obrovská nevýhoda, díky tomu, že celkem dost svádí programátory k vytváření nečitelného kódu. To je i případ kódu a rozvržení tohoto aktuálně používaného systému.

Jedním z dalších problémů, který se promítl do této práce je nekompatibilita některých prvků systému (viz kapitola 5.1 Roztřídění a aktualizace) a knihoven funkcí mezi starými a novými verzemi PHP.

PHP je nejvíce používáno ve spojení s webovým serverem Apache HTTP Server, který je dostupný pro velké množství platform jako jsou Linux, Mac OS X, Windows, Solaris.

MySQL a MyISAM

MySQL je databázový systém šířený hlavně pod licencí GNU GPL. Oproti jiným DBMS podporuje několik různých úložných engineů, kde každý je vhodný pro jiný účel. Engine, který je použit pro databázi upravovaného systému je MyISAM.

Nevýhodou úložiště MyISAM je nepodpora transakcí a cizích klíčů, takže nesprávnou manipulací s daty či špatně vytvořeným programem může docházet k nekonzistenci dat a odkazování na záznamy, které se v tabulce nenacházejí. MySQL standartně sice obsahuje jiný engine (InnoDB), který kromě jiných tyto vlastnosti podporuje. Jeho nevýhoda je ale rychlost, která je oproti MyISAM v některých (testovaných [6]) případech mnohonásobně pomalejší, což je nejčastější a podstatný důvod, proč je MyISAM používanější.

CRON

Je systemový nástroj v operačních systémech Unix a Linux umožňující spouštět programy v předem definovaných dobách nebo intervalech. U větších webových projektů je využíván velmi

často pro pravidelné spouštění scriptů, které mají za úkol provádět údržbu, průběžně kontrolovat nebo plnit jiné pravidelné úkoly.

4.2. AJAX

Pojem AJAX (Asynchronous Javascript And XML) označuje techniku pro výměnu dat mezi klientem a serverem na pozadí již načtené webové stránky. Tato technika využívající již existující standarty a jazyky. Umožňuje, aby části webových stránek byly změněny asynchronně bez nutnosti stahovat a překreslovat celý obsah nové stránky. AJAX je založen na kombinaci použití známých webových standartů.

Javascript

Interpretovaný scriptovací jazyk standartně integrovaný do všech moderních prohlížečů, u AJAXu se využívá k řízení celého přenosu (práci s XMLHttpRequest objektem) a následně změně DOMu již načtené stránky.

DOM

Pojem Document Object Model označuje objektovou reprezentaci XML nebo HTML dokumentu. Prvky takového dokumentu jsou v DOMu převedeny do datové struktury typu strom. Je to programové rozhraní pro přístup a manipulaci s prvky dokumentu. Všechny moderní prohlížeče již mají toto rozhraní implementované podle standartu W3C (tzv. W3C DOM). V technice AJAX se pomocí manipulace DOMu mění části dokumentů, podle dat, které byly vygenerovány na serveru.

Objekt XMLHttpRequest

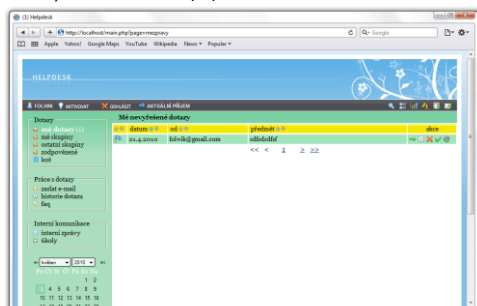
Rozhraní scriptovacího jazyku prohlížeče (dnes výhradně JavaScriptu) umožňující komunikaci přes protokoly HTTP a HTTPS přímo uvnitř webového prohlížeče. Nejčastější použití je pro odeslání požadavku na webový server, který jej zpracuje a vrátí vygenerované data zpět klientovi, které jsou následně v prohlížeči zpracována (zobrazeno na obrázku Obrázek 4.1). Data mohou být přenášena v různých formátech. Nejčastěji používané jsou tyto:

- | | |
|------------|---|
| čistý text | - data jako textový řetězec, přenášeny s MIME-typem text/plain |
| XML | - značkový metajazyk umožňující vytvářet vlastní značky a jazyky |
| JSON | - (JavaScript Object Notation) textový popis dat v lidsky-čitelném formátu, je převzat z JavaScriptu, lze v něm ukládat data reprezentována ve formě asociativního pole |
| HTML/XHTML | - úseky dokumentu, které mohou být bez dalšího zpracování vkládány přímo do již načtené webové stránky |

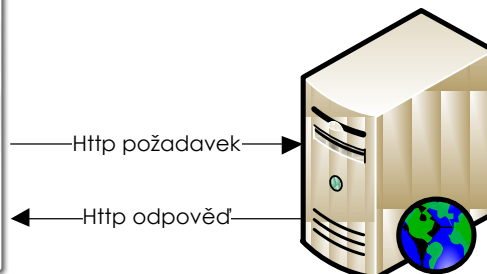
Problém je ve starších verzích prohlížeče Internet Explorer (IE 6,7), zde existuje alternativa `activexObject("Microsoft.XMLHTTP")`. Je tak pro tyto verze potřeba vytvářet jiné kódy.

Nejčastější použití AJAXu

- 1) zachycení události
- 2) vytvoření XMLHttpRequest objektu
- 3) odeslání Http požadavku



- 4) zpracování Http požadavku
- 5) odeslání Http odpovědi



- 6) zpracování vrácených dat Javascriptem
- 7) aktualizace části načtené stránky

Obrázek 4.1 Zobrazení průběhu nejčastějšího použití techniky AJAX

4.3. jQuery

Jedná se o JavaScriptový framework zaměřený především na usnadnění výběru prvků z DOMu. Používá k tomu CSS selektory s plnou podporou možností CSS1-CSS3, které jsou rozšířené ještě o několik dalších. Kromě zjednodušení selekce obsahuje také řadu dalších funkcí, které usnadňují nebo řeší často používané akce, jako jsou manipulace s DOMem, práce s CSS, událostmi, atributy objektů, komunikace pomocí AJAX, animace a další. Jeho velkými výhodami jsou velmi malá velikost (necelých 25 kB), rychlost, výborně popsané API, kompatibilita s nejčastěji používanými prohlížeči (IE 6.0+, FF 2+, Safari 3.0+, Opera 9.0+, Chrome), která řeší i rozdíly v implementaci JavaScriptu ve starších prohlížečích, což byly i hlavní důvody, proč jsem tento framework použil v rozšíření systému.

5. Návrh řešení a implementace

Po kompletní analýze systému, kde jsem se dozvěděl jak systém funguje, v kterých místech a případech má slabiny nebo chyby a jak je ve zdrojovém kódu rozdělen, přišla na řadu část, kdy se musely navrhnout přesné změny systému. Celou úpravu řešení problémů jsem rozdělil do několika bodů, které jsou popsány v následujících podkapitolách.

