



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

PODPORA ZPRACOVÁNÍ POŽADAVKŮ ZÁKAZNÍKŮ

SUPPORT FOR CUSTOMERS' REQUEST PROCESSING

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

VOJTĚCH KALIVODA

VEDOUcí PRÁCE
SUPERVISOR

Ing. ZBYNĚK KŘIVKA, Ph.D.

BRNO 2016

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav informačních systémů

Akademický rok 2015/2016

Zadání bakalářské práce

Řešitel: **Kalivoda Vojtěch**

Obor: Informační technologie

Téma: **Podpora zpracování požadavků zákazníků
Support for Customers' Request Processing**

Kategorie: Informační systémy

Pokyny:

1. Seznamte se s rámcem .NET a dále se seznamte s informačním systémem ALVAO Service Desk (systém pro řízení technické podpory k softwarovému produktu) včetně jeho architektury.
2. Dle pokynů konzultanta a vedoucího analyzujte požadavky na podporu vyhodnocování požadavků zákazníků a navrhnete rozšíření stávajícího informačního systému. Součástí návrhu bude i evidence nových návrhů, hlasování a diskuze o zákaznických požadavcích a možnost poskytování dat do dalších podsystémů (např. formou webové služby).
3. Rozšíření dle návrhu implementujte a testujte.
4. V závěru diskutujte nasazení do provozu a další možný rozvoj realizovaného rozšíření.

Literatura:

- MacDonald, M.: ASP.NET 4 a C# 2010 - tvorba dynamických stránek profesionálně. Zonér Press, Brno, 2011.
- Cartlidge, A. et al.: Úvodní přehled ITIL v3. itSMF Ltd., 2007. Dostupné na http://itsmf.cz/wp-content/uploads/2014/04/uvodni_prehled_itol_v31663.pdf [cit. 2015-10-12].
- ALVAO Documentation Library. Dostupné na <https://doc.alvao.com/support/doc/default.aspx> [cit. 2015-10-12]

Pro udělení zápočtu za první semestr je požadováno:

- Body 1, 2 a část bodu 3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese <http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Křivka Zbyněk, Ing., Ph.D., UIFS FIT VUT**

Konzultant: Ježdík Roman, Ing., ALVAO

Datum zadání: 1. listopadu 2015

Datum odevzdání: 18. května 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2

doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Tato práce popisuje realizaci podpory zpracování požadavků zákazníků společnosti ALVAO. Cílem práce je vytvořit rozšiřující nástroj/modul do systému ALVAO Service Desk, jenž bude zákazníkům vytvářet prostor pro jejich nápady na zlepšení systému. Kromě přidávání nových nápadů na zlepšení je jim také umožněno hlasovat a vytvářet tak mezi změnami priority. Práce se nejdříve zaměřuje na technologie, na kterých je ALVAO Service Desk postavený. V další části je proveden rozbor samotného systému. Následuje návrh nástroje a nakonec je popsána jeho implementace.

Abstract

This thesis describes realization of the request support for customers' of company ALVAO. Main goal is to create a extension/module for the ALVAO Service Desk system, which will create a space for customers' ideas for improvement of the system. Besides adding new improvement ideas, customers are allowed to vote and create priorities by that. First, the thesis is focused on technology on which it is built ALVAO Service Desk. In the next part is analysis of the system itself. Followed by design of the extension and finally description of its implementation.

Klíčová slova

podpora, ALVAO, ITIL, Service Desk, C#, MVC, .NET, zpracování požadavků zákazníků, webová aplikace

Keywords

support, ALVAO, ITIL, Service Desk, C#, MVC, .NET, customers' request processing, web application

Citace

KALIVODA, Vojtěch. *Podpora zpracování požadavků zákazníků*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Krivka Zbyněk.

Podpora zpracování požadavků zákazníků

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Zbyňka Křivky, Ph.D. Další informace mi poskytl Ing. Roman Ježdík, ředitel vývoje ve společnosti ALVAO, s.r.o. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Vojtěch Kalivoda
18. května 2016

Poděkování

V této sekci je možno uvést poděkování vedoucímu práce a těm, kteří poskytli odbornou pomoc (externí zadavatel, konzultant, apod.).

© Vojtěch Kalivoda, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	3
2 Rámec .NET	4
2.1 Jazyk C#	5
2.2 Návrhový vzor Model-View-Controller	6
3 ITIL	9
4 ALVAO	11
4.1 ALVAO Service Desk	11
4.1.1 Architektura systému	12
4.1.2 Role v systému ALVAO Service Desk	13
4.1.3 Vlastní úpravy a rozšíření	14
5 Analýza	16
5.1 Analýza požadavků	16
5.2 Analýza existujících nástrojů	16
6 Návrh a implementace	18
6.1 Specifikace požadované funkčnosti	18
6.2 Databázové schéma systému Service Desk	22
6.3 Návrh nástroje	23
6.3.1 Datové modely	23
6.3.2 Kontrolér	26
6.3.3 Grafické uživatelské rozhraní	26
6.3.4 Konfigurace systému	28
6.4 Implementace	28
6.4.1 Testování během implementace	28
6.4.2 Uživatelské příručky	29
6.4.3 Nasazení do provozu	30
7 Závěr	31
Literatura	32
Přílohy	33
Seznam příloh	34
A Obsah CD	35

B	Uživatelské příběhy	36
B.1	Uživatelský příběh zákazníka	36
B.2	Uživatelský příběh vývojáře ALVAO	37
C	Uživatelské příručky	38
C.1	Nápady – Příručka pro uživatele	38
C.2	Nápady – Správa	40
D	Výsledné snímky nástroje	42

Kapitola 1

Úvod

Tato práce popisuje realizaci podpory zpracování požadavků zákazníků společnosti ALVAO. Cílem práce je vytvořit rozšiřující nástroj/modul do systému Service Desk, jehož vývojářem je již zmíněná společnost. Nástroj najde využití pouze v instanci aplikace, kterou firma pro svůj chod interně využívá a jsou v ní uživatelské účty vývojářů a zákazníků firmy.

Nástroj pro zákazníky vytváří prostor, kde mohou přidávat vlastní nápady na zlepšení aplikace, následně mohou pro tyto navrhované změny hlasovat a vytvářet tak mezi nápady priority. Kromě vytváření priorit je uživatelům dovoleno k jednotlivým nápadům přidávat zpětnou vazbu a za její pomoci tak specifikovat požadované změny.

Důležitým aspektem je to, že nástroj pracuje pouze s prostředky, které jsou v systému běžně dostupné. Modul tak pouze přidává nové funkce nad již existujícími prostředky a dovoluje na ně nahlížet jiným způsobem, než je v aplikaci Service Desk běžné.

V první části této práce jsou popsány vybrané technologie pro tvorbu webových aplikací, na kterých je postavená aplikace Service Desk.

V další kapitole je uveden popis praktik, podle kterých je systém vyvíjen a také obecný popis prostředků, na kterých je systém postavený.

Další část práce je zaměřena na rozbor a popis samotného systému Service Desk. Lze v ní najít popis architektury, uživatelských rolí a možností, jak program upravovat.

Po rozboru je provedena analýza požadavků a aspektů, na které má být nahlíženo při tvorbě návrhu. Její součástí je také prozkoumání existujících aplikací s podobným zaměřením.

V návrhu nástroje, který následuje, jsou uvedeny specifické požadavky na funkčnost, dále je popsán způsob integrace do systému a také struktura nástroje.

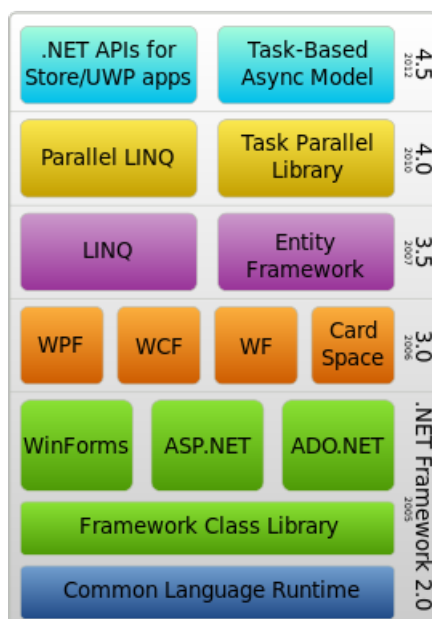
V závěru práce je popsána implementace, nezbytná konfigurace systému a také popis nasazení nástroje do provozu.

Kapitola 2

Rámec .NET

.NET Framework je rámec vyvíjený firmou Microsoft, určený pro běh na operačních systémech Microsoft Windows. Jeho vývoj začal na konci 90. let a stále pokračuje. Nejnovější verzí, vydanou v roce 2012, je .NET Framework 4.5.

Tak jak je vidět na obrázku 2.1, do rámce postupně přibývaly technologie pro tvorbu desktopových aplikací (WinForms, Windows Presentation Foundation), pro tvorbu webu a webových aplikací (ASP.NET) a také technologie pro zjednodušení připojení aplikací k databázím (ADO.NET, Entity Framework).



Obrázek 2.1: Komponenty .NET Framework [7]

Rámec pracuje od verze 2.0 s tzv. **Common Intermediate Language**. CIL je nízkoúrovňový jazyk nezávislý na platformě, do kterého jsou překládány všechny jazyky běžící nad rámcem. Tento mezikód je interpretovaný pomocí **Common Language Runtime**, což je virtuální stroj pracující na konkrétní platformě.

CLR se nestará pouze o interpretaci kódu, ale zajišťuje také správu paměti, kontrolu typů, provádí optimalizace atd. [5] Mezikód je platformě nezávislý a proto je také přenositelný. Překlad jazyků do stejného mezikódu umožňuje při vývoji kombinovat například

objektově orientované i funkcionální jazyky.

.NET Core

Microsoft na konci roku 2014 oznámil vývoj *.NET Core* – open source verze .NET Frameworku, díky níž je možné spustit aplikace překládané do CIL i na operačních systémech Linux a Mac. Vývoj stále probíhá, zdá se, že nezaplatněná implementace rámce se ještě dodělává, a tak doposud neumí pracovat s jazyky Visual Basic nebo F# a postrádá několik dalších částí frameworku.

Přestože politika Microsoftu se nejspíše začíná ubírat jiným směrem než doposud, malá podpora na jiných platformách prozatím stále zůstává největší nevýhodou .NET Frameworku.

Visual Studio

Microsoft vyvíjí pro programátory vývojové prostředí *Visual Studio*. Visual Studio je i přes svoji těžkopádnost užitečný nástroj, který umožňuje efektivní ladění kódu a jeho refaktori-zaci.

V souvislosti s vydáním .NET Core vydal Microsoft pro nezávislé vývojáře nové vývojové prostředí *Visual Studio Code*, které je kompletně open source a díky němuž je možné vyvíjet pod rámcem komerčně bez nutnosti investice do prostředků.

2.1 Jazyk C#

C# je silně typovaný, objektově orientovaný jazyk. Slouží především k vývoji desktopových a webových aplikací. Jeho vývoj začal před rokem 2000, současně s chystaným prvním vydáním rámce .NET. Jeho vývoj, stejně jako vývoj celého rámce, stále pokračuje. Zatím poslední verze 6.0 byla vydána v roce 2015. Do jazyka postupně přibývají prvky moderních programovacích jazyků.

O správu paměti se stará *Garbage collector*. Tento nástroj vyhledává reference na objekty, které již nejsou využívány a z paměti je uvolňuje. Vývojář se tak nemusí starat o manuální dealokaci paměti, kterou jeho program využívá. Tento proces vede k úspoře času a také pomáhá předcházet zbytečným únikům paměti.

