

České vysoké učení technické v Praze  
Fakulta elektrotechnická

katedra počítačové grafiky a interakce

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Student: **Bc. Pavel Volek**

Studijní program: Otevřená informatika (magisterský)  
Obor: Softwarové inženýrství

Název tématu: **Informační systém SYBA Shared Base**

Pokyny pro vypracování:

1. Analyzujte požadavky zadavatele na systém pro evidenci a správu obchodních kontaktů.
2. Navrhněte webovou aplikaci splňující požadavky dle bodu 1.
3. Aplikaci implementujte v prostředí ASP.NET MVC.
4. Vypracujte metodiku pro provádění jednotkových a integračních testů. V co největším rozsahu se zaměřte na automatizaci testování.
5. Systém ověřte s využitím vypracované metodiky a metodiku zhodnoťte.


Seznam odborné literatury:

Kuaté P.H., Harris T., Bauer Ch., King G.: NHibernate in Action, Manning Publications Co., 2009, ISBN 978-1-932394-92-4

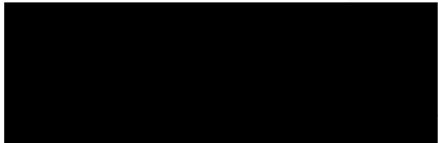
Arlow J., Neustadt I.: UML 2 a unifikovaný proces vývoje aplikací, Computer Press, 2011, ISBN 978-80-251-1503-9

Vedoucí: Ing. Jakub Ďuriga

Platnost zadání: do konce letního semestru 2014/2015

  
prof. Ing. Jiří Zára, CSc.  
vedoucí katedry



  
prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 24. 2. 2014

České vysoké učení technické v Praze  
Fakulta elektrotechnická  
Katedra počítačů



Diplomová práce  
**Informační systém SYBA SharedBase**

*Bc. Pavel Volek*

Vedoucí práce: Ing. Jakub Ďuriga

Studijní program: Softwarové inženýrství

Obor: Otevřená informatika

2015



## **Poděkování**

Na tomto místě bych chtěl poděkovat vedoucímu diplomové práce Ing. Jakubu Ďurigovi za metodické vedení a věcné připomínky při zpracovávání diplomové práce.

Dále bych rád poděkoval rodičům za podporu, kterou mi poskytují v průběhu celého studia.





## **Prohlášení**

Prohlašuji, že jsem práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze 5. 1. 2015

---

Pavel Volek



## **Abstrakt**

Tématem této diplomové práce je analýza, návrh, implementace a testování systému SYBA SharedBase pro společnost ČESKÁ OBALOVÁ ASOCIACE SYBA, s.r.o. Jedná se o webovou aplikaci na správu kontaktních údajů osob a společností, která bude nahrazovat dosavadní řešení používané v organizaci zadavatele.

## **Abstract**

The subject of this diploma thesis is the analysis, design, implementation and testing of the system SYBA SharedBase for the company ČESKÁ OBALOVÁ ASOCIACE SYBA, s.r.o. SYBA SharedBase is a web application for a management of contact details of individuals and companies that will replace the existing solutions used in the customer's organization.



# Obsah

Úvod.....	1
Struktura práce .....	1
Třetí kapitola .....	1
Poslední kapitola .....	1
Analýza .....	3
Vize a rozsah projektu .....	3
Charakteristika zadavatele .....	3
Podnikatelská příležitost a potřeby zadavatele.....	3
Návrh řešení .....	4
Rozsah a omezení.....	4
Specifikace požadavků .....	6
Úvod ...	6
Předmět specifikace .....	6
Obecný popis.....	7
Funkce systému.....	8
Datový model .....	41
Požadavky na vnější rozhraní .....	43
Návrh.....	45
Volba implementačního prostředí .....	45
Serverová část .....	45
Uživatelské rozhraní .....	45
Návrhový model .....	46
Architektura systému .....	46
Model nasazení .....	48
Návrh komponent .....	49
Testování.....	63
Úlohy a cíle .....	63
Řízení chyb .....	63
Spouštění testů .....	63
Regresní testování.....	64
Strategie testování .....	64
Rozsah .	64
Jednotkové testování .....	64
Integrační a systémové testování.....	67
Akceptační testování .....	68

Závěr .....	71
Literatura .....	73
Příloha A: Zkratky a akronymy .....	I
Příloha B: Návrh uživatelského rozhraní .....	III
Příloha C: Snímky obrazovky SYBA SharedBase .....	VII
Příloha D: Snímky obrazovky Katalogu dodavatelů .....	XIII
Příloha E: Snímky obrazovky SYBA SharedBase API .....	XV
Příloha F: Příklady testů .....	XVII
Příloha G: Obsah přiloženého CD .....	XXI

## Seznam obrázků

Obrázek 1 – Kontextový diagram systému SYBA SharedBase.....	7
Obrázek 2 – PU: Správa kontaktů .....	9
Obrázek 3 – PU: Správa skupin kontaktů .....	15
Obrázek 4 – PU: Správa společností.....	19
Obrázek 5 – PU: Správa kategorií společností.....	25
Obrázek 6 – PU: Odesílání hromadné korespondence .....	30
Obrázek 7 – PU: Synchronizace se službou Google Contacts.....	35
Obrázek 8 – PU: Správa společností.....	38
Obrázek 9 – Diagram třívrstvé architektury.....	47
Obrázek 11 – Diagram nasazení.....	49
Obrázek 12 – SYBA SharedBase Domain: Přehled balíčků .....	49
Obrázek 13 – SYBA SharedBase Domain: Balíček Entities.....	50
Obrázek 14 – SYBA SharedBase Domain: Balíček Enums.....	50
Obrázek 15 – SYBA SharedBase Data NHibernate: Přehled balíčků.....	52
Obrázek 16 – SYBA SharedBase Data NHiberante: Balíček Mappings .....	52
Obrázek 17 – SYBA SharedBase Data NHibernate: Balíček Repositories .....	53
Obrázek 18 – SYBA SharedBase Data NHibernate: Balíček Membership .....	53
Obrázek 19 – SYBA SharedBase Services: Přehled balíčků.....	54
Obrázek 20 - SYBA SharedBase Services: Balíček Services.....	55
Obrázek 21 – SYBA SharedBase Services: Balíček Repositories .....	55
Obrázek 22 – SYBA SharedBase Services: Balíček Helpers.....	56
Obrázek 23 –SYBA SharedBase Google: Přehled komponenty .....	56
Obrázek 24 – SYBA SharedBase API: Přehled balíčků .....	57
Obrázek 25 - SYBA SharedBase API: Balíček Controllers.....	58
Obrázek 26 – SYBA SharedBase API: Balíček Models.....	59
Obrázek 27 – SYBA SharedBase API: Balíček Providers.....	60
Obrázek 28 - SYBA SharedBase Frontend: Přehled balíčků.....	61