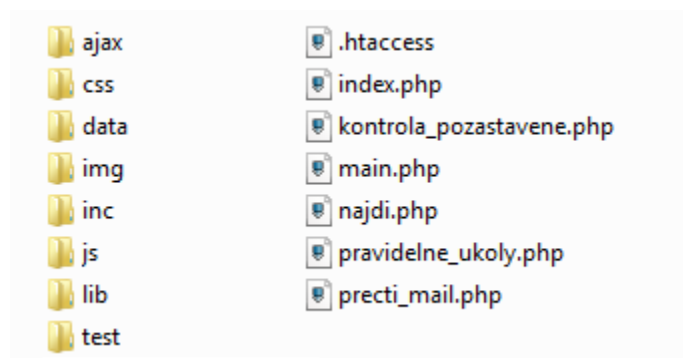
Postup úprav probíhal tak, že se každý problém řešil postupně. Vždy jsem nejprve zvážil, jak daný problém vyřešit, rozepsal jsem konkrétní úpravy, ty naimplementoval a následně hned testoval. Testování každého bodu úpravy zvláště hned po implementaci jsem zvolil proto, protože jsem měl čerstvě v paměti všechny prováděné úpravy. Odstranění případných chyb či nalezení nechtěných situací tak bylo lehčí, než kdyby se provádělo jen testování až po naimplementování všech úprav. Celkové testování práce, ale bylo také prováděno a je popsáno v kapitole 6. Testování.

5.1. Roztřídění a aktualizace

Změna adresářové struktury a názvů souborů

V původní verzi systému byly všechny skripty umístěny v hlavním adresáři. Bylo tedy složité rozpoznat a zapamatovat si, které soubory jsou určeny ke spuštění, které obsahují jen kusy kódu a jsou určeny pro vložení do jiných souborů, a které obsahuje jen funkce či reprezentují nějaké třídy.

Proto jsem se rozhodl, že nejprve všechny soubory roztřídím, přesunu a přejmenuju tak, aby bylo z názvu a umístění na první pohled jasné o jaký soubor se jedná a co se v něm zhruba nachází. Nové uspořádání by mělo být co nejpřehlednější a mělo by rozdělovat všechny typy souborů. Zvolil jsem strukturu, která je k vidění na obrázku Obrázek 5.1.



Obrázek 5.1 Nová adresářová struktura

V hlavním adresáři jsem ponechal jen soubory, které byly spuštěny buď pomocí odkazů v prohlížeči nebo byly nastaveny jako automatické úlohy v CRONu. Ostatní PHP soubory jsou rozděleny do složek `lib`, `inc` a `test`. V `lib` se nachází soubory reprezentující třídy nebo nějaké jiné ucelené části kódu. Ve složce `inc` jsou pak soubory, které se vkládají do jiných scriptů (pomocí `include` nebo `require`). Pro rozlišení těchto souborů jsem přidal předponu před původní název.

<code>page.*.php</code>	- takto jsou pojmenovány soubory reprezentující stránku v systému
<code>func.*.php</code>	- soubory s předponou <code>func</code> obsahují jen funkce
<code>inc.*.php</code>	- tyto soubory obsahují často používané kusy kódu

Přesunutím souborů do složek `inc` a `lib` se oddělily scripty, které by neměly být samostatně spuštěny. Aby se uživatelům zamezil přístup k těmto souborům přes `url`, přidal jsem do `.htaccess` souboru, který lokálně nastavuje parametry Apache webserveru pro danou složku, prepisovací pravidlo, jenž pro jakékoli umístění ve složkách `lib` a `inc` vrátí odpověď „403 Přístup odepřen“.

V adresáři `test` jsou všechny scripty určené jen pro testování, které nejsou potřeba pro zbytek systému. Další rozmístění jsou pak tyto:

<code>ajax</code>	- do této složce nejsou přemístěny žádné z původních souborů, ale budou zde umístěny scripty, které budou vyřizovat AJAX požadavky, popsány níže
<code>css</code>	- soubory kaskádových stylů
<code>data</code>	- složka pro ukládání příloh zaslaných v příchozích emailech nebo odpovědích
<code>img</code>	- obsahuje všechny obrázky
<code>js</code>	- v této složce jsou soubory s externími JavaScripty

Aktualizace scriptů

Kromě nutné změny cest ve všech scriptech díky přesunutí a přejmenování původních souborů bylo potřeba nahradit části kódu, které už v nejnovějších verzích PHP nejsou funkční. Jedná se hlavně o pole `$HTTP_GET_VARS` a `$HTTP_GET_POST` obsahující hodnoty proměnných zadaných v HTTP požadavku, které jsou od verze PHP 4.1.0 nahrazeny za pole `$_GET` a `$_POST`. Inicializace polí s původními identifikátory lze sice nastavit pomocí direktivy `register_long_arrays`, ale oficiálně jsou tyto pole označena jako zavržená [4].

Dále jsem také nahradil zkrácené zápisy otevíracích značek z důvodů, které jsou popsány v kapitole 3.3 Rozbor implementace stávajícího řešení, a nahradil javascriptový framework jQuery za nejnovější verzi. Ten byl použit u modulu Open Flash Chart, který je do systému vložen pro vykresluje statistik. Framework jQuery jsem také použil pro prvky uživatelském rozhraní, které jsou popsány níže.

Aby bylo jednoduché systém nainstalovat nebo přesunout, vytvořil jsem soubor `config.php` umístěný ve složce `inc`. V něm jsou přesunuty a přidány všechny potřebné

parametry (například přístupové údaje k databázi, cesty k adresářům apod.), zadané jako hodnoty v poli `$CONFIG` nebo jako konstanty. Ty jsou teď přehledně na jednom místě a nemusí se dohledávat v různých souborech, kde byly předtím napevno. Stačí pak v ostatních scriptech tento soubor na začátku načíst.

5.2. Aktivace více uživatelů

Jelikož stávající systém je již nějakou dobu zaběhnutý a obsahuje obrovské množství dat, které by bylo dobré zachovat i po nasazení upraveného systému, hledal jsem takové řešení, u kterého by nebylo nutné zasahovat do stávajících dat a budoucí aktualizace systému by byla co nejjednodušší.

Vycházel jsem tedy z vytvořených tabulek `helpdesk_uzivatel` a `helpdesk_skupina` a jejich atributů. Pro indikaci, zda je uživatel aktivní či ne jsem v prvně jmenované tabulce ponechal atribut `aktivni`.

Musel se tedy změnit proces, kdy se uživatel sám aktivuje. V předchozí verzi se vždy deaktivovali všichni ostatní uživatelé, nový se aktivoval a `id_uzivatele` se v tabulce `helpdesk_skupina` u příslušného záznamu změnilo na `id` nově aktivovaného. Pro možnost aktivace více uživatelů tedy postačuje u každého uživatele jen měnit atribut `aktivni`. Při aktivaci a deaktivaci se nebude měnit tento parametr u ostatních záznamů v tabulce `helpdesk_uzivatel`, ani se nebudou přepisovat záznamy v tabulce `id_skupina`. Tím je tedy zajištěna možnost mít více aktivních uživatelů.