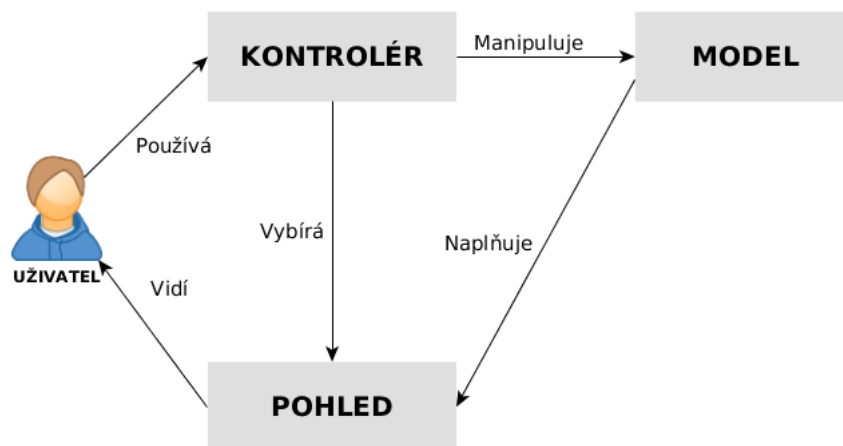
Součástí jazyka je *LINQ* neboli integrovaný dotazovací jazyk. Jak je vidět na krátké ukázce kódu níže, jeho syntaxe je blízká SQL, se kterým je velká část programátorů zvyklá pracovat. Jeho největší výhodou je to, že dovoluje zacházet s různými typy kolekcí stejným způsobem.

```
var query =
    from s in students
    where s.LastName == "Smith"
    where s.BirthDay < new DateTime(2000, 1, 1)
    select s;
```

V jazyce C# je možné při vytváření metody parametrům přiřazovat implicitní hodnoty. Díky tomu se parametr stává volitelným a programátor do něj přiřazuje hodnotu jen v případě potřeby. Takto je možné vyhnout se zbytečnému přetěžování a s tím spojeným nárůstem zdrojového kódu.

2.2 Návrhový vzor Model-View-Controller

Návrhové vzory jsou nejlepšími praktikami pro práci s objektově orientovanými jazyky. Zastupují principy pro nejlepší řešení často se vyskytujících problémů. Díky tomu, že principy nejsou závislé na platformě a implementačním jazyku, nalézají široké využití. V posledních letech se jedním z hojně využívaných vzorů stal *Model-View-Controller*.



Obrázek 2.2: Schéma pro návrhový vzor Model-View-Controller

Název tohoto návrhového vzoru vychází z jeho tří základních komponent:

- **Model** – Model zajišťuje základní operace nad trvalými daty. Je v něm uchována základní logika pro čtení dat, jejich modifikaci a zapisování do databáze. Model je jedinou částí aplikace, která komunikuje s databází.
- **Pohled (View)** – Pohled napojuje jednotlivé části modelu na grafické ovládací prvky a tvoří tak celé grafické rozhraní.
- **Kontrolér (Controller)** – Kontrolér je mostem mezi modelem a pohledem. Hlavní úlohou této komponenty je zpracovávat vstupy od uživatele, transformovat je na akce modelu, vybrat požadovaný pohled a tomu předat model ke zobrazení. [5]

Hlavní výhodou nabízí *MVC* v oddělení jednotlivých vrstev. Díky tomu, že jsou vrstvy navzájem nezávislé, je možné je nezávisle vyvíjet a testovat. Nezávislost rovněž napomáhá ke znovuvyužití jednotlivých komponent.

Součástí rámce *.NET* se *ASP.NET MVC* stal v roce 2009. Pro snazší práci Visual Studio umí při vývoji pracovat s různými druhy šablon, nebo také ladit javascript. Společně s vestavěnou podporou je do projektů přidáno i několik balíčků, které se pro vývoj webu nejčastěji používají. Tyto technologie by mohly být popsány i v samostatných kapitolách, ale protože jsou součástí projektu, tak jsou uvedené zde.

HTML je značkový jazyk pro tvorbu webových stránek. HTML za pomoci tagů transformuje obyčejný text do různých bloků, kterým je pak možné pomocí CSS měnit způsob zobrazení. V *ASP.NET MVC* HTML tvoří pohledy.

CSS jsou kolekce pro grafickou úpravu webových stránek. Jejich hlavní výhodou je oddělení vzhledu stránky od jejího obsahu. Pro aplikaci stylů využívají selektory prvků, tříd a identifikátorů.

jQuery je open source knihovna pro javascript. Oproti javascriptu má zjednodušenou syntaxi a nabízí mnoho vestavěných funkcí. V posledních letech začla tato knihovna nacházet široké využití. Je využitelná pro jednoduché i složité animace a velmi ulehčuje práci s asynchronním přenosem.

Bootstrap je sada kaskádových stylů, html a javascriptu, která slouží k jednoduché stylizaci stránek a vytváření responzivních webů.

Technologie ASP.NET Razor

Pro jednodušší napojení pohledů na modely je součástí jazyka technologie *ASP.NET Razor*. Tato technologie dovoluje psát úseky kódu v C# přímo do pohledu (view), který je jinak HTML dokumentem. V pohledu lze vytvářet dodatečné akce, které přímo nesouvisí s funkčností modelu nebo je možné provádět transformace dat do jiné podoby než jak jsou uložena.

Následující příklad ukazuje, jak přímo v pohledu vytvořit proměnnou, přiřadit jí hodnotu a následně ji vložit do HTML.

```
<html>
@{
    string text = "Title";
}
<body>
<h1>@text</h1>
</body>
</html>
```

Html helpers Součástí technologie *Razor* jsou také takzvané *HTML helpers*, které obsahují metody pro automatické generování HTML elementů. Na příkladu je vidět vytvoření odkazu do „Settings“ v aktuálním kontroléru. Takto lze vytáčet například ještě skryté položky (`HiddenFor`), popisy k položkám (`LabelFor`) nebo šifrované editory (`PasswordFor`).

```
@Html.ActionLink("User settings", "Settings")
```

Největší výhodou je skládání pohledů z jednotlivých komponent, čímž helpery velice pomáhají ke znovupoužitelnosti grafických prvků (pohledů).

Komponenty se dělí na dva základní typy:

1. Editor, který generuje html elementy pro editaci hodnot.
2. Display sloužící pro zobrazení hodnot pouze ke čtení.

V projektové složce pro sdílené komponenty se nachází pohled, který komponentu reprezentuje.

V příkladu níže je ukázáno použití. Prvním argumentem je vlasnost modelu „Name“, která je z pohledu komponenty její model. Druhým argumentem je název komponenty „Stringfield“, která se stará o vykreslení.

```
@Html.EditorFor(Model.Name, "StringField")
```

Technologie Razor zajišťuje, že komponenta je vložena do nadřazeného pohledu. Kus kódu uvedený výše bude ve výsledném pohledu vypadat přibližně takto:

```
<label for="Name">First Name:</label>  
<input name="Name" />
```

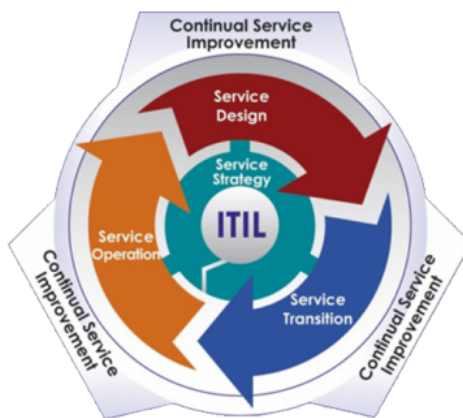
Kapitola 3

ITIL

Information Technology Infrastructure Library je sada praktik pro ITSM¹, zaměřených na řízení, správu a kvalitní využití IT služeb. Nesnaží se řešit konkrétní situace, ale jde spíše o sadu obecných doporučení, které jsou určeny k implementaci [6]. Jedním z důvodů úspěchu ITIL je, že se nezaměřuje jenom na samotné řízení IT služeb, ale na služby se snaží nahlížet i z pohledu zákazníka.

ITIL v současné době obsahuje pět publikací, z nichž každá je zaměřena na jinou část životního cyklu služby:

- Strategické procesy (service strategy).
- Návrh služeb (service design).
- Uvedení služby do provozu (service transition).
- Provoz služeb (service operation).
- Zlepšování služeb (continual service improvement).



Obrázek 3.1: Diagram životního cyklu služeb [6]

Dodržování principů, které tento balík obsahuje, přináší zlepšení komunikace, zlepšení podkladů pro rozhodování, jednodušší testovatelnost kódu, optimalizaci rizik, transparentnost procesů z hlediska provedení a celkové zvýšení produktivity.

¹IT Service Management - řízení IT služeb

Existuje také výkladový slovník ITIL, který definuje mnoho výrazů z oblasti IT služeb:

Služba je prostředek poskytování hodnoty zákazníkovi prostřednictvím výstupů, kterých zákazník chce dosáhnout bez vlastnictví specifických nákladů a rizik. Termín „služba“ se někdy užívá jako synonymum pro klíčovou službu, službu IT nebo balíček služby [1].

SLA (Service Level Agreement) je dohoda o úrovni poskytovaných služeb. Ve slovníku definovaná jako dohoda mezi poskytovatelem služeb IT a zákazníkem. Dohoda o úrovních služeb popisuje službu IT, dokumentuje cíle úrovní služeb a specifikuje odpovědnosti poskytovatele služeb IT a zákazníka. Jedna dohoda o úrovni služeb může pokrývat řadu služeb IT nebo více zákazníků [1].

Proces je strukturovaná množina činností navržená pro dosažení určitého specifického cíle. Proces má jeden či více definovaných vstupů a přetváří je do definovaných výstupů. Může obsahovat jakékoli role, odpovědnosti, nástroje a manažerské kontrolní mechanismy požadované pro spolehlivou dodávku výstupů. Proces může v případě potřeby definovat politiky, normy/standardy, směrnice, činnosti a pracovní instrukce [1].

Na těchto prostředcích je založena také aplikace ALVAO Service Desk 4.1, se kterou je úzce spjatá i podpora zpracování požadavků zákazníků.

Kapitola 4

ALVAO

ALVAO je balík produktů, které podporují efektivní práci a pořádek v organizacích. Skládá z produktů:

- ALVAO Asset Management - správa IT majetku a dalších aktiv.
- ALVAO Service Desk - řízení požadavků, úkolů a projektů.
- ALVAO Monitoring - sledování aktivit uživatelů a využití hw/sw.

ALVAO přináší nejvyšší hodnotu při společném používání uvedených produktů, ale jednotlivé aplikace mohou být používány i samostatně [2]. Všechny produkty jsou neustále vyvíjeny jednak podle vývoje v oboru informačních technologií, ale zároveň podle ohlasů a požadavků uživatelů, kteří je využívají pro svoji práci.