# Úvod

Tématem této diplomové práce je problematika informačního systému ČESKÉ OBALOVÉ ASOCIACE SYBA. Společnost ČESKÁ OBALOVÁ ASOCIACE SYBA s.r.o. sdružuje a prosazuje zájmy subjektů činných v sektoru balení. Společnost ke své činnosti využívá zastaralé informační systémy, časově náročné procesy a nadále nevyhovující nástroje. Aby společnost mohla svým klientům dodávat kvalitní služby, potřebuje kvalitní informační systém, který nahradí zastaralé a nevyhovující nástroje a zjednoduší zaběhlé pracovní postupy.

Hlavním cílem práce je analýza potřeb zadavatele, specifikace požadavků na nový informační systém SYBA SharedBase, návrh a implementace první verze systému. SYBA SharedBase bude webová aplikace nahrazující informační systémy a nástroje používané zadavatelem ke správě kontaktních údajů osob a společností, rozesílání hromadných e-mailů a autentizaci klientů v informačních systémech zadavatele.

Druhým cílem práce je vytvoření metodiky testování, navržené na míru vyvíjenému systému a zaměřené především na jednotkové a integrační testování, a následná aplikace této metodiky.

## Struktura práce

Práce je strukturována do tří hlavních logických celků – Analýza, Návrh a Testování.

Kapitola Analýza obsahuje analytickou dokumentaci projektu SYBA SharedBase. Tato dokumentace slouží k představení podnikatelské příležitosti, seznámení s potřebami zadavatele a návrhu řešení problému z podnikatelského hlediska. Nedílnou součástí je specifikace požadavků, dokument popisující požadované funkce vyvíjeného systému.

Druhá kapitola Návrh obsahuje návrhovou dokumentaci systému. Tato dokumentace obsahuje architektonický návrh systému jak celku, tak i jeho jednotlivých subsystémů. Popisuje architekturu systému, vzájemnou spolupráci vnitřních i vnějších komponent a datový model.

Třetí kapitola Testování obsahuje testovací dokumentaci systému SYBA SharedBase verze 1.0, analyzovaného a navrženého v předešlých kapitolách. V dokumentu je navržena metodika testování odpovídající rozsahu systému a požadavkům zadavatele.

Poslední kapitola Závěr obsahuje zhodnocení práce a jsou v ní diskutovány dosažené cíle.



# Analýza

Tato kapitola obsahuje souhrnnou analýzu potřeb zadavatele. Obsah analytické dokumentace je rozdělen do dvou hlavních sekcí:

1. **Dokumentace vize a rozsahu projektu** – dokumentuje kontext problému a cíle projektu vycházející z podnikatelských požadavků.
2. **Specifikace funkčních požadavků** – úplný popis systému, který má být vyvinut. Má formu definice případů užití a uživatelských, funkčních a nefunkčních požadavků na systém, které z podnikatelských požadavků plynou.

## Vize a rozsah projektu

### Charakteristika zadavatele

ČESKÁ OBALOVÁ ASOCIACE SYBA, s. r. o. (dále jen *zadavatel*), sdružuje a prosazuje zájmy subjektů činných v sektoru balení, obalů, obalových prostředků, balicích materiálů a strojů a také zájmy výrobců baleného zboží. Asociace je otevřena pro všechny zájemce, zejména z řad výrobců a dodavatelů obalové techniky i baleného zboží, a také pro poskytovatele balicích služeb či vzdělání v oblasti obalů a balicích procesů (ČESKÁ OBALOVÁ ASOCIACE SYBA, 2011).

Mezi hlavní aktivity asociace patří informační servis, konzultace, poradenství, pořádání seminářů a konferencí a vydávání elektronického oběžníku CS Packaging News.

### Podnikatelská příležitost a potřeby zadavatele

Důležitým aspektem práce zaměstnanců asociace SYBA je komunikace s členskými organizacemi a poskytování informací o společnostech činných v podnikatelském oboru asociace. Pro podporu těchto činností je nutné spravovat základní a kontaktní informace o společnostech a udržovat je aktuální. V současné chvíli jsou kontakty a informace o firmách drženy na několika místech:

- a) v souborech programu Microsoft Excel<sup>1</sup> (seznamy kontaktů pro e-mailové konference, seznamy společností, které jsou členy asociace),
- b) v souborech HTML na webových stránkách společnosti, které obsahují seznam členů asociace,
- c) v souborech HTML na webových stránkách společnosti, které obsahují seznamy dodavatelů obalů a obalových prostředků,
- d) v Google Kontaktech na jednotlivých Gmailových<sup>2</sup> účtech zaměstnanců,
- e) ve webové aplikaci Mailer<sup>3</sup> společnosti Inicia,
- f) v databázi přístupových údajů ke složkám na webovém serveru společnosti Inicia.

<sup>1</sup> Tabulkový procesor firmy Microsoft.