Pro některé skupiny je ale dobré mít možnost, že se její členové nebudou aktivovat či deaktivovat sami, ale bude zvolen jeden uživatel, kterému budou přednostně přiděleny všechny dotazy zaslané skupině. Pro zvolení takového uživatele je využit bývalý atribut `id_uzivatel` v tabulce `helpdesk_skupina`. Tohoto uživatele může vybrat administrátor. Aby se uživatelům zařazeným do nějaké skupiny zakázala možnost se v této skupině aktivovat, je využit již vytvořený atribut `aktivni` v tabulce `helpdesk_skupina`, který tuto možnost pro každý záznam umožňuje.

Část systému, ve které se nastavuje přiřazení uživatelů do skupin, a kde se vybírá způsob přiřazení skupiny je zobrazena na obrázku Obrázek 7.1 a popsána v kapitole 7.2 Nastavení skupin.

5.3. Výběr uživatele pro přidělení dotazu

Jelikož se změnil význam ukládaných dat v attributech `aktivni` a `id_uzivatel` tabulky `helpdesk_skupina`, byla potřeba změnit výběr aktivního uživatele při všech operacích, kdy se dotaz přiřazuje skupině.

5.3.1. Návrhy řešení

V předchozím řešení se až na případ, kdy se pozastavené dotazy přidělují po vypršení času pozastavení, vybíral uživatel podle `id_uzivatel` obsaženého v tabulce `helpdesk_uzivatel`. Nyní, kdy se v tomto atributu nenachází identifikátor aktivního uživatele, ale jen id uživatele, v případě, že byl některý vybrán napevno, nemůže se tento výběr použít. Musí se vždy rozeznávat zda je pro skupinu uživatel zadán pevně nebo se má vybírat z aktivních uživatelů skupiny. Tato informace se zjistí podle toho, jestli atributu `id_uzivatel` obsahuje identifikátor uživatele, nebo jestli má hodnotu 0. V případě, kdy obsahuje identifikátor, je uživatel rovnou vybrán, v opačném případě to znamená, že se uživatel vybere z aktivních patřících do této skupiny.

Celkový proces výběru uživatele tedy zahrnuje tyto kroky:

1. Kontrola, zda je pro danou skupinu nastaven výchozí uživatel.
2. Pokud není, zjistí se, kteří uživatelé ze skupiny jsou aktivní.
3. Výběr jednoho z aktivních uživatelů skupiny.
4. Pokud takový neexistuje, nepřirázovat uživateli, ale ponechat skupině.

U třetího kroku je nutné zajistit, aby v případě, kdy bude ve skupině více aktivních uživatelů, systém nevybíral vždy jednoho. Jedním řešením tohoto problému by bylo náhodně z těchto uživatelů vybírat. Mohlo by se ale stát, že by byli někteří uživatelé vybíráni mnohem častěji než ostatní a přiřazování by tak nebylo moc spravedlivé. Proto jsem tento způsob zamítnul a hledal jiné řešení.

Dalším možným řešením, u kterého by se použily jen dosavadní data a nedocházelo by k ukládání redundantních informací (viz [1]), bylo vybrat uživatele podle posledního záznamu v logu. Záznamy v tabulce `helpdesk_log_dotaz` totiž obsahují datum, čas, id uživatele a typ akce. Celá tabulka s atributy je zobrazena výše na obrázku Obrázek 3.2. Stačilo by tak ke každému vybranému uživateli dohledat záznam se shodným atributem `id_uzivatele`, atributem `co_se_del='prijem'` a nejvyšším datem a časem. Z takto získaných dat přiřazených uživateli z kroku 2 by se vybral jeden uživatel, který by měl nejnižší čas a datum, tzn. vybral by ten, který přijal dotaz jako poslední. Takto by bylo zaručeno, že dotazy budou spravedlivě rozdělovány mezi všechny aktivní uživatele, protože při každém přidělení vznikne také nový záznam v logu s aktuálním datem a časem.

Toto řešení sice provádí spravedlivé rozdělování a využívá dostupná data, ale také obsahuje problémy. První problém je, že se při výběru dat z tabulky `helpdesk_log_dotaz` musí pro každého uživatele selektovat jen specifické záznamy s potřebnými hodnotami atributů. Tato tabulka obsahuje pro každý dotaz několik vlastních záznamů a při existenci obrovského množství dotazů by se vyhledáváním a tříděním databáze zbytečně zatěžovala. Navíc datum a čas jsou zde uloženy v samostatných attributech, takže by se pro každého uživatele musely buď nejdříve vybrat záznamy s nejvyšším datem a následně v nich záznam s nejvyšším časem nebo tyto hodnoty spojit a až následně setřídít a vybrat záznam s nejvyšší hodnotou, což by ani v jednom případě nebylo moc efektivní.

Druhý problém může nastávat u vyzvedávání dotazů z emailové schránky, což je případ, kdy se tento výběr uživatele bude uplatňovat nejčastěji. Spočívá v tom, že dotazy zaslané na schránku nejsou zpracovávány ihned, ale vždy pravidelně v intervalech, jejichž délka je nastavena v CRONu. Hodnota času ukládaná do logu je ve formátu HH:MM:SS, takže nejmenší možná odchylka je sekunda. V případě, že se ve schránce nahromadí několik emailů a bude aktivních jen malý počet uživatelů skupiny, může se velice pravděpodobně stát, že délka zpracování několika dotazů bude kratší než jedna sekunda a většina dotazů se přidělí jen jednomu uživateli. Důvod je, že čas přidělení v logu bude u těchto dotazů stejný. Takže by například u dvou uživatelů (A, B) nastala situace, že 1. dotaz by byl přidělen uživateli A, druhý dotaz uživateli B. Oba uživatelé by měli v tuto dobu stejné poslední časy v logu a u dalších dotazů zpracovávaných ve stejnou sekundu by pak neexistoval požadovaný rozdíl. Jelikož zde není zavedena žádná náhodná hodnoty vybíral by se vždy prvně nalezený uživatel.

Tento problém by se dal vyřešit například ukládáním identifikátorů uživatelů, kterým již byl přidělen dotaz, v průběhu zpracování scriptu do pole a u každé zprávy kontrolovat, zda je již vybraný uživatel v poli. V případě, že by již byli všichni uživatelé v poli, vyprázdnilo by se a začalo od znova. Toto řešení není ale moc efektivní a také hlavně neřeší prvně jmenovaný problém a spíše systém ještě zatěžuje, takže jsem tento způsob výběru zavrhl.