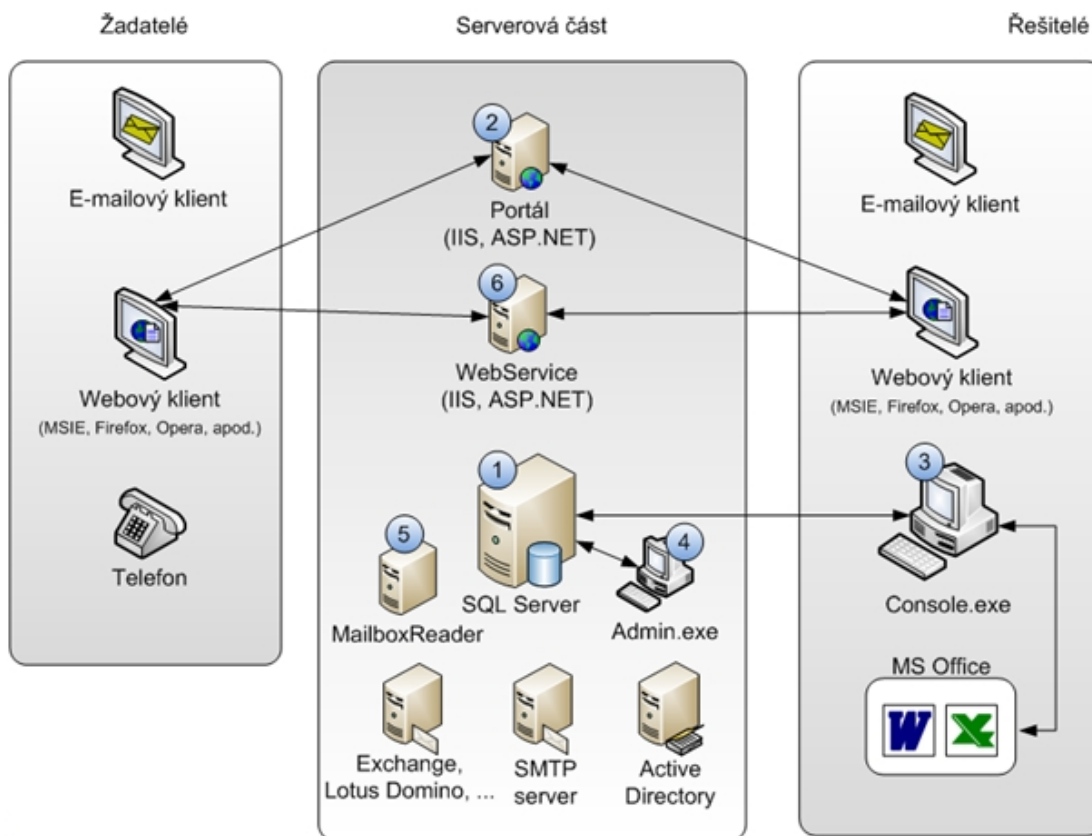
Společnost ALVAO dlouhodobě využívá rámce .NET Framework pro vývoj svých aplikací. Z tohoto důvodu probíhá implementace v jazycích C++ a C#.

4.1 ALVAO Service Desk

ALVAO Service Desk (dále také SD Portál) je systém technické podpory, který:

- Zajišťuje sběr požadavků od uživatelů,
- průběžně informuje žadatele o řešení jejich požadavků,
- dokumentuje komunikaci mezi žadatelem a řešitelem,
- dokumentuje způsob řešení,
- archivuje splněné požadavky,
- vyhodnocuje práci techniků a celého oddělení technické podpory.

Systém umožňuje vytváření libovolných služeb s vlastními procesy, oprávněními, e-mailovými schránkami apod. Díky univerzálnímu přístupu je možné ho využívat napříč různými oblastmi IT [3].



Obrázek 4.1: Architektura ALVAO Service Desk

4.1.1 Architektura systému

System je tvořen serverovou a klientskou částí. Serverovou část aplikace tvoří:

- Databázový systém MS SQL Server, kde jsou uložena veškerá data.
- Portál ALVAO WebApp, ke kterému se klienti připojují přes libovolného webového klienta (např. MS IE, Mozilla Firefox, Opera apod.).
- Webová služba (WebService), která slouží k automatickému vytváření oznamovacích zpráv.
- MailboxReader – komponenta, která rozesílá emailové zprávy a dokáže automaticky načítat požadavky. Využívá přitom protokolu IMAP.
- ALVAO Admin – aplikace pro administraci systému.

Klientskou část tvoří:

- Libovolný webový klient pro připojení na portál.
- Win32 aplikace Service Desk Console, která je alternativou pro webového klienta. Lze v ní provádět stejné akce a operace jako přes webový portál.

Základní entity se kterými aplikace pracuje:

ALVAO Office 365 Požadavky Katalog služeb Báze znalostí Aktuální zprávy

Prohledat požadavky ...

nový požadavek z katalogu služeb nebo zapsat požadavek

řešené testing volné pro sprint všechny požadavky mé požadavky nepřidělené bp ...

ikony	Priorita	Žadatel	Číslo pož.	Název požadavku
	střední	Jan Prudík	52030	SD - OPRAVIT: Požadavek - Nový navázaný požadavek/Přidat vazbu - Typ vazby - v
	střední	Petra Kalábová	53873	ALVAO WA - OPRAVIT: SuggestionBox - pokud je vyplněný text příliš dlouhý, comb
	střední	Jan Prudík	54517	Sběr požadavků, diskuze a hlasování o prioritě přes web - návrh
	střední	Jan Prudík	54516	Sběr požadavků, diskuze a hlasování o prioritě přes web - analýza
	střední	Jan Prudík	48055	SD - Požadavek - zrušit systémovou položku "Požadovaný termín"
	střední	Roman Ježdík	53676	Sběr požadavků, diskuze a hlasování o prioritě přes web

Zobrazit všechny položky včetně archivu

Obrázek 4.2: Seznam požadavků v ALVAO Service Desk WebApp

- **Služba** je pro uživatele (žadatele) prostředkem pro řešení jeho požadavků. Každá služba má svůj řešitelský tým (viz 4.1.2), vlastní dohodu o úrovni služeb – SLA, má také definovaný proces, kterým požadavky prochází během svého životního cyklu. Služby v aplikaci většinou zastupují reálné služby, jako jsou například Technická podpora, Oddělení vývoje, Marketing apod.
- **Proces** je tvořen posloupností stavů, mezi kterými přechází požadavek během svého řešení. Na procesu je možné nastavit například schvalování nebo automatické přepnutí do stavů při určitých událostech. Jeden proces může využívat jedna nebo více služeb.
- **SLA** v Service Desku definuje kvalitu, s jakou je služba poskytována. V aplikaci ALVAO Admin, která slouží na správu celého systému, jdou vytvářet libovolná SLA a přiřazovat je jednotlivým službám. SLA využívá i webová služba, která prochází statistiky a generuje oznámení, která pomocí emailů upozorňují řešitele na neřešení požadavků.
- **Oznámení** jsou krátké emailové zprávy, které generuje webová služba ALVAO Web-Service. Po vygenerování je zpráva umístěna do databázové tabulky, kterou vyčítá služba MailboxReader běžící na pozadí. MailBoxreader komunikuje přes protokol SMTP s emailovým serverem a stará se tak o rozeslání zpráv. Oznámení je možné různě nastavovat v aplikaci Admin. Mají několik úrovní nastavení. Systém dovoluje správcům služeb vytvořit globální nastavení pro službu. Každý uživatel si však může vytvořit svoje vlastní nastavení, které je před globálním upřednostněno. Jejich vygenerování nastává při různých událostech jako je vytvoření požadavku, změna stavu požadavku, změna řešitele atp.

4.1.2 Role v systému ALVAO Service Desk

V ALVAO Service Desk jsou implementované uživatelské role:

- **Žadatel** – do systému přidává své požadavky, sleduje jejich stav a čeká na vyřízení. Vidí pouze své vlastní požadavky.

- Řešitel – vidí všechny požadavky ve svých službách včetně všech detailů. Požadavky jsou mu předávány k řešení a následnému uzavření.
- Operátor – tato role slouží k předávání požadavků řešitelům na službě.
- Manažer – vidí všechny požadavky ve svých službách. Slouží k vyhodnocování práce operátorů a řešitelů.
- Správce – přiděluje uživatelské role a provádí další administrační záležitosti.
- Schvalovatel – uživatel s touto rolí vidí pouze požadavky, které je potřeba schválit, což je smyslem jeho činnosti.

Role je možné kombinovat, tak aby odpovídaly rolím fyzických osob, které systém využívají. V menších týmech pak může například roli operátora a řešitele zastupovat jeden uživatel.

4.1.3 Vlastní úpravy a rozšíření

Pro náročné uživatele systém Service Desk nabízí vlastní úpravy a rozšíření. Jedná se o různé speciální konfigurace a přidané zdrojové kódy, ke kterým má přístup pouze správce systému.

Popisy, jak provádět úpravy a přidávat rozšíření, jsou, včetně ukázkových příkladů, umístěny v dokumentaci k produktu. Hlavní výhodou, kterou úpravy přinášají, je to, že změny, které by za normálních podmínek vyžadovaly programovou úpravu, mohou být realizované i po nainstalování aplikace.

Vlastní formuláře

Nejširším pojmem jsou v tomto ohledu *vlastní formuláře*, což jsou externí soubory se zdrojovým kódem v C#. Tyto soubory správce systému vkládá do vyhrazené složky *Custom*, nacházející se v adresáři s nainstalovaným SD Portálem. Jsou linkované dynamicky za běhu, a tak mohou být kdykoli modifikované. Implementace takového formuláře vyžaduje alespoň minimální znalost databáze ALVAO.

V aplikaci Admin je několik míst, kde se dají formuláře použít. V nastavení služby se nachází pole pro *vlastní workflow*. Do pole se vkládá cesta k vlastnímu formuláři, který se má zobrazit místo standardního detailu požadavku.

Kromě workflow je na službě možné nastavit vlastní formulář pro nový požadavek. Správce si může přidat položky na formulář v libovolném pořadí, může některé skrýt nebo naopak zobrazit (např. vlastní položky).

Vlastní položky

Jedno z hojně využívaných rozšíření je přidávání vlastních položek (sloupců) do databáze. V databázi je k tomu připraveno několik tabulek se suffixem *Cust*. *tHdTicketCust* pro vlastní položky na požadavku, *tHdPersonCust* pro vlastní položky na osobách, *tHdSectionCust* pro vlastní položky na službě atd.

Vlastní položky se vytváří aplikačně v ALVAO Admin. Vybírá se typ a název, případně seznam hodnot. Správce vybere do jaké tabulky položku chce přidat a vytvoří jí. Poté když ji přidá do některého z procesů, tak ji uživatelé vidí na vyhrazených místech.

Úpravy hlavní nabídky

Určité změny dovoluje i hlavní nabídka. V dokumentaci je ukázkový databázový skript, který v nabídce vytvoří uživatelem definovaný odkaz ve formě tlačítka. I tímto způsobem je možné odkazovat na vlastní formuláře.

V dokumentaci se nachází ještě několik dalších rozšíření, žádné z nich již nebylo využito při implementaci nástroje a proto není potřeba je zde více popisovat.

Informace pro tuto kapitolu byly čerpány z dokumentace aplikace Service Desk [4].

Kapitola 5

Analýza

Tato kapitola je zaměřena na vymezení cílů a stanovení podmínek a aspektů pro návrh a implementaci. Nejdříve je zde provedena analýza zadání a za ním následuje krátký popis průzkumu existujících nástrojů pro sběr zpětné vazby.

5.1 Analýza požadavků

Pro správné pochopení záměrů firmy ALVAO byla provedena konzultace s panem Ing. Romanem Ježdíkem. Na ní bylo představeno, jak přibližně bude podpora zpracování požadavků zákazníků vypadat.