<sup>2</sup> E-mailová služba firmy Google.

<sup>3</sup> Systém na rozesílání hromadné e-mailové korespondence.

Tento způsob udržování informací je nevhodný z několika hledisek. Zprvce jsou informace o společnostech roztříštěny a duplikovány na několika místech, což má za následek jejich nekonzistenci. Taková data jsou často změněna pouze na jednom místě, zatímco v jiných dokumentech zůstávají zastaralé údaje. Zaměstnanci zodpovědní za správu údajů musí přesně vědět, v jakých souborech či systémech jsou udržovány jaké informace, a v případě aktualizace musí upravovat údaje na všech odpovídajících místech. Dalším problémem je rozmanitost typů souborů a systémů, ve kterých jsou informace udržovány. Zaměstnanci musí často kvůli provedení aktualizace informací provádět změnu zdrojového kódu technických dokumentů (soubory HTML), ovládat rozhraní pro úpravu údajů v používaných softwarových nástrojích (Mailer, G-mail) nebo kontaktovat dodavatele externích služeb se žádostí o opravu údajů (Inicia).

Společnost SYBA potřebuje systém pro jednotnou centrální úpravu údajů o osobách a společnostech. Tento systém umožní spravovat veškeré informace na jednom místě a jejich následnou distribuci do externích softwarových nástrojů. Takový systém zaměstnancům ušetří čas a navíc odstraní problém se vznikem nekonzistencí při aktualizaci údajů. Společnost tak bude efektivněji svým klientům zprostředkovávat aktuální informace.

## Návrh řešení

### Hlavní představa

Systém SYBA SharedBase je internetová aplikace na správu firemních kontaktů osob a společností. Na rozdíl od stávajícího řešení bude mít zadavatel správu veškerých firemních kontaktů na jednom místě, což povede k deduplikaci kontaktů, zefektivnění jejich správy a snížení nákladů spojených s udržováním kontaktů aktuálních.

### Hlavní funkce

#### Funkce Popis

- FC-1 Vkládání, udržování a mazání kontaktů na osoby.
- FC-2 Vkládání, udržování a mazání skupin kontaktů.
- FC-3 Vkládání, udržování a mazání informací o firmách.
- FC-4 Vkládání, udržování a mazání kategorií firem.
- FC-5 Provádění synchronizace s účty zaměstnanců na službě Google.
- FC-6 Poskytování dat v systému, autentizace uživatelů prostřednictvím API.
- FC-7 Rozesílání hromadné e-mailové korespondence prostřednictvím služby třetích stran.

## Rozsah a omezení

### Rozsah prvních verzí

Funkce	Verze 1	Verze 2	Verze 3
FC-1	Plně implementována.	-	-
FC-2	-	Kontakty do skupiny lze vybírat i při úpravě skupiny.	-
FC-3	Plně implementována.	-	-

FC-4	Kategorie firem budou mít jednoduchou stromovou strukturu.	Bude možné vytvořit více stromových struktur kategorií, některé kategorie mohou být ve více strukturách.	-
FC-5	Manuální synchronizace kontaktů.	-	Synchronizace kontaktů bude probíhat automaticky v časových intervalech.
FC-6	CRUD operace prostřednictvím API.	Autentizace uživatelů prostřednictvím API.	-
FC-7	-	-	Plně implementována.

# Specifikace požadavků

## Úvod

### Předmět specifikace

Tato specifikace je určena pro představitele ČESKÉ OBALOVÉ ASOCIACE SYBA, s.r.o. (dále jen zadavatel) a pro vývojáře, kteří budou vyvíjet systém specifikovaný tímto dokumentem. Měl by sloužit zadavateli a dodavateli software jako podklad o podobě požadovaného systému.

Specifikace obsahuje přehled a popis jednotlivých požadavků na podobu realizovaného systému, a to tak podrobně, aby bylo zaručeno porozumění problému a návrhu jeho řešení oběma stranami. Slouží jako podklad pro následný návrh systému.

### Typografická konvence

V této specifikaci mají jednotlivé funkční požadavky na systém přiřazeny výrazy priority, které se řídí metodou prioritizace MoSCoW (Ash, 2007). Priorita požadavků je určena vzhledem k verzi 1.0 projektu. Aby byly jednotlivé požadavky snadno dohledatelné a aby se na ně mohlo snadno odkázat v žádosti o změnu, historii změn atp., jsou požadavky opatřeny jedinečným označením – hierarchickou textovou značkou, systémem navrženým Tomem Gilbem (Gilb, 1988).

Výraz česky	Anglicky	Vysvětlení
Nutné	Must have	Požadavky, které systém musí splnit. Tyto požadavky musí systém v plánované verzi splnit, jinak bude jeho doručení považováno za neúspěšné.
Měl by mít	Should have	Požadavky, které by systém měl splnit, pokud je to možné. Tyto požadavky jsou stejně důležité jako nutné požadavky, avšak jejich splnění není nezbytné v plánované verzi, nebo mohou být splněny i nepřímo.
Mohl by mít	Could have	Požadavky, které by systém mohl splnit, pokud zbude čas. Tyto požadavky jsou málo důležité, vyjadřují přidanou hodnotu, kterou by systém mohl v plánované verzi mít, pokud zbude čas.
V dalších verzích systému	Won't have	Požadavky, které budou splněny v dalších verzích systému. Tyto požadavky jsou nejméně důležité, protože s jejich splněním není počítáno v plánované verzi systému, ale mělo by s nimi být počítáno ve verzích dalších.