5.3.2. Výsledné řešení

Pro výsledné řešení jsem použil minulou myšlenku, která rozdělovala dotazy na základě času posledního příjmu uživatele. Tento čas se ale nebude získávat ze záznamů v logu, ale bude přímo u každého uživatele. Jedná se o jednoduchou formu denormalizace, kdy budou ukládány redundantní informace z důvodů zvýšení efektivity (popsáno v [2]). V tabulce `helpdesk_uzivatel` jsem proto přidal sloupec `posledni_cas`, do kterého se aktuální čas bude ukládat. Protože v MySQL u datového typu `TIME` nelze uložit menší časovou délku než sekunda (viz [5]), použil jsem typ `CHAR` s délkou 19 znaků. Do tohoto atributu bude uložena hodnota, která se získá z PHP funkce `microtime()`. Ta vrací UNIXové časové razítko typu string ve tvaru „`microsec sec`“. Tuto hodnotu ale před zápisem do databáze převedu na tvar „`sec.microsec`“, která má délku právě 19 znaků (8 znaků sekundy, tečka, 8 znaků mikrosekundy). Takto bude u každého uživatele uložena hodnota, která se dá jednoduše porovnávat bez použití dalších dotazů či spojování tabulek. Není tak nutné procházet tabulky logů, které mají několikanásobně více záznamů než tabulka uživatelů. Jediná nutnost je aktualizovat tento čas po každém přiřazení dotazu uživateli.

SQL dotaz pro výběr aktivního uživatele tedy vypadá takto:

```
SELECT u.id_uzivatel, u.posledni_cas
FROM helpdesk_uzivatel u JOIN helpdesk_priprazeni_skupin ps ON
    u.id_uzivatel = ps.id_uzivatel
WHERE u.aktivni = 1 AND
```

```
u.smazan = 0 AND
ps.id_skupina = $skupina
ORDER BY u.posledni_cas ASC
LIMIT 1
```

PHP script pro aktualizaci času uživatele vypadá takto:

```
$mt = explode(" ", microtime());
$mt = $mt[1].substr($mt[0], 1);
mysql_query("UPDATE helpdesk_uzivatel
            SET posledni_cas = '$mt'
            WHERE id_uzivatel = $priradit");
```

Výběr uživatele tímto postupem, kdy se pro každé přiřazení nejdříve zkontroluje zda je zadán přednostní uživatel pro skupinu, v případě že ne, vybere se jeden z aktivních uživatelů v rámci skupiny, je možné použít ve všech případech – výběr ze schránky, přeposílání, přiřazení po skončení doby pozastavení. Oproti původnímu přiřazování, kdy mohla vznikat chyba například při pozastavování, kde se nehlídalo přiřazení uživatelů do skupin, či přenastavení chování skupin během doby pozastavení dotazu, je tento výběr uživatele ošetřen.

5.3.3. Přiřazení dotazu při neexistence žádného aktivního uživatele

Může se stát, že postupem popsáním v předchozí podkapitole nebude vybrán žádný uživatel – tzn., že nebude nastaven výchozí uživatel pro skupinu a ani nebude ze skupiny nikdo aktivní. Takto se v tabulce helpdesk_dotaz vytvoří záznam s `id_uzivatel=0`. Během vyzvedávání se také ale vytvoří záznam v tabulce helpdesk_prirazeni_zprav_skupinam, kde se propojí dotaz se skupinou, do které byl zaslán.

V neupravené verzi systému byly tyto dotazy přiřazeny uživateli jen v momentě, kdy se aktivoval některý z uživatelů skupiny. Pro skupiny, ve kterých byly nastaveni jen výchozí uživatelé tak neexistoval způsob tyto dotazy někomu přiřadit, protože výchozí uživatelé nemají možnost se aktivovat. Tato situace mohla nastat v případě, že by po nějakou dobu pro skupinu nebyl vybrán jak výchozí uživatel, tak aktivování žádní uživatelé skupiny. Během této doby by se nahromadily dotazy přiřazené jen skupině a uživateli s `id_uzivatele=0`. Pokud by pak byl administrátorem vybrán výchozí uživatel, nedošlo by v rámci skupiny k aktivaci žádného uživatele (z důvodu nemožnosti se aktivovat a deaktivovat v rámci skupiny, pro kterou je vybrán výchozí uživatel) a dotazy této skupiny by zůstaly nepřijazeny uživateli.

Tento proces jsem proto rozdělil tak, aby pro skupiny, kde je možné se aktivovat, zůstala prováděna kontrola po aktivaci uživatele, ale pro skupiny, ve kterých je vybrán výchozí uživatel probíhala kontrola průběžně. Tato kontrola je prováděna před načtením úvodní stránky a stránky „mé dotazy“, což je stránka, na které uživatel má k dispozici a kontroluje jemu přiřazené dotazy.

Na ostatních stránkách tato kontrola nemá smysl, protože dotazy uživatele se na nich neobjevují.

Tímto se eliminují případy, které mohly nastávat v minulé verzi a je ošetřen i případ, kdy by došlo k přenastavení skupiny, ve které se objevují nepřřazené dotazy.

5.4. Vyzvedávání dotazů

Změna, která umožňuje aktivaci více uživatelů vyžadovala úpravy v procesu vyzvedávání dotazů z emailové schránky. Hlavní změna zde byla výběr uživatele, kterému se dotaz přiřadí. V minulé verzi existoval v tabulce `helpdesk_nastaveni` atribut `prirad_vychozimu`, který určoval, zda se bude vyzvednutý dotaz přiřazovat aktivnímu uživateli nebo tomu, který je nastaven v `helpdesk_skupina`. Mohla zde vznikat chyba, ta je popsána výše v kapitole 3.3.2 Aktivace a výběr aktivního uživatele. Nastávaly tam 2 případy, v jednom nezáviselo na tom, jak je tato hodnota nastavena, a v druhém musela být zaškrtnuta (nabývat hodnoty „on“), aby docházelo k přiřazení uživatelům.

Tento atribut byl tedy téměř nepotřebný, proto jsem ho z tabulky odstranit. Funkčnost, která jím měla být ovlivněna se dá měnit díky vytvořené úpravě, která řídí způsob výběru uživatele pro přiřazení dotazu. V nastavení skupiny je již totiž možnost vybrat konkrétního uživatele nebo umožnit uživatelům skupiny se samostatně aktivovat. Při použití výběru uživatele, který je popsán v přecházející kapitole není atribut `prirad_vychozimu` vůbec potřebný. Odstraněním parametru a převedením funkčnosti do nastavení skupiny se vyřešily bývalé komplikace a toto nastavení se v uživatelském rozhraní přesunulo na jedno místo.

Popis procesu vyzvedávání emailu v bodech

O vyzvedávání se stará script `precti_mail.php`, který je CRONem spouštěn v pravidelných intervalech. Proces probíhá v těchto krocích.