Společnost ALVAO interně využívá systém Service Desk pro řízení chodu firmy. Do tohoto systému mají přístup i zákazníci, kterým je účet vytvořen po zakoupení některého z produktů. Cílem práce je vytvoření nástroje/modulu pro systém Service Desk, který bude podporu zpracování požadavků zákazníků zprostředkovávat.

Prvním důležitým krokem bylo pochopení, jak je modul spjatý se systémem Service Desk. Ze zadání byly vyjmuty tři pilíře, které ukazují, co má nástroj do systému přinést a co je potřeba při návrhu dodržet.

1. *Přidat novou funkčnost* – Sbírat relevantní data pomocí koncových uživatelů. Nechat je vyjadřovat vlastní myšlenky a návrhy na zlepšení skrze požadavky. Vybízet je k interakci, dovolit jim hlasovat a reagovat na změny v komentářích.
2. *Zachovat stávající funkčnost* – Umožnit aplikaci Service Desk provádět s danými požadavky veškeré běžné akce. Požadavek může být upraven, dají se na něm posílat emailové zprávy, lze přidávat poznámky, prohlížet deník atd.
3. *Neměnit kód aplikace ani databázové schéma* – Přidat nástroj jako rozšiřující modul, ne jako pevnou součást aplikace. Rozšíření není určeno pro produkční verzi, využití najde pouze v rámci firmy, tzn. slouží právě pro tu instanci aplikace Service Desk Portál, kterou firma používá interně.

5.2 Analýza existujících nástrojů

V rámci analýzy byl proveden průzkum webových aplikací, které mají podobné zaměření jako připravovaný nástroj. Při této příležitosti jsem krátce prozkoumal portál *uservoice.com*.

Uservice.com

Nástroj *uservice* je určen pro vývojářské týmy a společnosti, které s jeho pomocí můžou od uživatelů dostávat zpětnou vazbu a nové nápady. Uživatelé zde vytváří příspěvky, u kterých je jim dovoleno hlasovat a diskutovat.

Uživatelské rozhraní se snaží vybízet k interakci. Na úvodní (zároveň hlavní) straně zaujme nově příchozího velké editační pole pro vytváření nového příspěvku. V něm uživatel nevytváří přímo nový příspěvek, ale při vyplňování názvu jsou mu nejdříve našeptávané již existující příspěvky s podobným názvem a až po potvrzení, že žádný z nich neodpovídá uživatelskému nápadu, jde pokračovat dále.

Hlavní strana dále zobrazuje přehled již vytvořených příspěvků. V přehledu se u každého dá hlasovat a zobrazuje se počet komentářů. Přes kliknutí na název příspěvku se uživatel dostává na detail. Na něm je popis, pole pro nový komentář a pod ním všechny přidané komentáře.

Nejslabší stránkou *uservice* je formátování textu a vkládání příloh k příspěvkům. Ani jedno totiž v systému není možné. Za nejsilnější část lze naopak považovat konzoli pro správu. Ta nabízí vytváření různých fór, nastavení lokalizace, upravení vzhledu, zobrazení různých dashboardů atd.

Protože se jedná o externí systém, jeho použití nebylo brané v úvahu. Zákazníci jsou už v systému Service Desk přítomni a nemá smysl je odkazovat na aplikaci třetí strany. Každý příspěvek by musel navíc mít v SD Portálu ekvivaletní požadavek, což by vedlo ke zvyšování rizika neaktuálnosti informací, pomalejší reakční době a celkovému zvýšení režie při správě.

Kapitola 6

Návrh a implementace

Tato kapitola se zabývá návrhem nástroje a jeho následovnou implementací. První část je zaměřena na specifikaci požadované funkčnosti, v další části je popsána část databázového schématu systému Service Desk a po něm následuje specifický návrh. Na konci kapitoly jsou uvedeny implementační detaily, testování a nasazení do provozu.

Ve firmě ALVAO bylo pro nástroj vybráno jméno **Nápady**. Tento název, ačkoliv se může jevit lehce neformální, celkem dobře vystihuje ten druh požadavků, se kterými nástroj pracuje. Jeho účelem je sbírat zlepšovací nápady od skupiny koncových uživatelů, filtrovat je a třídit podle získané zpětné vazby. Tento název je pro nástroj použitý i v návrhu.

6.1 Specifikace požadované funkčnosti

Na konzultaci s Ing. Romanem Ježdíkem byly vytvořeny uživatelské příběhy z pohledu zákazníka a z pohledu zaměstnanců firmy. Jejich znění je možné najít v příloze **B**.

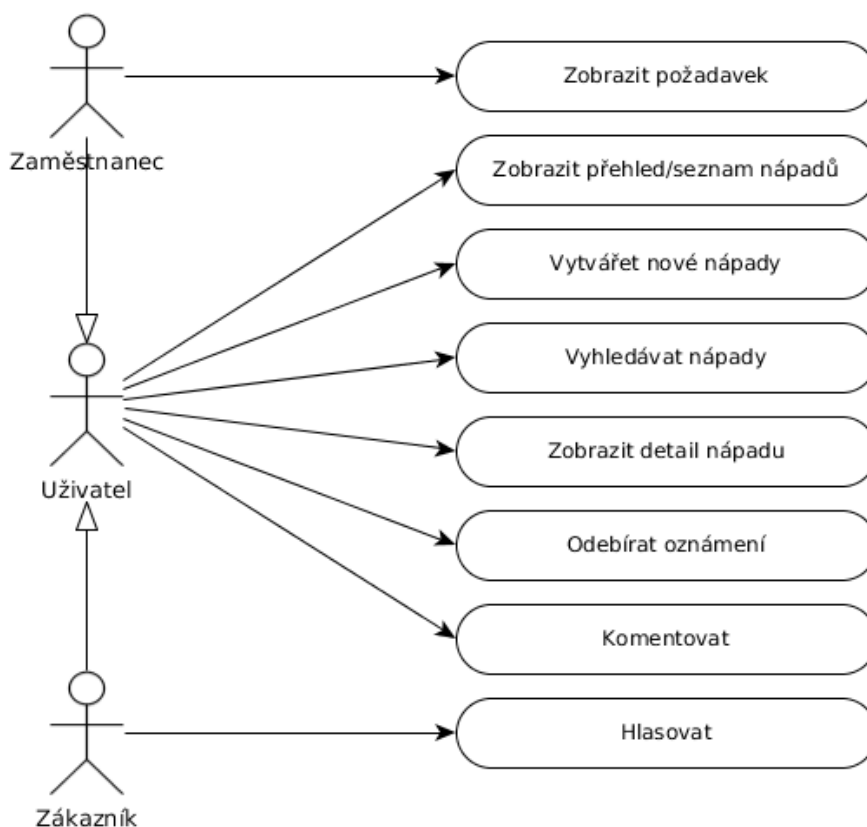
Za pomoci příběhů došlo k vytvoření use case diagramu (obrázek **6.1**), jenž shrnuje požadovanou funkčnost. Use case zachycuje pouze tu funkčnost, která je předmětem práce. Z příběhů rovněž vyplývá, že nástroj nepotřebuje rozlišovat mezi všemi uživatelskými rolemi v systému, ale stačí když bude rolišovat mezi skupinou výhradně žadatelskou (zákazníci) a na všechny ostatní role nahlížet jako na druhou uživatelskou skupinu (zaměstnanci). „Výhradně žadatelskou“ je zde zmíněno záměrně. Z diskuze během konzultace vyplynulo, že možnost vytváření nových nápadů mají mít i zaměstnanci firmy ALVAO. Stejně tak není problém, aby kdokoli z nich přidával komentáře k nápadům, dostával oznámení atd. Zákazníci tak mají oproti zaměstnancům jen jedinou výhodu a tou je hlasování, které vytváří ve změnách priority. Oproti tomu zaměstnanci můžou nápad zobrazit jako běžný požadavek a upravovat ho.

Nejlepší praktikou pro návrh se ukázalo připravit sadu pohledů, kterou bude potřeba podrobně navrhout, a ke každému popsat funkčnost, kterou nabízí. Návrh byl průběžně zasílán k posouzení zadavatelem, a tak do něj postupně byly doplněny všechny potřebné informace.

Seznam nápadů

Do hlavní nabídky je přidán odkaz, přes který se uživatel dostane na hlavní stranu Nápadů – seznam nápadů. Seznam nápadů je tvořen požadavky ze speciální vyhrazené služby.

Je na něm možné zobrazit si sloupce:



Obrázek 6.1: Use case diagram nástroje

- Hodnocení – součet vah hlasů k danému nápadu (viz 6.1). Od každého uživatele se počítá pouze poslední přidaný hlas k danému nápadu.
- Můj hlas – volba, kterou uživatel hlasoval k danému nápadu naposledy.
- Název nápadu – název požadavku.
- Produkt-modul/aplikace – vlastní položka na požadavku z interního provozu aplikace.
- Počet hlasů – počet uživatelů, kteří hlasovali k danému nápadu. Každý uživatel se počítá pouze jednou.
- Počet komentářů – počet komentářů přidaných k nápadu.
- Číslo nápadu – číslo požadavku.
- Aktivita za posledních 12 měsíců – počet komentářů a hlasů přidaných k nápadu během posledních 12 měsíců.
- Autor – žadatel požadavku.
- Stav – aktuální stav požadavku.
- Odebíraný – v poli je hodnota „Ano“ nebo „Ne“ podle toho, zda uživatel je nebo není odběratelem oznámení u daného nápadu (viz 6.1).

Uživatel může podle těchto sloupců seznam řadit a přidávat na něj filtry.

Podobně jako na běžném seznamu požadavků, jsou pro uživatele připraveny různé pohledy/karty gridu, ve kterém se seznam zobrazuje. Ve výchozím stavu uživatel vidí seznamy:

- Odebírané – nápady, u kterých uživatel v naposledy hlasoval volbou „Musím mít“, „Uvítal bych“ a nebo „Nežádoucí“ (viz 6.1).
- Mé – Nápady jejichž žadatelem je právě přihlášený uživatel.
- Nejlépe hodnocené – Na této kartě je grid seřazen sestupně podle nejvyšší hodnoty v poli *Hodnocení*.
- Nejaktivnější – Pro tento pohled je v dotazu speciální pole, které počítá všechny komentáře a hlasy, které byly přidány za poslední rok a je. Podle této položky je seznam na tomto pohledu řazen.
- Nejnovější – Všechny požadavky, včetně již uzavřených, řazené sestupně podle identifikačního čísla.

Pohledy si každý uživatel může vytvářet, rušit, upravovat a následně nastavení ukládat.

Nový nápad

Ze seznamu nápadů má uživatel možnost přímo přejít na formulář pro nový nápad. Při vyplňování názvu nového nápadu vidí v seznamu nápady s podobným názvem. Jednoduše může přejít na detail podobného nápadu a prohlédnout si jeho obsah. Během této činnosti neztrácí data, která do formuláře již vepsal. Na formuláři se se dále nachází popis nápadu, který může být formátovaný a můžou k němu být přidány přílohy. Pokud se jedná o uživatele s právem hlasovat (zákazník), tak sem u tato volba nabízí již na tomto formuláři, avšak vybírá pouze z pozitivních voleb.

Pro podporu rychlejšího zpracování je na formuláři i nepovinná vlastní položka *Produkt-modul/aplikace*, kterou využívá vývojový tým. Tomu pomáhá upřesnit na kterou aplikaci a její část požadavek cílí. Uživatel při vyplnění vybírá ze seznamu hodnot, které již v systému jsou přítomny.