Tabulka 1 – Výrazy priority MoSCoW

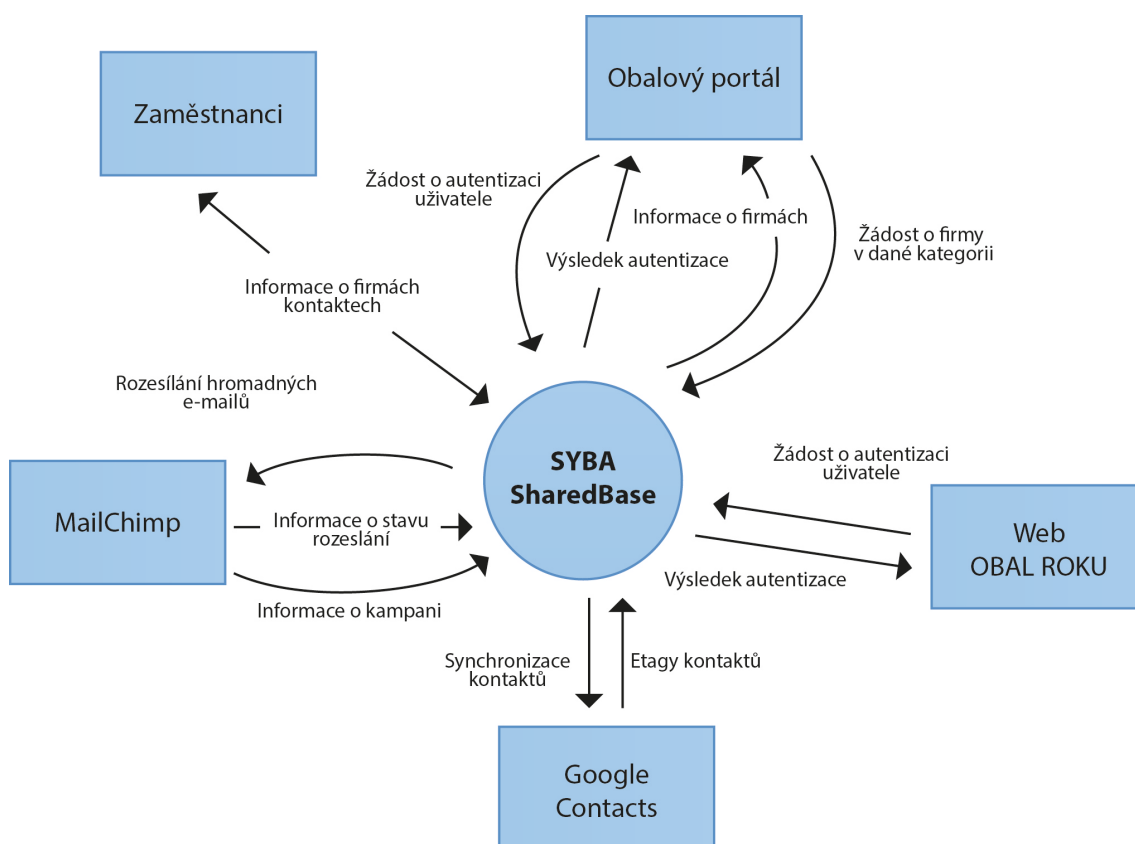
### Rozsah projektu a funkce systému

Informační systém SYBA SharedBase umožní zadavateli na jednom místě spravovat údaje o osobách a společnostech. Vize a rozsah projektu jsou popsány v samostatné kapitole (viz Vize a rozsah projektu).

## Obecný popis

### Kontext systému

Systém SYBA SharedBase je nový systém, který nahradí stávající nástroje na správu kontaktů a společností. Vývoj je naplánován na několik verzí, v první verzi je plánována správa kontaktů a společností, jednoduchá synchronizace se službou Google Kontakty a zpřístupnění dat v systému prostřednictvím webových služeb. V dalších verzích bude rozšířena synchronizace kontaktů, systém bude napojen na externí službu na zasílání hromadné korespondence a bude poskytovat rozhraní na autentizaci uživatelů prostřednictvím webové služby. Kontextový diagram na obrázku Obrázek 1 – Kontextový diagram systému SYBA SharedBase zobrazuje vnější entity a systémová rozhraní verze 3.0.



Obrázek 1 – Kontextový diagram systému SYBA SharedBase

### Třídy uživatelů a jejich vlastnosti

Třída	Popis
Klienti	Klienti ČESKÉ OBALOVÉ ASOCIACE SYBA, s. r. o., jsou fyzické nebo právnické osoby, využívající služeb této firmy. Jedná se o subjekty činné v sektoru balení, obalových prostředků, balicích materiálů a strojů. Data ze systému SYBA SharedBase jsou jim poskytována zejména prostřednictvím webové aplikace Obalového portálu či přímo zaměstnanci asociace.
Členové asociace	Členové ČESKÉ OBALOVÉ ASOCIACE SYBA, s. r. o., jsou klienti platící roční poplatky. Navrhovaný systém budou využívat pro získávání informací o osobách a firmách v tomto průmyslovém odvětví, systém bude navíc sloužit pro jejich autentizaci pro přístup k veřejnosti nepřístupným informacím na



	stránkách Obalového portálu.
Zaměstnanci	Zaměstnanci asociace mají díky systému aktuální informace o osobách a společnostech na jednom místě, které jsou díky synchronizaci se službou Google Kontakty dostupné i v jejich mobilních zařízeních. Jsou tak schopni snadno a rychle dodávat kvalitní informace klientům společnosti. Systém jim zároveň umožní komunikovat s klienty prostřednictvím hromadné elektronické korespondence.
Správci systému	Někteří ze zaměstnanců asociace mají přístup ke správě informací uložených v systému a jsou zodpovědní za udržování aktuálních údajů a ve verzi 1.0 za jejich synchronizaci se službou Google Kontakty.
Vývojáři	Vývojáři dalších systémů využívajících data a služby systému SYBA SharedBase mají přístup k detailní dokumentaci systémem poskytovaného rozhraní.