1. Načtení nastavení emailových všech (`SELECT * FROM helpdesk_nastaveni`)
2. Pro každou schránku vybrány uložené emaily přes POP3 protokol
3. Zpracování každého emailu
 1. Zjištění jestli pro danou skupinu je nastaven výchozí uživatel, pokud ne, vybrání jednoho z aktivního uživatele skupiny. Pokud není nikdo vybrán, bude použito `id_uzivatel=0`.
 2. Připojení do extérní databáze, ve které jsou uloženy data klientů firmy a zjištění, zda emailová adresa patří některému z klientů. Podle této informace se vybere priorita tazatele.
 3. Pokud již v této databázi v tabulce `helpdesk_tazatel` neexistuje záznam s touto emailovou adresou, vytvoří se nový záznam tazatele, obsahující mimo jiné prioritu zjištěnou v minulém bodě.

4. Rozeznání z předmětu emailu, zda se jedná o odpověď na již zodpovězený dotaz. V případě, že ano, uložení identifikátoru vlákna.
 5. Vložení záznamu do tabulky `helpdesk_dotaz`, obsahující kromě obsahu emailu, identifikátor tazatele a zdrojové schránky, číslo vlákna. V případě úspěšného vložení se provedou kroky 6 - 11.
 6. Odeslání oznamovacího emailu o přijetí dotazu tazateli.
 7. Aktualizace posledního času uživatele, který byl vybrán pro přiřazení dotazu (jen pokud byl nějaký vybrán).
 8. Vytvoření záznamu v tabulce `helpdesk_priřazeni_zprav_skupinam`, jež propojuje dotaz mezi skupinou. Tato informace se využívá pro pozdější přidělení v případě, kdy nebyl vybrán žádný uživatel.
 9. Přidání záznamů do tabulky `helpdesk_log_dotaz` s informacemi o přijetí dotazu a případně o přiřazení uživateli.
 10. Pokud email má přílohy, uloží se do přednastavené umístění na disku a vytvoří se záznamy v `helpdesk_priloha`.
 11. Smazání zpracovaného emailu ze schránky.
4. Uzavření spojení s emailovou schránkou.

5.5. Pozastavení a přeposílání dotazů




Pro výběr uživatele u pozastavení i přeposílání dotazů je použit stejný algoritmus jako u přiřazování dotazů při vyzvedávání zpráv ze schránky. Kroky procesu přeposílání zůstaly stejné jako v neupravené verzi, bylo zde ale nahrazeno vybírání uživatele, kde se místo pouhého načtení `id_uzivatele` z tabulky `helpdesk_skupina` použil výše popsany výběr. Přibyly také nutná aktualizace posledního času u vybraného uživatele.

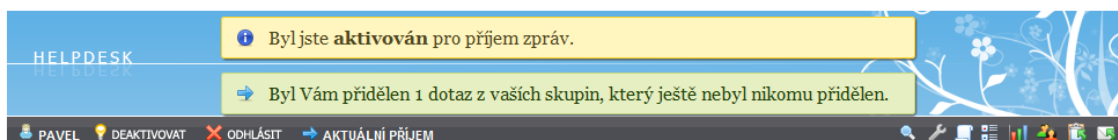
U přeposílání i pozastavení funguje také přiřazení jen skupině v případě, že v danou chvíli neexistoval žádný uživatel, kterému by bylo možno dotaz přiřadit. Tyto dotazy se pak stejně jako u vyzvedávání přidělí, když se v systému takový uživatel objeví. Tato úprava zajišťuje, že se přeposlaný i pozastavený dotaz v jakémkoliv případě přidělí uživateli a nezůstane nepřidělen.

Upravený script (`kontrola_pozastavene.php`), který kontroluje pozastavené soubory, zda jim již vypršel čas pozastavení, nyní obsahuje výběr uživatele, který bere do úvahy i přiřazení skupin. Řeší tak problém popsany v kapitole 3.3.2 Aktivace a výběr aktivního uživatele, který mohl nastávat v minulé verzi. Zvýší se tak bezpečnost přidělení dotaz, protože byla ošetřena možnost, kdy mohlo dojít k přidělení dotazu uživateli jiné skupiny, a také se zlepšil udržení konzistence. Nemůže se tak totiž stát, že by byl dotaz přidělen jedné skupině a následně, kvůli bývalé chybě, přidělen uživateli jiné skupiny.

5.6. Úpravy uživatelského rozhraní

Jelikož v původním systému neexistovala žádná oznámení o provedených akcích, neměli uživatelé ponětí, zda se jimi iniciovaný proces provedl v pořádku, s jakými výsledky, či nastala chyba. Bylo tedy vhodné tyto informace uživatelům nějak předat. Navrhl jsem tedy výpis krátkých zpráv umístěných na dobře viditelném místě v horní části obrazovky. Tyto zprávy jsou zde zobrazeny po dobu 5ti sekund, pak začnou postupně ze zhora mizet, aby nepřekážely v další práci se systémem. Pokud by uživatel chtěl skrýt zobrazenou zprávu dříve, může to provést kliknutím na ní. Pro lepší rozeznání typu zprávy existují 3 barevné verze:

-  Žlutá (notice) - oznámení o provedené akci iniciované uživatelem
-  Zelená (message) - oznámení o přiřazení dotazu
-  Červená (error) - chybová zpráva



Obrázek 5.2 Ukázka dvou zpráv oznamujících provedené akce (žlutá - oznámení, zelená - přiřazení dotazu)

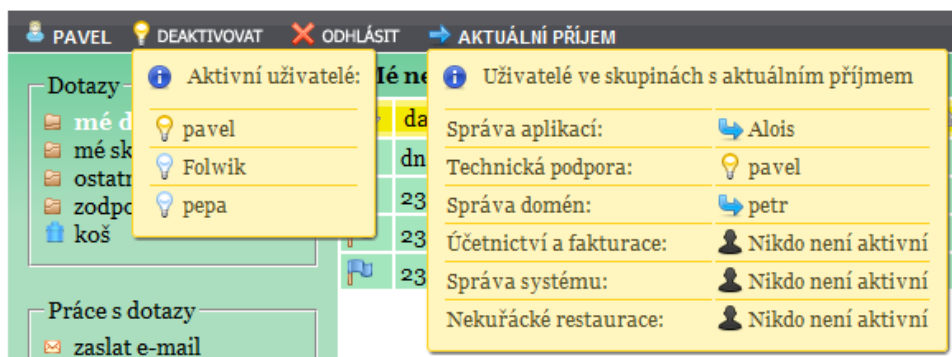
Zprávy jsou vytvářeny v PHP a ukládány jako session proměnné. Vypisovány jsou až při zobrazení uživatelského rozhraní. Díky ukládání do sessions je možné vytvořit takovou zprávu ve scriptu, který nevykresluje uživatelské rozhraní nebo přesměrovává na jiný script. Zobrazování, animace a interaktivní prvky jsou vytvořeny v JavaScriptu s využitím jQuery. Tento způsob oznamování se dá jednoduše použít i při dalším rozšiřování systému. Pro vytvoření zprávy stačí v kódu jen přiřadit do sessions další prvek. Toto přiřazení vypadá takto:

```
$_SESSION['INFOBOXES'][ ] =  
    array("class" => "message",  
          "text" => "Bylo Vám přiděleno $pocet dotazů z vašich  
                    skupin, které ještě nebyly nikomu přiděleny.");
```