Detail nápadu

K detailu nápadu se uživatel dostane přes kliknutí na konkrétní nápad v seznamu. Oproti běžnému detailu, je detail nápadu jednodušší a „odstiňuje“ uživatele od nepodstatných informací, které požadavek sice nabízí, ale z pohledu Nápadů nejsou důležité. Zobrazuje se zde popis nápadu, aktuální hodnocení a seznam všech přidávaných komentářů.

Aby zaměstnanci mohli provádět na požadavku běžné úkony dostupné v SD Portálu, je zde pro ně připraven příkaz *Zobrazit požadavek*, po jehož provedení se místo nápadu zobrazí uživateli standardní detail požadavku.

Hlasování Zákaznická uživatelská skupina zde má možnost hlasovat. Škála pro hlasování je tvořena řetězci „Musím mít“, „Uvítal bych“, „Nezajímá mě“ a „Nežádoucí“. Volba „Musím mít“ zvyšuje hodnocení nápadu o dva body, „Uvítal bych“ o jeden bod, volba „Nezajímá mě“ nemá vliv na hodnocení a volba „Nežádoucí“ hodnocení o jeden bod sníží. Hodnocení nápadu se mění při změně hlasu. Pokud dojde k realizaci nápadu, tak je uzavřen. Od této

chvíle již není možné přidávat k němu nové hlasy, hodnocení se však na detailu zobrazuje i nadále.

Hlasovat je možné pouze na detailu nápadu.

Komentáře Pro všechny role je k dispozici pole pro nový komentář. Komentář může mít formátovaný text a dají se k němu přidávat přílohy. Po přidání se komentář zobrazuje v přehledu mezi ostatními.

Vyhledávání nápadů

Uživatelé nástroje mají možnost nápady vyhledávat. K tomuto účelu je pro ně na seznamu nápadů připravené pole pro vyhledávání. Po vyhledání vidí uživatel výsledky podobně jako obyčejný seznam nápadů. Většina uživatelů při vyhledávání záměrně cílí na jeden konkrétní nápad. Z tohoto důvodu není potřeba vytvářet speciální pohledy, ale přidat na seznam pouze jeden nezbytný, který na sobě nemá nastaveny žádné filtry.

Nastavení nápadů

Přes seznam nápadů je také možné dostat se do nastavení nápadů. Protože nástroj toho nenabízí moc, je zde zatím¹ jediná volba. Tou je zapnutí/vypnutí zaslání oznamovacích zpráv nástroje.

Oznámení Stejně jako v celém systému, jsou součástí Nápadů oznamovací zprávy. První důležitým aspektem při jejich návrhu, je určení událostí, při kterých by se oznámení měly zasílat.

V zákaznickém příběhu je řečeno, že oznámení mají být zaslány když:

- K nápadu přibyl komentář.
- Nápad byl rozdělen.
- Nápad byl odstraněn.
- Komentář byl odstraněn.
- Nápad byl přejmenován.
- Nápad byl realizován.

Automaticky se oznámení vytvoří jen prvním případě. V ostatních případech provede zaměstnanec danou akci (standardní funkce SD Protálu) a informuje o tom zákazníky komentářem s doplňujícími informacemi.

Z komentáře je vytvořený oznamovací email. Do předmětu má vloženou značku požadavku (jeho identifikační číslo a krátký řetězec identifikující službu) a název požadavku, což by pro uživatele mělo mít dostatečnou vypovídající hodnotu. V adrese odesílatele je emailová adresa technické podpory. Adresa skutečného odesílatele není zvolena z toho důvodu, že by docházelo k předávání emailových adres, což je oproti jménům, považované za citlivější informace. Obsah zprávy je tvořen odesílatelem, časem kdy byl komentář odeslán, a za ním následujícím zněním komentáře. Na konec zprávy jsou ještě přidané tři doplňující informace:

¹Do budoucna se počítá s rozšířením nastavení.

1. Důvod zaslání oznámení, tzn. že příjemce v minulosti hlasoval pro konkrétní nápad. Název nápadu je ve zprávě jako odkaz. Uživatel se tak může pomocí jednoho kliknutí dostat na detail nápadu a reagovat na komentář.
2. Informaci o tom, že nemá na email odpovídat, ale komentovat přes nástroj Nápady.
3. Návod pro zrušení zasílání pomocí hlasu „Nezajímá mě“ a odkaz do nastavení, kde jde zasílání oznámení vypnout trvale.

Za normálních podmínek jsou jasně daná pravidla, komu se mají oznámení zasílat. Pravidlem pro odběr oznamovacích zpráv, které jsou v režii nástroje, bylo stanoveno hlasování. Uživatel svým hlasem vyjadřuje především to, jak ho daná změna zajímá. V tomto ohledu lze vypustit volbu „Nezajímá mě“, která je předzvěstí uživateli reakce při otevření oznamovací zprávy. Jakmile tedy uživatel zahlasuje „Musím mít“, „Uvítal bych“ nebo „Nežádoucí“, stává se automaticky odběratelem oznámení pro daný nápad.

6.2 Databázové schéma systému Service Desk

Stěženi bod zadání byla integrace do systému Service Desk. Tento fakt přináší nástroji výhody v tom, že lze využít připravených tříd, na druhou stranu databázové schéma nesmí být změněno, což je velkou nevýhodou.

Aby modul správně fungoval, je tedy nejdříve nutné správně rozvrhnout se kterými tabulkami bude pracovat a také jakým způsobem. Databáze ALVAO je celkem rozsáhlá, protože je společná pro všechny vyvíjené aplikace. SD Portál používá jen část schématu a modul, který je tématem této práce operuje jen s několika z nich. Do následující sekce byly vybrány ty, se kterými nástroj přímo či nepřímo pracuje.

tHdTicket

Záznamy v této tabulce jsou požadavky. Každý požadavek má položku pro čas vytvoření, žadatele, řešitele, název, proces, SLA atd. V kontextu nástroje stojí za zmínku především dva cizí klíče. Jedním je *id* služby, na kterou byl požadavek podán, což hraje klíčovou roli v implementovaných dotazech, protože určuje zda požadavek je nebo není na službě nástroje. Druhým je *id* zakládací události, která je z pohledu nápadu jeho popisem.

tAct

Do této tabulky se zaznamenávají všechny události na požadavcích. Jsou zde i zakládací zprávy, vyplněné uživatelem na formuláři pro nový požadavek (nápad). Záznamy tvoří změny položek, emailové zprávy, poznámky atd. U každé události je údaj, ke kterému požadavku patří. Přes pomocnou tabulku *tActKind* se odlišují různé druhy událostí od sebe. Pomocná tabulka se nemění, jsou v ní pouze systémové druhy událostí. Úpravou by mohly vzniknout problémy při aktualizaci celého systému. Z tohoto důvodu v návrhu není zvažovaná její úprava.

Popis nápadu, komentáře a hlasování se stejně jako jiné události ukládá právě do této tabulky.

ProfileValue

Část uživatelských nastavení se ukládá do *ProfileValue*. Každý ze seznamů zobrazujících se v gridu má identifikační řetězec (obvykle podle třídy, která ho implementuje), pod kterým zde má každý uživatel uloženo svoje nastavení jednotlivých pohledů.

ProfileValue není určena výhradně pro ukládání pohledů, proto do ní nástroj ukládá i uživatelské nastavení zasílání oznamovacích zpráv.

tDocument

V *tDocument* se ukládají dokumenty ze všech aplikací. Nápadům slouží hlavně pro ukládání příloh ke komentářům a základacím zprávám. Protože jsou v ní uloženy i citlivá data z aplikace Asset Management, jsou před jakýmkoli čtením dobře testovaná práva dokument vidět.

tHdTicketCust

Jedna z tabulek pro hodnoty vlastních položek. Do *tHdTicketCust* se zapisují hodnoty vlastních položek na požadavcích.

Na formuláři pro nový nápad je vlastní položka *Produkt-modul/aplikace*. Stejně jako pro všechny ostatní vlastní položky, existuje pro ni záznam v tabulce *tColumn*. Jeho *id* je použité jako cizí klíč do tabulky *tColumnValue*, kde se nachází seznam hodnot.

6.3 Návrh nástroje

Service Desk Portál je webová aplikace napsaná v jazyce C# podle návrhového vzoru MVC². Proto je kombinace jazyka C# s MVC jasnou volbou i pro návrh a implementaci nástroje. Aby nemusela být provedena změna v produktu, jsou všechny části nástroje vlastními formuláři.

6.3.1 Datové modely

Podle návrhového vzoru MVC byly vytvořené datové modely. Nejdříve vzniklo několik základních pro každý pohled ze specifikace funkčnosti, následně bylo přidáno několik dalších, jednodušších modelů, které jsou využity při zpracování asynchronních dotazů.

IdeasModel

Model určený pro seznam nápadů. Protože je tento model hned na začátku zdrojového kódu, jsou v něm umístěné statické proměnné, které slouží jako parametry pro fungování nástroje. Normálně by tyto hodnoty byly v tabulce *tProperty*, do které jsou hodnoty přidávány při vytváření nebo upgradu databáze a aplikačně je možné je pouze upravovat. To sice znamená, že by tam údaje nešly vložit pomocí databázového skriptu, ale úpravou by mohly vzniknout problémy při aktualizaci systému. Z toho důvodu byla zvolena parametrizace skrze proměnné.

- Proměnná *IdeasSectionId* slouží pro uchování *id* služby na které nástroj bude fungovat. Hodnota je využívána téměř ve všech dotazech, protože nástroj pracuje pouze s požadavky na vyhrazené službě.
- *ProductColumnId* uchovává *id* vlastní položky *Produkt-modul/aplikace* v *tColumn*, Pomocí této proměnné se získá seznam hodnot, ze kterého si uživatel vybírá při zakládání nového nápadu. Je umístěna sem, protože v testovacích prostředích a v provozu bude pravděpodobně *id* rozdílné.

²Model-View-Controller

- Proměnná *ApplicationPrefix*, ve které je uložen řetězec s adresou serveru, na kterém SD Portál právě běží. Slouží pro správné načítání obrázků v komentářích a v označovacích zprávách. V testovacím prostředí je hodnota nastavená na „localhost“, před nasazením na server je potřeba změnit, podle reálné adresy serveru na kterém aplikace Service Desk běží.

Jako statické proměnné jsou zde i pomocné řetězce, které využívá nástroj při ukládání událostí (viz 6.3.1).

Grid Hlavním obsahem modelu je *IdeasGrid*, což je třída implementující abstraktní třídu *WebGrid*. *WebGrid* je firemní třída určená pro zobrazování různých seznamů. V celém systému je použita na různých místech, nejčastěji na zobrazení požadavků.

První část implementace spočívá v definici metod, které grid používá. V metodě *DeclareColumns* se definují datové sloupce, které se v seznamu budou zobrazovat. Každému sloupci se nastaví jeho zobrazovaný název, typ a název sloupce v dotazu.

V metodě *FormatQuery()* se sestavuje velký dotaz na selekci všech potřebných dat. V seznamu nápadů se kromě standardních údajů uložených v databázi zobrazují i údaje, které je nutné dopočítat (pole *Hodnocení*, *Počet komentářů*, *Počet hlasů*, *Můj hlas*, *Odběr* a *Aktivita za posledních 12 měsíců*). Z tohoto důvodu jsou do dotazu přidány i poddotazy, které potřebné data najdou a hodnoty dopočítají.

Důležitou částí implementace gridu je jeho název. Pod ním se totiž ukládají nastavení pohledů do databáze. Název se vloží přiřazením hodnoty do statické proměnné *PROFILE_NAME*. Ukládání je v abstraktní třídě již implementované a není potřeba ho nijak upravovat. Do databáze se uloží pouze textový řetězec ve formátu XML. V každém pohledu je *id* všech sloupců³, jejich velikost, zda je podle nich řazeno, případně hodnota filtru.