## Provozní prostředí

- PP-1 Systém bude odkudkoliv přístupný prostřednictvím sítě WWW.
- PP-2 Systém musí fungovat v prohlížečích Chrome 30.0 až 35.0, Internet Explorer 9.0 a 10.0.
- PP-3 Systém musí být nasazen na zadavatelem používaných serverech u společnosti Quantasoft, s.r.o. (Windows Server 2008, .NET Framework 4.0, 4.5, 4.5.1, MS SQL Server).
- PP-4 Systém bude komunikovat prostřednictvím webových služeb s Google Contacts API verze 2.

## Omezení návrhu a implementace

- ON-1 Systém bude jednojazyčný, používaný česky mluvícími uživateli.
- ON-2 Veškerý kód HTML musí odpovídat standardu HTML 5.
- ON-3 Serverová část aplikace musí být navržena podle návrhového vzoru MVC a naprogramována v jazyce C#.
- ON-4 Systém musí využívat pro perzistenci dat databázi MS SQL Server.
- ON-5 Data poskytovaná prostřednictvím webové služby musí být dostupná ve formátech JSON a XML.
- ON-6 Veškerý kód systému bude nahraný v GIT repozitářích zadavatele na službě GitHub.

## Uživatelská dokumentace

- UD-1 Systém bude poskytovat podrobnou online dokumentaci pro vývojáře popisující poskytované rozhraní webové služby, URL adresy přístupových bodů, popisy a příklady přenášených dat alespoň ve formátu JSON.

## Předpoklady a závislosti

- PZ-1 Provoz modulu na rozesílání hromadné korespondence závisí na externí e-mailové službě, která musí být dostupná s aktivní licencí.

## Funkce systému

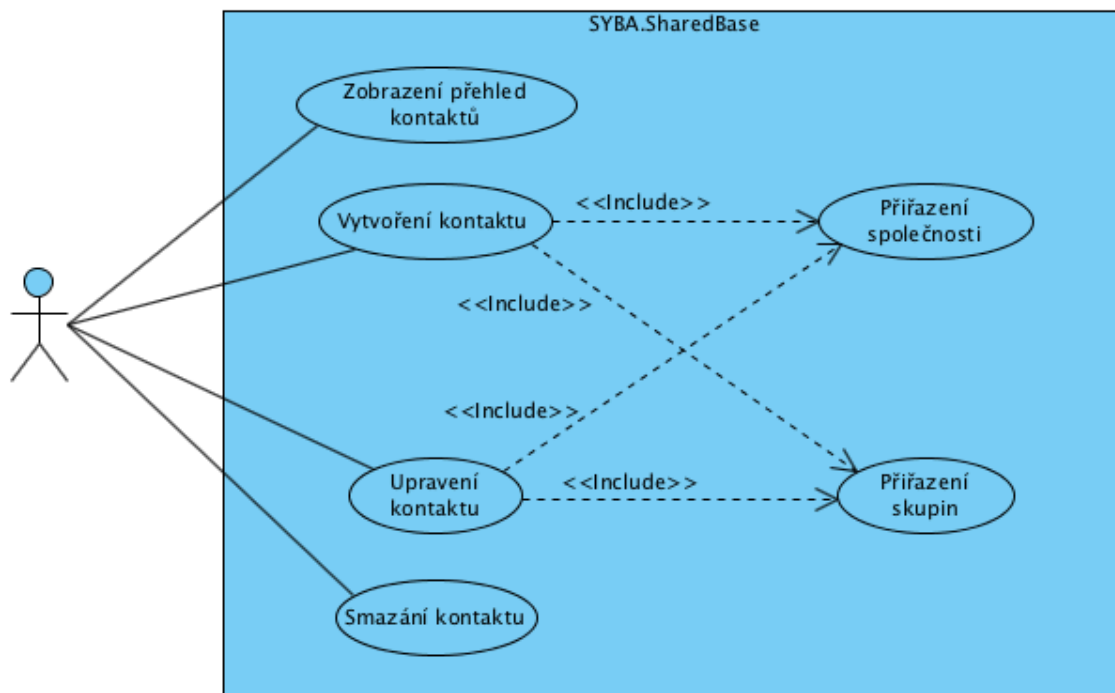
V této sekci jsou rozepsány jednotlivé požadavky na funkce systému SYBA SharedBase. U každé funkce je uveden její krátký popis.

## 1. Správa kontaktů

### 1.1. Popis

Systém bude vést databázi kontaktů. Autorizovaný uživatel může vytvořit nový záznam o kontaktu, aktualizovat informace ve stávajícím záznamu nebo záznam smazat.

### 1.2. Případy užití



Obrázek 2 – PU: Správa kontaktů

#### 1.2.1. Vytvoření kontaktu

**Případ užití číslo:** PU-1.1

**Název:** Vytvoření kontaktu

**Aktéři:** Editor

**Popis:** Uživatel se prostřednictvím Internetu přihlásí k systému. Zvolí akci vedoucí k vytvoření nového záznamu, vyplní informace o kontaktu a systém tento kontakt uloží do databáze.

**Vstupní podmínky:** 1. Uživatel je přihlášený do systému.

**Výstupní podmínky:**

1. V systému je veden nový záznam kontaktu.
2. Aktualizuje se seznam změn připravených k aktualizaci s propojenými systémy.
3. V případě zrušení zadávání nového kontaktu (viz alternativní cesta číslo 1) nebude v databázi vytvořen žádný záznam.

**Běžná cesta:**

1. Uživatel zvolí akci k vytvoření nového kontaktu.
2. Systém zobrazí formulář záznamu o kontaktu.
3. Uživatel vyplní formulář.
  - 3.1. Uživatel vyplní informace o kontaktu.