Aktuální informace o uživateli

V některých případech při přeposílání dotazu skupině může chtít uživatel odesílající dotaz vědět, kdo přesně odeslaný dotaz vyzvedne. Tuto informaci není možné vždy zjistit z nastavení skupin (zobrazeno na obrázku Obrázek 7.1). Vytvořil jsem proto informační okno, ve kterém jsou pro každou skupinu vypsáni uživatelé, kteří v daný moment ve skupině přebírají zprávy. Například u skupiny, která má nastaveno, že se její uživatelé mohou aktivovat, je zde vypsán uživatel s nejnižším posledním časem. Výběr uživatele je zde stejný jako výběr popsáný v předchozích kapitolách, tím je zaručena správnost výsledků. Na obrázku Obrázek 5.3 je

vpravo toto okno vidět, zobrazí se po kliknutí na tlačítko aktuální příjem. Je zde rozeznáno, zda je uživatel vybrán z aktivních (ikona žárovky) či je nastaven jako výchozí (modrá šipka). Po najetí nad název uživatele je zobrazen podrobnější popis, vysvětlující způsob přiřazení dotazu.



Obrázek 5.3 Okna zobrazující aktivitu uživatelů a informace o aktuálním příjmu skupin


Další okno, které poskytuje informace o tom, kteří uživatelé jsou v daném momentě aktivováni je dostupné po najetí nad tlačítko Aktivovat / Deaktivovat (zobrazeno na obrázku Obrázek 5.3). Kvůli větší izolovanosti jsou zde obsaženi jen uživatelé patřící do stejných skupin jako přihlášený uživatel.

Obě dvě funkce jsou implementovány pomocí technologie AJAX, informace jsou tak vždy aktuální, protože se vyhodnocují právě v momentě zachycení uživatelské akce. Díky těmto funkcím získávají uživatelé větší informovanost o aktuálním dění, což vede k zlepšení použitelnosti systému.


Zpřehlednění systému

Mezi další úpravy uživatelského rozhraní patří zpřehlednění některých navigačních a informačních prvků. Jedny z nich jsou nadpisy přidávané na každou stránku, včetně jiných verzí pro každou uživatelskou roli. Ty se v předešlé verzi systému neobjevovaly nikde, což mohlo vést k dezorientaci uživatelů. Nyní tento krátký titulek napomáhá k lepší orientaci. Další úprava přispívající k zpřehlednění je změna vzhledu odkazů pro navigaci podstránky. Ty jsem z čistě textové verze změnil na vzhled tlačítek, u kterých je rozlišeno, zda se jedná o aktuálně zobrazenou stránku, či ne. V systému se také nachází kalendář (umístěný v postranním panelu), pomocí kterého lze filtrovat zobrazení dotazů jen pro vybrané datum. Tato funkce ale fungovala jen pro několik málo stránek. Rozšířil jsem ji proto i do všech ostatních, které vypisují seznam dotazů, zpráv nebo úkolů. Velká nevýhoda byla, že v uživatelském rozhraní neexistovalo žádné oznámení o tom, že jsou data filtrována jen pro určité datum a ani nebylo vždy jasné, zda filtraci daná stránka podporuje. Proto jsem tuto informaci na všechny stránky přidal. Tyto úpravy nejsou sice nijak rozsáhlé či výrazně viditelné, ale zpřehledňují a sjednocují jednotlivé prvky systému, což zvyšuje pohodlí uživatelů a použitelnost systému.

Rozdíly mezi předchozí a upravenou verzí stránky jsou zobrazeny na obrázcích Obrázek 5.4 a Obrázek 5.5.

		
<p>přijaté odeslané napsat novou</p>		
datum ↕	od ↕	předmět ↕
dnes - 00:40	pavel	testovací interní zpráva

Obrázek 5.4 Navigace stránky v původní verzi

DHLÁSIT → AKTUÁLNÍ PŘÍJEM 		
Přijaté interní zprávy		
<input type="button" value="Přijaté"/> <input type="button" value="Odeslané"/> <input type="button" value="Napsat novou zprávu"/>		
Zobrazeny jen zprávy z 5. 5. 2010		
datum ↕	od ↕	předmět ↕
dnes - 00:39	pavel	testovací imterní zpráva

Obrázek 5.5 Upravená navigace stránky s novými prvky

6. Testování

Po dokončení všech úprav bylo nutné celý systém otestovat. Tento test zahrnoval odzkoušení možných případů, které v systému mají nastávat. Testování každé samostatné úpravy probíhalo již během její implementace a vzniklé chyby byly ihned odstraněny. Konečné testy zahrnovaly hlavně testování vyzvedávání, přeposílání a pozastavení dotazů. Kontroloval se stav databáze, vliv úprav na původní části systému a správnost zobrazení v uživatelském rozhraní.

Scripty, které se mají ve výsledném systému spouštět v intervalech pomocí CRONu byly při testech spouštěny ručně. Ruční spouštění bylo potřeba z důvodu, aby se požadovaný proces neprovedl dříve než byla vytvořena modelová situace. Například testování vyzvedávání zpráv bylo prováděno tak, že se nejprve nastavil způsob příjmu dotazů ve skupině, aktivita jejich uživatelů, dále se do emailové schránky poslal přesný počet emailů, byl spuštěn script `precti_mail.php` a následně se kontroloval očekávaný výsledek.

V systému bylo v průběhu celkového testování nalezeno pár drobných chyb. Některé z nich se vyskytovaly i v původní verzi. Jedna z nich byla vygenerování chybného SQL dotazu pro výpis zpráv ostatních skupin. Objevovala se v případě, kdy žádná zpráva patří jině skupině než té, do které patří přihlášený uživatel, neexistovala. V systému naplněném daty se neprojevovala. Byla ale jako všechny nalezené chyby, ať už původní či nově vzniklé, opravena.

Chyba ve výpisu nepřřazených dotazů

Největší odhalenou chybou byla chyba ve výpisu nepřřazených a pozastavených dotazů. Ty měly být mimo jiných zobrazeny na stránkách „mé skupiny“ (dotazy přiřazené skupině, ale ne přihlášenému uživateli), „ostatní skupiny“ (dotazy přidělené ostatním skupinám) a „detail dotazu“. Důvod byl smazání uživatele „nikdo“ (záznamu z tabulky `helpdesk_uzivatel` s `id_uzivatel=0`). SQL dotazy, které vybíraly potřebná data z databáze spojovaly záznamy z tabulky `helpdesk_dotaz` se záznamy z tabulky `helpdesk_uzivatel`. Jelikož pro nepřřazené dotazy, které měly hodnotu atributu `id_uzivatel=0` neexistoval v tabulce s uživateli žádný takový záznam, nebyly do výsledné relace zahrnuty. Řešením bylo použití vnějšího spojení (`LEFT OUTER JOIN`) mezi tabulkami `helpdesk_dotaz` a `helpdesk_uzivatel`, jenž do výsledků zahrnovalo i nepřřazené dotazy.