IdeaNewModel

Pro nový nápad je připraven model *IdeaNewModel*. Tento model dědí z třídy určené pro vytváření nových požadavků. Využitím dědičnosti je zařízeno správné zpracování údajů. Uživatel však většinu běžných položek nevidí a ani nepotřebuje. Údaje jsou v tomto případě doplněny automaticky. Služba je daná, protože nástroj běží pouze na jedné (statická proměnná ze seznamu nápadů), autorem (žadatelem) je vždy právě přihlášený uživatel, SLA se nastaví výchozí pro daného uživatele a zbytek polí se nechává nevyplněných, případně je doplní některý ze zaměstnanců před předáním řešiteli.

Při zpracování formuláře se nedříve vytvoří nový požadavek a poté se provedou potřebné úpravy. Ke každému požadavku se při vytvoření automaticky generuje zakládací zpráva do tabulky *tAct*. Forma, kterou to běžný požadavek ukládá není vyhovující a mohla způsobit zmizení popisu nápadu po sloučení nápadů (to se provádí při vzniku duplicit). Proto se po vytvoření požadavku ještě upravuje zakládací zpráva tak, aby odpovídala formě komentáře (více v 6.3.1). Po sloučení (standardní funkce Service Desku) dojde k tomu, že se popis duplicitního nápadu objeví jako komentář u staršího z nápadů. Za popis nápadu se považuje zakládací událost. Při sloučení zůstává zakládací událostí pouzta z požadavku do kterého se slučuje.

Zákazníci mají právo při vyplňování nového nápadu vybrat si svůj výchozí hlas. Ostatní role nemají na formuláři k dispozici pole pro výběr hlasu, takže se hodnota zapisuje jen když není prázdná.

³Id je přiřazeno podle pořadí sloupce v definici.

Pro položku *Produkt-modul/aplikace* se v modelu načítá seznam hodnot z *tColumnValue*.

IdeaDetailModel

Jak již název napovídá, tak tento model slouží pro zobrazení detailu nápadu. Obsahuje v sobě požadavek, který je z pohledu nástroje skrytý za nápadem. Při vytváření se nejdříve načte požadavek a za pomoci načtených údajů se naleznou a dopočítají všechny speciální údaje, které se na detailu nápadu zobrazují (*Hodnocení, Komentáře, Uživatelův hlas*).

Model neslouží pouze pro zobrazení detailů, ale je současně formulářem pro nový komentář. Proto je jeho součástí také pole pro tuto položku a pro přílohy, které uživatel může ke komentáři přidat.

IdeaCommentModel

Tento jednoduchý model je vytvářen při odeslání nového komentáře. V podstatě je v něm uchováno pouze znění komentáře a jeho přílohy.

Protože jsou komentáře běžnými událostmi, tak by se na jejich odlišení (resp. všech speciálních událostí nástroje) velmi hodily vlastní položky na událostech. Pro tabulku *tAct* však k dispozici nejsou a nástroj se musí s odlišením vypořádat jiným způsobem. Na ukládání se tedy používá nejmíň využívaný druh událostí – *Ostatní*. Do názvu události se navíc ukládá řetězec „IdeaComment“, což by mělo zaručit, že nikdo ze zaměstnanců nepřidá nový komentář omylem při vytváření události na požadavku. Zákazníci nemají k přidávání normálních událostí přístup a tak tato situace nepřipadá v úvahu.

S touto podobou komentáře se počítá ve všech modelech a jejich dotazech.

Oznámení Po zápisu komentáře do databáze se automaticky generuje oznámení o komentáři. Přes *id* požadavku se naleznou všichni, kdo hlasovali pro nápad volbami, které slouží jako kritéria pro odběr, tak jak je popsáno v 6.1. Poté se zkontroluje jejich nastavení a těm, kteří mají zasílání zapnuté se oznámení pošle. Vygenerovaná oznamovací zpráva se vkládá do tabulky, kterou na pozadí neustále prochází MailboxReader. Ten zprávu odešle a postará se o její odstranění z tabulky.

UserVoteModel

Pro hlasování, které se zasílá na pozadí je připraven tento malý model. Obsahuje pouze *id* nápadu/požadavku a *id* volby.

Stejně jako u události komentáře je volba uložená v hlavičce události (sloupec *sAct*) jako řetězec s prefixem „IdeaVote#“, za nímž následuje hodnota hlasu. Na hlasy je stejně jako na komentáře využitý druh události *Ostatní*. Řetězce pro jednotlivé hlasy vypadají následovně:

- Musím mít – *IdeaVote#2*
- Uvítal bych – *IdeaVote#1*
- Nezajímá mě – *IdeaVote#0*
- Nežádoucí – *IdeaVote#-1*

IdeasSearchModel

Tento model obsahuje grid stejně jako model pro seznam nápadů. Tento grid je však jednodušší a neumožňuje zobrazit všechny sloupce, které nabízí seznam nápadů. Díky tomu má v definici méně sloupců a jeho dotaz není zdaleka tak složitý.

Zadaný řetězec je vyhledáván v názvech existujících nápadů.

IdeasSettingsModel

Model pro nastavení zahrnuje pouze jednu jedinou položku pro nastavení zasílání oznámení. V případě, že uživatel od nasazení nástroje nastavení nezměnil, jeho záznam není v databázi a volba je zapnutá. Pokud nastavení již změněno bylo, záznam v databázi je. Tento fakt ovlivňuje způsob zpracování formuláře po odeslání. Zasílání oznámení je implicitně zaplé, proto se před zápisem kontroluje zda záznam už existuje. Pokud záznam existuje, pouze se upraví. V opačném případě se do tabulky vloží záznam nový.

6.3.2 Kontrolér

Nástroj má pouze jeden kontrolér, který obsahuje metodu pro každý pohled/stránku nástroje.

Kromě zpracování požadavků na jednotlivé dokumenty, se kontrolér stará také o zpracování asynchronních dotazů a asynchronně zaslaných dat. Těmi jsou zaznamenaní hlasu a hledání podobných nápadů podle názvu.

Za zmínku stojí metoda *Idea*. Ta totiž dokáže přes volitelný URL parametr vynutit vypnutí *custom workflow*. Toho je využito pro zobrazení požadavku běžným způsobem. Předtím, než se tak stane, se kontroluje zda uživatel má na vypnutí potřebná práva (není pouze žadatel).

6.3.3 Grafické uživatelské rozhraní

Návrh grafického uživatelského rozhraní byl vytvořený tak, aby nástroj vzhledem „zapadal“ do systému Service Desk. K tomuto účelu dobře slouží kaskádové styly, které jsou použité ve standardních pohledech.

Návrh díky nim nebyl zaměřen na to, jak budou prvky vypadat, ale především na jejich rozmístění v jednotlivých pohledech. K tomuto účelu byly vytvořeny jednoduché pseudografické náhledy. Jeden z nich je vidět na obrázku [6.2](#).

Seznam nápadů

Samotný seznam se zobrazuje ve firemní komponentě pro grid. Ta umí pracovat s třídou *WebGrid*, proto stačí komponentě předat vlastnost, která má datový typ implementační třídy. Komponenta sama zařizuje vše potřebné, včetně vytváření pohledů/karet s libovolnými filtry a řazením.

Přes kliknutí na jeden z nápadů v seznamu si uživatel zobrazí detail nápadu, který se objeví se na spodu stránky.

Vlevo nad gridem je pro uživatele připraveno tlačítko, které uživatele přenesse na stránku pro nový nápad.

Vprávním horním rohu se nachází vyhledávání. Pokud uživatel toto pole vyplní a potvrdí vyhledání, tak je přesměrován na seznam nalezených nápadů.

Pod vyhledáváním je příkaz *Nastavení nápadů*, který uživatele přenesse na stejnojmenný pohled.

Nový nápad

Pro přidávání nápadů slouží formulář pro nový nápad.

První pole na formuláři je název nápadu. Během toho, co ho uživatel vyplňuje, tak se pod ním plní tabulka pro podobné nápady. Pomocí jQuery se dynamicky vyprazdňuje a naplňuje až do maximální velikosti deseti řádků. Pokud nedošlo k žádné shodě v názvech existujících nápadů, tak uživatel vidí informaci o tom, že je tabulka prázdná. Uživateli vyplňující formulář se detaily nalezených nápadů otevírají na nové kartě, tak aby nedošlo ke ztrátě již vyplněných dat.

Pro popis nápadu je do stránky vložen editor na tvorbu formátovaného textu. Stejně jako na jiných místech v aplikaci je k tomu využít CKeditor⁴, který je open source.

Pod editorem může žadatel (výhradní) vybrat hodnotu pole *Můj hlas*. Protože se nepředpokládá, že by zakládal změnu o kterou nestojí, tak jsou zde k dispozici pouze volby *Musím mít* a *Uvítal bych*.

Na konec formuláře je přidáné pole *Produkt-modul/aplikace*. Seznam hodnot pro tuto položku je tvořen řetězci *ALVAO Asset Management*, *ALVAO Service Desk*, *ALVAO Service Desk Web App* atd. Ne všichni uživatelé budou její hodnotu schopni určit, a tak má pole ve výchozím stavu prázdnou hodnotu.

Detail nápadu

Detailu nápadu nabízí přehled hodnocení, popisu nápadu a také přidávaných komentářů. Přímo na něm jde hlasovat a komentovat.

ZOBRAZIT POŽADAVEK					
Hodnocení 22 Počet hlasů: 18	Hlasové ovládání Produkt-modul/aplikace: ALVAO Service Desk Číslo nápadu: 53676 @ img01.jpg ALVAO by se mělo ovládat hlasem. Naši konzultanti používají service desk a hodilo by se jim vykazovat práci a psát si poznámky už během cesty domů od zákazníka (např. v autě). Proto by se jim hodilo moci ovládat service desk pomocí hlasu.				
Váš hlas: <input checked="" type="radio"/> musím mít <input type="radio"/> uvítal bych <input type="radio"/> nezajímá mě <input type="radio"/> nežádoucí	Nový komentář: [<HTML editor>				
	Přílohy: [Připojit přílohu] [Odeslat komentář]				
	Komentáře (2)				
	<table border="1"><tr><td>[fotka] Josef Plevač [ikona žadatele] img01.png (26 kB) Tohle bych určitě uvítal, ovládat hlasem by se mi hodilo. Přidávám screenshot, kde by se zapnutí této volby líbilo.</td><td>24. 12. 2015 19:05</td></tr><tr><td>[fotka] Petr Suchý Ano, tahle funkcionlita se mi zamlouvá.</td><td>24. 12. 2015 19:00</td></tr></table>	[fotka] Josef Plevač [ikona žadatele] img01.png (26 kB) Tohle bych určitě uvítal, ovládat hlasem by se mi hodilo. Přidávám screenshot, kde by se zapnutí této volby líbilo.	24. 12. 2015 19:05	[fotka] Petr Suchý Ano, tahle funkcionlita se mi zamlouvá.	24. 12. 2015 19:00
[fotka] Josef Plevač [ikona žadatele] img01.png (26 kB) Tohle bych určitě uvítal, ovládat hlasem by se mi hodilo. Přidávám screenshot, kde by se zapnutí této volby líbilo.	24. 12. 2015 19:05				
[fotka] Petr Suchý Ano, tahle funkcionlita se mi zamlouvá.	24. 12. 2015 19:00				

Obrázek 6.2: Pseudografický návrh detailu nápadu

Pro uživatele s právem hlasovat se na levé straně nachází radio tlačítka, na kterých jsou vidět jednotlivé volby hlasování. Stránka je za pomoci CSS tříd z bootstrapu responzivní,

⁴<http://ckeditor.com/>

tak jak se při úzkém zobrazení hodnocení přesouvá mezi názvem a popisem nápadu.