- 3.2. Systém provede případ užití PU-1.2 Přiřazení společnosti.
  - 3.3. Systém provede případ užití PU-1.3 Přiřazení skupin kontaktů.
  - 4. Uživatel systému sdělí, že kontakt je hotový.
  - 5. Systém validuje zadané informace o kontaktu.
  - 6. Validní záznam o kontaktu systém uloží do databáze.
  - 7. Systém zobrazí editační formulář nově vytvořeného kontaktu.
- Alternativní cesty:**
- 1. Uživatel se rozhodne zrušit vytváření nového kontaktu (kdykoli po kroku 1 a před krokem 6).
    - 1.1. Uživatel zvolí akci ke zrušení nového kontaktu.
    - 1.2. Systém zruší zadávání a zobrazí stránku přehledu kontaktů.
- Výjimky:**
- 1. Zadané informace o kontaktu nejsou validní (po kroku 5).
    - 1.1. Systém zobrazí chybová hlášení u špatně vyplněných údajů.
    - 1.2. Uživatel opraví tyto údaje.
    - 1.3. Systém se vrátí zpět na krok 5.
- Vložené případy užití:** PU-1.2 Přiřazení společnosti  
PU-1.3 Přiřazení skupin kontaktů
- Zvláštní požadavky:** Nejsou.

### **1.2.2. Přiřazení společnosti**

- Případ užití číslo:** PU-1.2
- Název:** Přiřazení společnosti
- Aktéři:** Editor
- Popis:** Uživatel je ve stavu zadávání informací o kontaktu. Z nabídky si může zvolit společnost zaměstnavatele osoby, které se zadávaný kontakt týká.
- Vstupní podmínky:**
- 1. Uživatel je přihlášený do systému.
  - 2. Systém je ve stavu zadávání nebo úpravy záznamu kontaktu.
- Výstupní podmínky:**
- 1. Ke kontaktu je přiřazena žádná, nebo jedna společnost.
- Běžná cesta:**
- 1. Systém zobrazí seznam společností vedených v systému.
  - 2. Uživatel vybere společnost.
  - 3. Systém uloží vztah kontaktu a vybrané společnosti do databáze.
- Alternativní cesty:** Nejsou.
- Výjimky:**
- 1. V systému nejsou vedeny žádné záznamy společností.
    - 1.1. Konec případu užití.
- Vložené případy užití:** Nejsou.
- Zvláštní požadavky:** Nejsou.

### **1.2.3. Přiřazení skupin**

- Případ užití číslo:** PU-1.3
- Název:** Přiřazení skupin
- Aktéři:** Editor

<b>Popis:</b>	Uživatel je ve stavu zadávání informací o kontaktu. Z nabídky si může zvolit skupiny kontaktů, do kterých má být tento kontakt přiřazen.
<b>Vstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Uživatel je přihlášený do systému.</li> <li>2. Systém je ve stavu zadávání nebo úpravy záznamu kontaktu.</li> </ol>
<b>Výstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Ke kontaktu je přiřazena žádná, nebo jedna, nebo více skupin kontaktů.</li> </ol>
<b>Běžná cesta:</b>	<ol style="list-style-type: none"> <li>1. Systém zobrazí seznam skupin kontaktů vedených v systému.</li> <li>2. Uživatel vybere skupiny.</li> <li>3. Systém uloží vztahy kontaktu a vybraných skupin kontaktů do databáze.</li> </ol>
<b>Alternativní cesty:</b>	Nejsou.
<b>Výjimky:</b>	<ol style="list-style-type: none"> <li>1. V systému nejsou vedeny žádné skupiny kontaktů. <ol style="list-style-type: none"> <li>1.1. Konec případu užití.</li> </ol> </li> </ol>
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

#### **1.2.4. Upravení kontaktu**

<b>Případ užití číslo:</b>	PU-1.4
<b>Název:</b>	Upravení kontaktu
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel může změnit informace uložené v záznamu kontaktu.
<b>Vstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Uživatel je přihlášený do systému.</li> <li>2. Kontakt je vedený v systému.</li> </ol>
<b>Výstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. V databázi je uložen změněný záznam kontaktu.</li> <li>2. Aktualizuje se seznam změn připravených k aktualizaci s propojenými systémy.</li> <li>3. V případě zrušení úpravy kontaktu (viz alternativní cesta číslo 1) nebude v databázi vytvořen ani změněn žádný záznam.</li> </ol>
<b>Běžná cesta:</b>	<ol style="list-style-type: none"> <li>1. Uživatel zvolí akci k úpravě stávajícího kontaktu.</li> <li>2. Systém zobrazí formulář záznamu o kontaktu.</li> <li>3. Uživatel provede změny ve formuláři. <ol style="list-style-type: none"> <li>3.1. Uživatel upraví informace o kontaktu.</li> <li>3.2. Systém provede případ užití PU-1.2 Přiřazení společnosti.</li> <li>3.3. Systém provede případ užití PU-1.3 Přiřazení skupin kontaktů.</li> </ol> </li> <li>4. Uživatel systému sdělí, že kontakt je hotový.</li> <li>5. Systém validuje zadané informace o kontaktu.</li> <li>6. Validní záznam o kontaktu systém uloží do databáze.</li> <li>7. Systém zobrazí editační formulář kontaktu.</li> </ol>
<b>Alternativní cesty:</b>	<ol style="list-style-type: none"> <li>1. Uživatel se rozhodne zrušit úpravu kontaktu (kdykoli po kroku 1 a před krokem 6). <ol style="list-style-type: none"> <li>1.1. Uživatel zvolí akci ke zrušení úpravy kontaktu.</li> </ol> </li> </ol>

1.2. Systém zruší editaci a zobrazí stránku přehledu kontaktů.

<b>Výjimky:</b>	Nejsou.
<b>Vložené případy užití:</b>	PU-1.2 Přiřazení společnosti PU-1.3 Přiřazení skupin kontaktů
<b>Zvláštní požadavky:</b>	Nejsou.