7. Uživatelská a programátorská dokumentace

V této kapitole je popsáno jak nainstalovat a spustit upravený systém. Detailně je popsána stránka s nastavením skupin, což je stránka na které se projeví změny v ovládání vzniklé úpravami vnitřních procesů. Dále je zde také popsáno, které soubory byly konkrétně změny, jak byly přepracovány a kde se nachází popis každé části daných souborů.

7.1. Instalace systému

Pro nasazení tohoto systému jsou vytvořeny 2 postupy instalace. První je čistá instalace, která vytvoří prázdný systém připravený k novému použití. Druhý postup je pro aktualizaci stávajícího systému.

Čistá instalace

1. Vytvoření MySQL databáze a emailové schránky přístupné přes protokol POP3.
2. Pro definici tabulek a vložení potřebných záznamů je vytvořen soubor `create.sql` umístěný ve složce `code_sql`. Na konci tohoto souboru jsou okomentované 2 SQL příkazy, které je nutno vyplnit. Jeden je pro vytvoření záznamu s informacemi o zdrojové emailové schránce. Druhý je pro vytvoření uživatele s právem administrátora. Oba příkazy je možno vložit několikrát (pro zaevidování více zdrojových schránek nebo více administrátorů). Po změně je nutné script nad vytvořenou databází spustit.
3. Zkopírování obsahu složky `code_web` na zvolené umístění webového serveru. Mělo by se jednat o umístění dostupné přímo z domény nebo subdomény bez dalšího zanořování.
4. Změna parametrů pro připojení k databázi v souboru `inc/config.php`. Zde je vytvořeno dvourozměrné asociativní pole `$CONFIG` obsahující parametry připojení k databázovému serveru a výběru databáze. Zadávají se zde 2 spojení. Jedno je `db_interni` – připojení na databázi vytvořenou v prvním bodě. Druhé je `db_externi` – připojení na databázi obsahující data o klientech firmy (popsánov kapitole 5.4 Vyzvedávání dotazů).
5. Vytvoření záznamů v CRONu pro skripty `kontrola_pozastavene.php`, `pravidelne_ukoly.php` a `precti_mail.php`. Časový interval by měl být zvolen v závislosti na předpokládaném počtu zasláných emailů v daném intervalu.

Po dokončení těchto bodů budou v systému vytvořeni uživatelé a schránky, které byly zapsány v bodě 2. Bude existovat jedna skupina s názvem „Technická podpora“, do které budou směřovány emaily ze všech vytvořených zdrojových schránek. Všechny ostatní potřebná

nastavení jako jsou vytváření uživatel, přiřazení do skupiny, výběr typu přiřazování dotazů je dostupné přímo v systému po přihlášení pod administrátorským účtem.

Aktualizace minulé verze

1. První krok je smazání původního obsahu složky, kde se systém nachází. Nutné je ale ponechat složky `prilohy` a `priloha_odpoved`.
2. Další krok je zkopírování obsahu složky `code_web` na umístění původního systému.
3. Dále je nutné přesunout složky `prilohy` a `priloha_odpoved`, které se ponechaly v prvním kroku, do složky `data`. Tyto složky se tam již nacházejí, ale jsou prázdné, takže se mohou přepsat.
4. Stejně jako u čisté instalace v kroku 4 je potřeba změnit parametry pro připojení k databázím.
5. Poslední krok je nad databází spustit SQL script `alter.sql` umístěný ve složce `code_sql`.

Po této úpravě bude systém aktualizován na upravenou verzi. Všichni uživatelé budou nastavení jako neaktivní. U žádné skupiny nebude vybrán výchozí uživatel a bude nastaven tak, že se její uživatelé nemohou sami aktivovat. Administrátor pak bude muset nastavit skupiny znova. Toto vynulování bylo potřeba z důvodu změny významu některých atributů.

7.2. Nastavení skupin

Na stránku, kde se nastavuje přiřazení uživatelů do skupin a způsob přidělování zpráv v rámci skupiny se lze dostat po kliknutí na ikonu dvou lidí umístěnou v pravé části okna a po stisknutí tlačítka „Skupiny“, které se následně objeví. Výsek této stránky je zobrazen také níže na obrázku Obrázek 7.1.

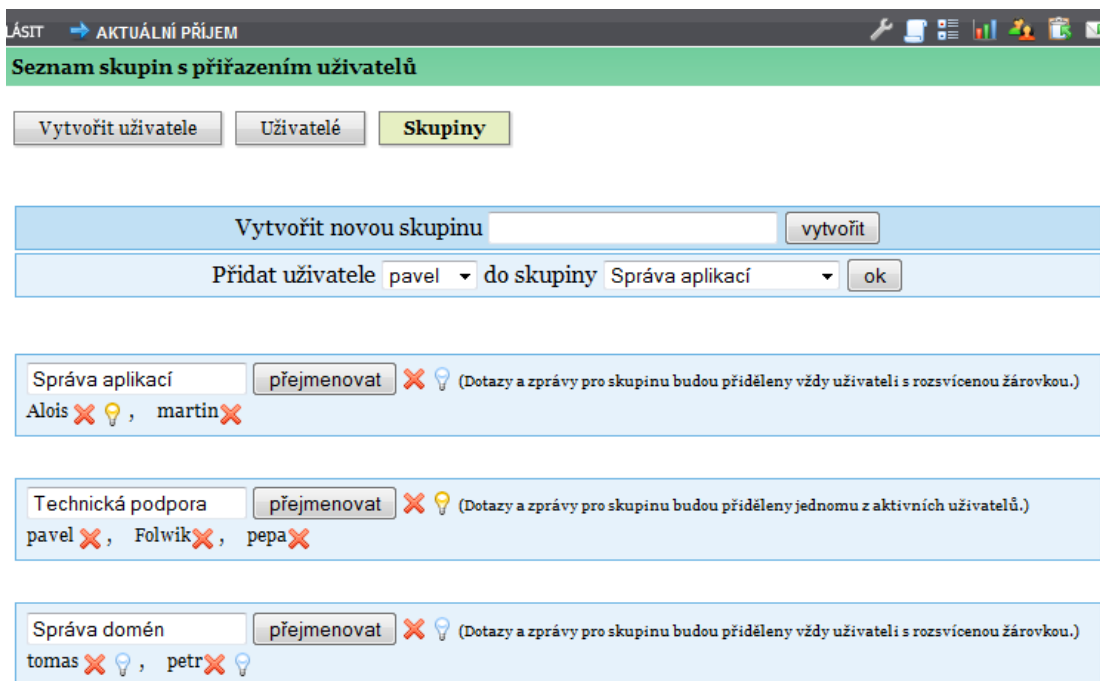
První modrá sekce obsahuje formuláře pro vytvoření nové skupiny a přiřazení uživatelů do skupin. Každá další sekce reprezentuje již jednu skupinu. Kromě formuláře pro přejmenování skupiny a tlačítka ve tvar křížku, kterým se skupina smaže, je zde ikona žárovky. Ta indikuje zda je uživatelům skupiny dovoleno se aktivovat. Povolení či zakázání aktivace se provede kliknutím na tuto ikonu. Informace o tom, komu se budou přidělovat dotazy zasláné skupině je v následujících závorkách.