Pro zobrazení požadavku běžným způsobem, se pro všechny role, které nejsou pouze žadatelem, zobrazuje v pravém horním rohu příkaz *Zobrazit požadavek*, který přepne vizualizaci do běžné podoby.

Pokud je nápad (požadavek) uzavřený, tak se místo radio tlačítek zobrazuje popis „Uzavřeno“.

Vyhledávání nápadů

V pohledu vyhledávání se stejně jako na seznamu nápadů zobrazuje grid, ve kterém jsou nalezené požadavky. Je však značně jednodušší a pro uživatele je zde připraven pouze jeden výchozí pohled s názvem „základní“. Detail se stejně jako na seznamu zobrazí na spodní stránce po kliknutí na konkrétní nápad.

Nastavení nápadů

Nastavení je velmi jednoduché. Nachází se zde pouze jeden checkbox, kterým se zapíná/vypíná zaslání oznámení o nových komentářích. Nic dalšího tento pohled nenabízí.

6.3.4 Konfigurace systému

V této podkapitole jsou popsány nastavení, které je v SD Portálu potřeba provést, aby nástroj správně fungoval.

Nová služba

Pro nástroj je nejvýhodnější vyhradit zcela novou službu *Vývoj produktů/Nápady*. Služba bude určena výhradně na požadavky, se kterými bude rozšiřující modul pracovat.

Na službě je nastaveno vlastní workflow (*Custom/ALVAO/Ideas/Idea?id={0}*), díky kterému se požadavky této služby zobrazují vždy jako nápady.

Na službě jsou zapnuty standardní oznámení pro žadatele i operátory při vytvoření požadavku i při jejich vyřešení. Po založení nápadu přijde žadateli standardní oznamovací zpráva. Přestože v ní je odkaz na standardní detail požadavku, tak díky vlastní workflow dojde k přesměrování a při pokusu o zobrazení uživatel uvidí požadavek jako nápad.

Na službě je nastaven vlastní formulář pro podání nového požadavku (*Custom/ALVAO/Ideas/IdeaNew*). Díky tomu uživatelé podávají na službu nové nápady i když k formuláři nepřistupují přes Nápady.

SLA je nastavené pro všechny, takže podávat nové nápady mohou všichni.

6.4 Implementace

Implementace proběhla ve vývojovém prostředí Visual Studio. Podle návrhu byly vytvořeny všechny modely, kontrolér i pohledy.

Pro úpravy vzhledu byl ještě následně přidán soubor s kaskádovými styly, který upravuje vzhled podle potřeb zadavatele.

6.4.1 Testování během implementace

Součástí implementace byla průběžná kontrola, zda se hodnoty do databáze zapisují tak, jak mají.

Zprvu bylo testované, zda se dobře vytváří nové nápady. Při této činnosti byla nalezena chyba. Při vytváření nápadu za uživatele, který měl na službě roli řešitele, mu byl požadavek automaticky předán k řešení, protože je tato volba implicitně zapnutá. Řešením bylo v konstruktoru modelu předání řešiteli vypnout.

V průběhu implementace také proběhl test oznámení. V této funkčnosti byla také nalezena chyba. V oznámení došel pouze komentář a nebyly v něm informace o odesílateli a času odeslání. Pro průzkumu, proč se tak děje, bylo zjištěno, že z komentáře se nejdříve vytvořila celá html stránka, ze které následně bylo vyextrahováno tělo, doplněny potřebné informace a tento obsah se vložil zpět do html stránky jenž představovala oznamovací zprávu. Ke špatnému zobrazení obsahu zprávy docházelo z toho důvodu, extrakce těla dokumentu nebyla naimplementována spolehlivě. Tato chyba byla vyřešena tak, že se komentář nejdříve vloží do oznamovací zprávy a až ve chvíli, kdy je celá zpráva hotová, se přidají html tagy.

V průběhu implementace byly provedeny testy zaměřené na rychlost selekce dat z databáze. Nejsložitější dotaz se nachází v metodě *FormatQuery* v modelu pro seznam nápadů. Přímo v dotazu se dopočítávají hodnoty do polí *Hodnocení*, *Počet komentářů*, *Počet hlasů* a *Aktivita za posledních 12 měsíců*. Z toho důvodu je celkem rozsáhlý. Pomocí SqlExpressProfileru byla měřena doba zpracování dotazu SQL Serverem. Výsledky jsou umístěny do následujících tabulek. První řádek v klidovém stavu databáze. Druhý je měřen při lehké zátěži.

Start time	End time	Duration (ms)
08:16:53.703	08:16:53.720	18
08:22:04.800	08:22:04.820	20

Tabulka 6.1: Testování na malé databázi

Start time	End time	Duration (ms)
08:18:26.940	08:18:27.273	333
08:26:49.697	08:26:50.043	347

Tabulka 6.2: Testování na kopii interní databáze

Z výsledků je patrné, že SQL Server nemá problém s rychlostí zpracování tohoto dotazu.

6.4.2 Uživatelské příručky

Do systému Service Desk Portál jsou umístěné příručky pro uživatele a pro pracovníky. Jejich kompletní znění je možné najít v příloze C. Příručky jsou v systému umístěné do *Báze znalostí*⁵, která je pevnou součástí systému.

Příručky jsou návody pro užívání a správu nástroje. Uživatelská příručka je zaměřená především na používání nástroje. Je v ní návod, jak vytvářet nápady, jak hlasovat a k čemu je hlasování užitečné. Druhá z příruček je vytvořená pro pracovníky ALVAO, kteří se starají o správu nástroje. V ní je popsán pozadí nápadů – jak události nápadů vypadají v deníku, jak za pomoci komentářů informovat zákazníky, co dělat při vzniku duplicit atd.

⁵Knowledge base z ITIL

6.4.3 Nasazení do provozu

Před nasazením do provozu bylo potřeba nejprve nahrát vlastní formuláře na server do vyhrazené složky, a poté upravit hodnoty konfiguračních proměnných na začátku zdrojových souborů.

Společnost ALVAO vyžadovala termín nasazení do 12. května, kdy pořádala workshop se zákazníky. Nasazení do interního provozu proběhlo 5. května 2016. Po nasazení byl nalezen vizuální problém. Na celém nástroji bylo špatné odsazení obsahu a postraní posuvníky díky tomu nebyly na krajích stránky. Seznam nápadů se navíc při zobrazení detailu nápadu posouval směrem doleva.

Po krátkém zkoumání byla zjištěna příčina. Implementace proběhla na nejnovější, toho času nevydané verzi systému, která na serveru ještě nebyla k dispozici. Od vydání verze, která byla na serveru, došlo pravděpodobně k upravení používaných kaskádových stylů a tím pádem ke vzniku zmiňovaných chyb. K opravě stačilo předefinovat styly, které odsazení způsobovaly.

Chybu se tak podařilo opravit rychle, ještě před workshopem a nástroj díky tomu mohl být v rámci workshopu prezentován.

Kapitola 7

Závěr

Cílem práce bylo vytvořit ve stávajícím informačním systému prostor pro zákazníky firmy ALVAO, kde mohou vyjádřit svoje nápady na zlepšení systému a navzájem si tyto nápady hodnotit.

Pro správné pochopení, jak systém Service Desk funguje, byly nejdříve provedeny průzkumy a studie všech technologií, na kterých je postaven. Část tohoto průzkumu byla věnována i terminologii prostředků, které jsou součástí systému.

V jedné z kapitol byl proveden rozbor architektury, popis prostředků a také byly představeny uživatelské role.

V provedné analýze byly stanoveny aspekty, které bylo potřeba dodržet při návrhu a implementaci.

Na provedených konzultacích byly osvětleny všechny specifické požadavky, díky kterým následně vznikl celkem podrobný návrh nástroje. Podle návrhu byla provedena implementace v níž se podařilo splnit všechny požadované funkce a také dodržet všechny důležité aspekty z analýzy.

Díky tomu je výsledný nástroj implementován tak, aby běžel nad standardními prostředky systému, ale dokázal je prezentovat jiným způsobem, než jak je tomu obvykle. Modul tak přidává systému novou funkčnost a zároveň neomezuje to, co SD Portál standardně nabízí. Velkým úspěchem je fakt, že se implementace obešla bez změn v produkční verzi systému.

V závěru práce byly sepsány příručky pro uživatele a správce. Také díky nim mohl být nakonec nástroj uveden do provozu.

Literatura

- [1] ITIL – výkladový slovník a zkratky v češtině. [Online], Naposledy navštíveno 17. 5. 2016.
URL http://itsmf.cz/wp-content/uploads/2013/12/itil_2011_czech_glossary_v2.0.pdf
- [2] ALVAO Documentation Library: Co to je ALVAO. [Online], Naposledy navštíveno 17. 5. 2016.
URL http://doc.alvao.com/support/doc/cs/alvao_8_2/default.aspx
- [3] ALVAO Documentation Library: Co to je ALVAO Service Desk. [Online], Naposledy navštíveno 17. 5. 2016.
URL http://doc.alvao.com/support/doc/cs/alvao_8_2/alvao_service_desk/introduction/default.aspx
- [4] ALVAO Documentation Library: Implementace systému v organizaci. [Online], Naposledy navštíveno 17. 5. 2016.
URL http://doc.alvao.com/support/doc/cs/alvao_8_2/alvao_service_desk/implementation/default.aspx
- [5] MacDonald, M.: *ASP.NET 4 a C# 2010*. Brno: Zoner Press, 2011, ISBN 978-80-7413-131-8.
- [6] RYCHLÝ, M.: Organizace předmětu INI a úvod do ITIL. [Online], Naposledy navštíveno 17. 5. 2016.
URL <https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/INI-IT/lectures/INI01--intro-to-itil.print.pdf>
- [7] Wikipedia: .NET Framework version history. [Online], Naposledy navštíveno 17. 5. 2016.
URL https://en.wikipedia.org/wiki/.NET_Framework

Přílohy

Seznam příloh

A	Obsah CD	35
B	Uživatelské příběhy	36
	B.1 Uživatelský příběh zákazníka	36
	B.2 Uživatelský příběh vývojáře ALVAO	37
C	Uživatelské příručky	38
	C.1 Nápady – Příručka pro uživatele	38
	C.2 Nápady – Správa	40
D	Výsledné snímky nástroje	42

Příloha A

Obsah CD

- src/ – adresář se zdrojovými soubory nástroje
- doc/
 - tex/ – adresář se zdrojovými soubory pro latex
 - xkaliv06.pdf – tato práce ve formátu pdf

Příloha B

Uživatelské příběhy

B.1 Uživatelský příběh zákazníka

1. Zákazník si koupí produkt ALVAO a obdrží přihlašovací údaje do portálu technické podpory (SD WA) s informací o možnosti podávat nápady a hlasovat pro ně (funkčnost **Nápady**).

Jde na portál TP do Nápadů, přečte si seznam odebíraných/vlastních/nejlépe hodnocených/nejaktivnějších/nejnovějších nápadů, případně včetně diskuzí. Hlasuje pro ně. Vkládá komentáře.