#### **1.2.5. Smazání kontaktu**

<b>Případ užití číslo:</b>	PU-1.5
<b>Název:</b>	Smazání kontaktu
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel může smazat záznam kontaktu.
<b>Vstupní podmínky:</b>	<ol style="list-style-type: none"><li>1. Uživatel je přihlášený do systému.</li><li>2. Kontakt je vedený v systému.</li></ol>
<b>Výstupní podmínky:</b>	<ol style="list-style-type: none"><li>1. Z databáze je kontakt označen ke smazání.</li><li>2. Aktualizuje se seznam změn připravených k aktualizaci s propojenými systémy.</li><li>3. V případě zrušení požadavku na smazání kontaktu nebude v databázi smazán ani změněn žádný záznam.</li></ol>
<b>Běžná cesta:</b>	<ol style="list-style-type: none"><li>1. Uživatel zvolí akci ke smazání stávajícího kontaktu.</li><li>2. Systém se zeptá uživatele, zda chce záznam opravdu smazat.</li><li>3. Uživatel požadavek potvrdí.</li><li>4. Systém označí záznam ke smazání.</li></ol>
<b>Alternativní cesty:</b>	<ol style="list-style-type: none"><li>1. Uživatel se rozhodne zrušit smazání kontaktu (po kroku 2).<ol style="list-style-type: none"><li>1.1. Uživatel zvolí akci ke zrušení smazání kontaktu.</li><li>1.2. Systém zruší požadavek na smazání kontaktu.</li></ol></li></ol>
<b>Výjimky:</b>	Nejsou.
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

#### **1.2.6. Zobrazení přehledu kontaktů**

<b>Případ užití číslo:</b>	PU-1.6
<b>Název:</b>	Zobrazení přehledu kontaktů
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel může zobrazit přehled kontaktů, výsledky na tomto přehledu může filtrovat a řadit podle zvolených atributů.
<b>Vstupní podmínky:</b>	<ol style="list-style-type: none"><li>1. Uživatel je přihlášený do systému.</li></ol>
<b>Výstupní podmínky:</b>	Nejsou.
<b>Běžná cesta:</b>	<ol style="list-style-type: none"><li>1. Uživatel zvolí akci k zobrazení přehledu kontaktů.</li><li>2. Systém zobrazí seznam kontaktů, textové pole pro zadání řetězce pro filtrování výsledků a ovládání řazení u atributů Jméno a Společnost.</li><li>3. Pokud je v systému vedeno více než 50 kontaktů, systém na stránce</li></ol>

	zobrazí navigaci mezi stránkami kontaktů.
<b>Alternativní cesty:</b>	<ol style="list-style-type: none"> <li>1. Uživatel chce filtrovat výsledky v přehledu kontaktů (po kroku 2 nebo 3). <ol style="list-style-type: none"> <li>1.1. Uživatel vyplní řetězec pro filtrování výsledků.</li> <li>1.2. Systém zobrazí výsledky odpovídající zadanému filtru.</li> </ol> </li> <li>2. Uživatel chce seřadit výsledky v přehledu kontaktů (po kroku 2 nebo 3). <ol style="list-style-type: none"> <li>2.1. Uživatel zvolí atribut a typ řazení (vzestupně, sestupně).</li> <li>2.2. Systém zobrazí výsledky seřazené podle zvoleného atributu a typu řazení.</li> </ol> </li> <li>3. Uživatel chce přejít na jinou stránku s kontakty (po kroku 3 běžné cesty a po kroku 1.2 nebo 2.2 alternativních cest). <ol style="list-style-type: none"> <li>3.1. Uživatel zvolí akci přechodu na předchozí nebo následující stránku, nebo stránku podle čísla jejího pořadí.</li> <li>3.2. Systém zobrazí kontakty odpovídající stránce, na které se uživatel nachází.</li> </ol> </li> </ol>
<b>Výjimky:</b>	Nejsou.
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

### 1.3. Funkční požadavky

Id požadavku	Priorita	Popis
Kontakt. Model	Nutné	Systém bude vést databázi kontaktů osob a s nimi spojených údajů (viz <a href="#">datový model kontaktu</a> ).
Kontakt.Vytvořit	Nutné	Systém umožní vytvořit nový kontakt.
Kontakt.Vytvořit.Společnost	Nutné	Systém umožní přiřadit ke kontaktu maximálně jeden záznam společnosti.
Kontakt.Vytvořit.Skupiny	Nutné	Systém umožní přiřadit kontakt do žádné, jedné nebo více skupin kontaktů.
Kontakt.Vytvořit.Validace	Nutné	Systém bude validovat vkládané e-mailové adresy a telefonní čísla.
Kontakt.Náhled	Nutné	Systém umožní zobrazit náhled kontaktu.
Kontakt.Náhled.Schránka	Mohl by mít	Systém umožní zkopírovat obsah náhledu kontaktu do schránky operační paměti.
Kontakt.Upravit	Nutné	Systém umožní upravit stávající kontakt.
Kontakt.Upravit.Zrušit	Nutné	Pokud uživatel upravuje kontakt, systém mu umožní ukončit editaci bez zachování provedených změn.
Kontakt.Smazat	Nutné	Systém umožní smazat stávající kontakt.
Kontakt.Smazat.Potvrzení	Nutné	Pokud uživatel požaduje smazání kontaktu, systém požaduje potvrzení požadavku.
Kontakt.Smazat.Zrušit	Nutné	Uživatel může zrušit požadavek na smazání kontaktu.