Na dalším řádku je seznam uživatelů zařazených do skupiny. Vpravo od uživatelského jména jsou dvě klikatelné ikony. První – červený křížek – slouží k odebrání uživatele ze skupiny. Druhá – ikona žárovky – slouží k výběru výchozího uživatele skupiny.

Na obrázku Obrázek 7.1 jsou zobrazeny tři skupiny. Skupina s názvem „Správa aplikací“ má zakázáno aktivování uživatelů (ikona žárovky nesvíí), ale vybraného výchozího uživatele skupiny Aloise. Pokud by byla potřeba změnit výchozího uživatele z Aloise na Martina, stačí kliknout na ikonu žárovky u uživatele Alois, tím se zobrazí zhaslé žárovky u všech uživatelů

skupiny (tak jako je to nyní na obrázku u poslední skupiny „Správa domén“), a kliknout na žárovku u uživatele martin.

Při povolení aktivace uživatelů v rámci skupiny to bude vypadat jako u skupiny „Technická podpora“. Pokud před tímto nastavením byl vybrán výchozí uživatel, automaticky se mu tato funkce odebere. Je vidět, že v tomto případě nelze vybrat výchozího uživatele.



Obrázek 7.1 Výsek stránky s nastavením skupin

7.3. Rozdělení upravených zdrojových kódů

Zdrojové kódy, jichž se týkaly hlavní úpravy byly celkově přepracovány. Částí kódu byly rozděleny a seskupeny podle procesu jenž vykonávají. Takové rozdělení pro soubor `precti_mail.php` je například popsáno po bodech v kapitole 5.4 Vyzvedávání dotazů. Ve zdrojovém kódu je pak pro každý bod napsán krátký komentář objasňující, co tato část kódu provádí.

Obdobně byly rozděleny i další hlavní soubory, jedná se především o ty, do kterých se při úpravách zasahovalo nejvíce (`main.php`, `kontrola_pozastavene.php`, `index.php`). Toto rozdělení a zpřehlednění by mělo nápomoci v analýze systému při případném dalším rozšiřování. Zpřehlednění struktury adresářů a souborů, které také přispívá k větší přehlednosti zdrojových dat bylo popsáno v kapitole 5.1 Roztřídění a aktualizace.

8. Závěr

Výsledkem této bakalařské práce je upravený systém podle požadavků popsaných v kapitole 3.2. Ten nyní umožňuje rozdělování zaslaných dotazů na emaily technické podpory mezi více uživatelů systému. Toto rozdělování je pro koncové uživatele systému neviditelné, takže není nutné žádné přeškolení. Výsledné řešení navíc zabraňuje i vzniku chyb, které mohly v původní verzi nastávat a řeší nepřesnosti spojené s nastavováním způsobu přidělování dotazů. Zvolený způsob výběru uživatele pro přidělení přijatého dotazu byl sjednocen se všemi procesy, ve kterých je tento výběr potřebný. Dosáhne se tak vždy požadovaných výsledků a je možné ho použít i při dalším rozšiřování systému.

Kromě těchto požadovaných úprav jsem do systému přidal i prvky rozšiřující uživatelské rozhraní, které uživatelům napomáhají v lepší orientaci a informovanosti o aktuálním stavu systému. Zvýší se tak použitelnost systému a díky poskytnutí a zpřehlednění informací i rychlost práce se systémem. Pro lepší orientaci ve zdrojových i ostatních souborech potřebných pro běh systému jsem změnil adresářovou strukturu a názvy souborů, a přeorganizoval hlavní části původních zdrojových kódů. Pro další vývoj systému tak bude vše přehledně rozdělené a jasně vypovídatelné.

Další vývoj

Při dalším vývoji tohoto systému, kde by se přidávaly nové části, by mohl být systém rozdělen na jakési moduly, kde by každý modul řešil své vlastní funkce. Ve zdrojových kódech by byly tyto moduly od sebe odděleny a byla by možnost jednoduše jednotlivé moduly upravovat bez nutnosti zásahu do jádra celého systému. U nových rozšíření by pak mohly být využity i již vytvořené oznamovací okna, které jsem použil pro oznamování procesů spojených s aktivací a přiřazováním dotazů.

9. Literatura

- [1] KRÁTKY, Michal; BAČA, Radim. Databázové systémy. Ostrava : Katedra informatiky Fakulta elektrotechniky a informatiky Technická univerzita Ostrava, 2009. 258 s. Dostupné z WWW: <<http://dbedu.cs.vsb.cz/dbcb/dbcb.pdf>>.
- [2] BAČA, Radim. Správa databází 2010 [online]. Ostrava : VŠB - Technická univerzita Ostrava Fakulta elektrotechniky a informatiky, 2010-01-29 [cit. 2010-05-05]. Správa databází - 6.přednáška, 15 s. . Dostupné z WWW: <<http://dbedu.cs.vsb.cz/courses/2009-2010/sd/doc/sd-prednaska6.pdf>>.
- [3] PHP: Hypertext Preprocesor [online]. 2010-04-04 [cit. 2010-05-05]. Dostupné z WWW: <<http://www.php.net/>>.
- [4] PHP [online]. 2010-04-30 [cit. 2010-05-05]. \$_POST - Manual. Dostupné z WWW: <<http://www.php.net/manual/en/reserved.variables.post.php>>.
- [5] MySQL 5.1 Reference Manual [online]. 2010 [cit. 2010-05-05]. The DATETIME, DATE, and TIMESTAMP Types. Dostupné z WWW: <<http://dev.mysql.com/doc/refman/5.1/en/datetime.html>>.
- [6] VRÁNA, Jakub. PHP triky : Weblog o elegantním programování v PHP pro mírně pokročilé [online]. 2008-05-16 [cit. 2010-05-05]. Rychlost vkládání do InnoDB tabulek. Dostupné z WWW: <<http://php.vrana.cz/rychlost-vkladani-do-innodb-tabulek.php>>.
- [7] JQuery : JavaScript Library [online]. 2010 [cit. 2010-05-05]. Dostupné z WWW: <<http://jquery.com/>>.

10. Přílohy

A. Přiložené CD

- /text/ - Složka obsahující text bakalářské práce.
- /code_web/ - Zdrojové kódy a ostatní soubory jichž se týkaly úpravy.
- /code_sql/ - SQL scripty pro vytvoření tabulek a vložení základních dat.