Má vlastní nápad. Hledá, jestli už nápad existuje. Pokud ho najde, přečte si jeho obsah. Pokud nápad odpovídá tomu, co zákazník chce, tak pro něj hlasuje. Případně vkládá komentář.

Pokud nenajde odpovídající nápad, tak začne vytvářet nový. Během vyplňování názvu nápadu vidí podobné nápady. Prochází nabízené nápady. Pokud některý odpovídá tomu, co zákazník chce, tak pro něj hlasuje.

Pokud nenajde odpovídající nápad, vytvoří nový a rovnou pro něj hlasuje.

2. Zákazník obdrží oznámení o tom, že někdo přidal komentář k nápadu, který ho zajímá. Případně reaguje dalším komentářem.
3. Zákazník vypne oznámení a místo toho si čas od času projde nápady, které ho zajímají, přečte si komentáře. Zjistí, co je nového a případně reaguje hlasováním (případně zruší svůj hlas) nebo komentářem.
4. Zákazník obdrží oznámení o tom, že nápad, který ho zajímal, byl realizován. Na základě toho updatuje ALVAO.

Doplňující informace

Zákazník je informován o tom, že:

- K nápadu přibyl komentář.
- Nápad byl rozdělen – provádí vývojář/konzultant ručně.
- Nápad byl odstraněn – provádí vývojář/konzultant ručně.
- Komentář byl odstraněn – provádí vývojář/konzultant ručně.

- Nápad byl přejmenován – provádí vývojář/konzultant ručně.
- Nápad byl realizován – provádí vývojář/konzultant ručně.

B.2 Uživatelský příběh vývojáře ALVAO

1. Zákazník vytvoří nový nápad nebo přidá komentář a vývojář o tom dostane oznámení. Reaguje komentářem nebo jinou akcí:
 - Rozdělení nápadu.
 - Odstranění nápadu.
 - Odstranění komentáře.
 - Přejmenováním nápadu.
2. Je vydaná nová verze ALVAO. Pomocí komentářů jsou zákazníci informováni, že nápady, které je zajímaly, byly realizovány.
3. Zákazník podá zlepšovací návrh na Technickou podporu. Pracovník TP vidí, že navrhovanou změnu by mohli využít i další zákazníci. Navrhne zákazníkovi místo zlepšovacího návrhu vytvořit nápad, aby pro něj mohli hlasovat i ostatní zákazníci.

Doplňující informace

Vývojář může dělat stejné věci jako zákazník, kromě hlasování.

Příloha C

Uživatelské příručky

C.1 Nápady – Příručka pro uživatele

Nápady jsou místem, kde můžete prezentovat vaše vlastní návrhy na změny v produktech ALVAO, hlasovat o jejich prioritě a také o nich diskutovat. Vaše nápady jsou sdíleny s ostatními zákazníky, proto není vhodné uvádět v nich citlivé informace. Nejžádanější změny se snažíme realizovat v budoucích verzích produktů ALVAO.

Seznam nápadů

V hlavní nabídce klikněte na příkaz **Nápady**, který zobrazí seznam nápadů. Na stránce můžete přepínat pohledy:

- Odebírané – nápady, u kterých odebíráte nové komentáře, tzn. pro které jste hlasovali volbou „Musím mít“, „Uvítal bych“ nebo „Nežádoucí“. Viz **Hlasování a komentování**.
- Mé – Vaše vlastní nápady.
- Nejlépe hodnocené – nápady, které mají nejvyšší hodnocení.
- Nejaktivnější – nápady, ke kterým za posledních 12 měsíců přibylo nejvíce komentářů a hlasů.
- Nejnovější – nově podané nápady.

Po kliknutí na konkrétní nápad se na spodu stránky zobrazí jeho detail.

Vyhledávání nápadů

Do vyhledávacího řádku vpravo nahoře zadejte číslo nápadu nebo část jeho názvu. Nalezené nápady se zobrazí v tabulce. Po kliknutí na konkrétní nápad se na spodu stránky zobrazí jeho detail.

Hlasování a komentování

V seznamu nápadů klikněte na konkrétní nápad. Ve spodní části stránky se zobrazí detail nápadu. V jeho levé části můžete hlasovat volbami:

- „Musím mít“ – změna je pro vás velmi důležitá.
- „Uvítal bych“ – změna je pro vás užitečná, ale není nezbytně nutná.
- „Nezajímá mě“ – nápad pro vás nemá žádnou hodnotu.
- „Nežádoucí“ – změnu si nepřejete, vadila by vám.

Každý z hlasů má váhu (Musím mít: 2, Uvítal bych: 1, Nezajímá mě: 0, Nežádoucí: -1). Výsledné hodnocení (hlasování) je pak spočítáno jako součet vah jednotlivých hlasů.

Hlasováním „Musím mít“, „Uvítal bych“ nebo „Nežádoucí“ se zároveň přihlásíte k odběru oznámení o nových komentářích k danému nápadu. Viz **Nastavení oznámení**.

Poznámka: Po uzavření nápadu už není možné pro něj hlasovat.

V detailu nápadu jsou zobrazeny komentáře od ostatních uživatelů. Vlastní komentář můžete napsat do vstupního pole nad seznamem komentářů a případně připojit i přílohy.

Vytvoření nápadu

1. Pro vytvoření nápadu přejděte na hlavní stránku Nápadů.
2. Klikněte na tlačítko **Nový nápad**.
3. Otevře se formulář pro vytvoření nápadu.
4. Zadejte název nápadu. Během vyplňování se pod vstupním polem zobrazují již existující nápady s podobným názvem.
5. Pokud stejný nápad již existuje, klikněte na něj a hlasujte, případně napište svůj komentář.
6. Pokud nápad neexistuje, vyplňte **Popis** nápadu.
7. K nápadu také můžete přidat **Přílohy**, např. screenshot aplikace s popisem zamýšlené změny.
8. Podle toho jakou má pro vás nápad prioritu, zvolte příslušnou volbu v poli **Můj hlas**.
9. Vyberte **Produkt-modul/aplikaci** podle aplikace, do které je změna určena.
10. Vytvoření nápadu potvrďte tlačítkem **OK**.

Nastavení oznámení

V seznamu nápadů klikněte vpravo nahoře na příkaz *Nastavení oznámení*. Pokud je volba **Zasílat oznámení o nových komentářích k odebíraným nápadům** zapnutá, systém vám automaticky pošle oznámení v okamžiku, kdy někdo přidá komentář k nápadu, který odebíráte, tzn. hlasoval/a jste pro něj volbou „Musím mít“, „Uvítal bych“ nebo „Nežádoucí“. Pokud tuto volbu vypnete, nebudou vám chodit tato oznámení o nových komentářích.

Tip: Pokud nechcete z určitého nápadu dostávat oznámení, hlasujte pro něj volbou „Nezajímá mě“.

C.2 Nápady – Správa

Nápady jsou požadavky ze služby *Vývoj produktů/Nápady*.

Poznámka: Je důležité přidávat oznamovací komentáře přes WA – Nápady, aby se o tom dozvěděli i ostatní zákazníci (odběratelé).

Události nápadů

Všechny události, které nástroj Nápady používá, jsou druhu *Ostatní*.

Hlasování

Hlasování je zaznamenáno jako událost v deníku požadavku s předmětem „IdeaVote#...“. Za znakem „#“ následuje číslo označující váhu hlasu.

- „Musím mít“ – „IdeaVote#2“
- „Uvítal bych“ – „IdeaVote#1“
- „Nezajímá mě“ – „IdeaVote#0“
- „Nežádoucí“ – „IdeaVote#-1“

Komentáře

Komentáře přidané přes nástroj Nápady poznáte od ostatních událostí v deníku tak, že mají jako předmět události uveden „IdeaComment“.

Přejmenování nápadu

Pokud název nápadu neodpovídá jeho obsahu, vhodně ho přejmenujte. Komentářem o tom informujte zákazníky:

Nápad byl přejmenován z <> na <>.

Rozdělení nápadu

Pokud nápad obsahuje více než jednu navrhovanou změnu, rozdělte ho. Tzn., pro každou dílčí změnu vytvořte za žadatele samostatný požadavek ve službě *Vývoj produktů/Nápady*, přidejte vazbu na původní požadavek (nápad) typu „následuje po“ a zapněte na něm volbu Nápad. Do každého požadavku запиšte za zákazníka novou událost druhu *Ostatní* a do předmětu zkopírujte původní hlas (IdeaVote#...). Původní požadavek uzavřete, aby pro daný nápad již nebylo možné hlasovat.

O rozdělení informujte pomocí komentáře k původnímu nápadu:

Nápad byl rozdělen do těchto dílčích nápadů:

<URL nově vzniklého nápadu č.1>

<URL nově vzniklého nápadu č.2>

Odstranění nápadu

Pokud zjistíte, že obsahem nápadu není návrh na změnu (například popis chyby), přesuňte ho do správné služby, případně ho odstraňte a emailem o tom informujte zákazníka:

Dobrý den,

váš nápad byl odstraněn, protože ...

S pozdravem

Tým Technické podpory ALVAO

Odstranění komentáře

V případě že některý z komentářů obsahuje vulgaritu, napadá některého z uživatelů nebo vůbec nesouvisí s daným nápadem, odstraňte ho a případně vložte za autora cenzurovanou verzi s upozorněním „Cenzurováno“.

Realizace nápadu

Pokud byl nápad realizován, uzavřete ho. Informujte o tom zákazníky pomocí komentáře: Tento nápad byl realizován ve verzi <číslo verze>, kterou můžete najít na stránce: <odkaz na novou verzi>.

Příloha D

Výsledné snímky nástroje

The screenshot displays the ALVAO web interface. At the top, a blue navigation bar contains the ALVAO logo and menu items: Office 365, Požadavky, Katalog služeb, Báze znalostí, Aktuální zprávy, and Nápady. The main content area features a suggestion titled "Seznam notebooků bez zapnutého Bitlockeru" (List of laptops without BitLocker enabled). The suggestion includes the following text: "Příběh: Firemní bezpečnostní směrnice nařizuje, aby všechny notebooky měly zapnutý BitLocker na všech pevných discích. Technik má povinnost to pravidelně kontrolovat. Asset Management by mohl na základě detekce sestavit seznam notebooků, které porušují tuto směrnici." To the right of the title is a "ZOBRAZIT POŽADAVEK" button. On the left, there is a "Hodnocení" (Rating) section with a score of 1 and a "Počet hlasů: 2" (Number of votes: 2). Below this is a "Váš hlas:" (Your vote) section with radio buttons for "Musím mít" (I must have), "Uvítal bych" (I would like), "Nezajímá mě" (I'm not interested), and "Nežádoucí" (Unwanted). A "Nový komentář:" (New comment) section contains a rich text editor with a toolbar and a large empty text area. Below the editor is a "Přílohy:" (Attachments) section with a "Připojit přílohu" (Attach file) button and an "Odeslat komentář" (Send comment) button. At the bottom, it shows "Komentáře (0)" (Comments (0)).

Obrázek D.1: Detail nápadu



Nový nápad

Název: *

Podobné nápady

Hodnocení Název nápadu

Můj hlas

Produkt-modul/aplikace

Autor

Číslo nápadu Stav

Tabulka je prázdná

Popis:

Rich text editor toolbar with icons for bold, italic, underline, strikethrough, bulleted list, numbered list, link, unlink, text color, background color, font face, font size, text color, link, unlink, and image. Below the toolbar is a large empty text area.

Přílohy:

Připojit přílohu

Můj hlas: *

Produkt-modul/aplikace:

OK

Obrázek D.2: Nový nápad