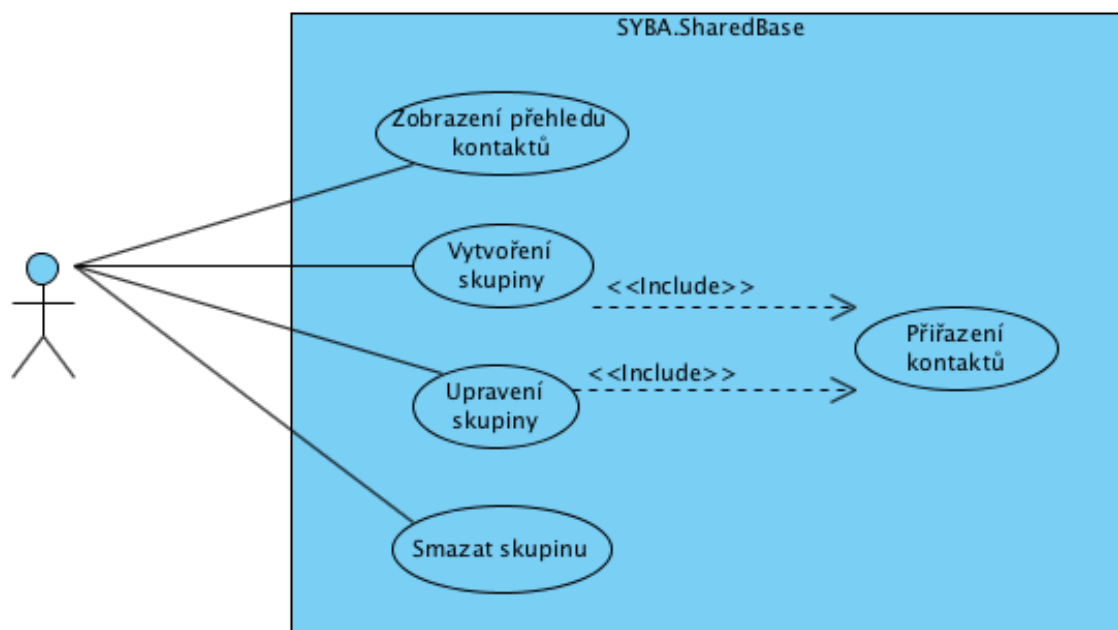
Kontakt.Smazat.Společnost	Nutné	Smazání kontaktu ponechá v systému záznam případné přidružené společnosti.
Kontakt.Smazat.Skupina	Nutné	Smazání kontaktu ponechá v systému záznamy případných přidružených skupin kontaktů.
Kontakt.Přehled	Nutné	Systém umožní zobrazit přehled všech kontaktů. U každého kontaktu v přehledu bude zobrazovat jméno, e-mailové adresy, název společnosti, názvy skupin kontaktů a akce na zobrazení detailu, úpravu a smazání kontaktu.
Kontakt.Přehled.Filtr	Měl by mít	Systém bude filtrovat přehled kontaktů na základě uživatelem zadaného textového filtru.
Kontakt.Přehled.Filtr.Skupina	Mohl by mít	Systém zobrazí v přehledu pouze kontakty v uživatelem zvolené skupině.
Kontakt.Přehled.Filtr.Firma	Mohl by mít	Systém zobrazí v přehledu pouze kontakty v uživatelem zvolené společnosti.
Kontakt.Přehled.Stránkování	Nutné	Systém bude stránkovat přehled kontaktů v počtu 50 kontaktů na stránku.
Kontakt.Přehled.Řazení	Měl by mít	Systém bude řadit kontakty v přehledu podle uživatelem zvoleného atributu (jméno, společnost) a typu řazení (vzestupně, sestupně).

## 2. Správa skupin kontaktů

### 2.1. Popis

Systém bude vést databázi skupin kontaktů. Autorizovaný uživatel může vytvořit nový záznam skupiny kontaktů, aktualizovat informace ve stávajícím záznamu, přidávat nebo ubírat kontakty v této skupině nebo záznam smazat. V případě vytvoření nového záznamu, úpravy nebo smazání je třeba označit všechny přidružené kontakty jako změněné, aby se změny projevíly při následné synchronizaci s vnějšími systémy.

## 2.2. Případy užití



Obrázek 3 – PU: Správa skupin kontaktů

### 2.2.1. Vytvoření skupiny kontaktů

**Případ užití číslo:** PU-2.1

**Název:** Vytvoření skupiny kontaktů

**Aktéři:** Editor

**Popis:** Uživatel se prostřednictvím Internetu přihlásí k systému. Zvolí akci vedoucí k vytvoření nového záznamu, vyplní informace o skupině kontaktů a systém tento záznam uloží do databáze.

**Vstupní podmínky:** 1. Uživatel je přihlášený do systému.

**Výstupní podmínky:**

1. V systému je veden nový záznam skupiny kontaktů.
2. Aktualizuje se seznam změn připravených k aktualizaci s propojenými systémy.
3. V případě zrušení zadávání nového záznamu (viz alternativní cesta číslo 1) nebude v databázi vytvořen žádný záznam.

**Běžná cesta:**

1. Uživatel zvolí akci k vytvoření nové skupiny kontaktů.
2. Systém zobrazí formulář záznamu o skupině kontaktů.
3. Uživatel vyplní formulář.
  - 3.1. Uživatel vyplní informace o skupině kontaktů.
  - 3.2. Systém provede případ užití PU-2.2 Přřazení kontaktů.
4. Uživatel systému sdělí, že záznam je hotový.
5. Systém validuje zadané informace o skupině kontaktů.
6. Validní záznam systém uloží do databáze.
7. Systém zobrazí editační formulář nově vytvořené skupiny kontaktů.

**Alternativní cesty:**

1. Uživatel se rozhodne zrušit vytváření nové skupiny kontaktů (kdykoli po kroku 1 a před krokem 6).



- 1.1. Uživatel zvolí akci ke zrušení nového záznamu.
- 1.2. Systém zruší zadávání a zobrazí stránku přehledu skupin kontaktů.

- Výjimky:**
1. Zadané informace o skupině kontaktů nejsou validní (po kroku 5).
    - 1.1. Systém zobrazí chybová hlášení u špatně vyplněných údajů.
    - 1.2. Uživatel opraví tyto údaje.
    - 1.3. Systém se vrátí zpět na krok 5.

**Vložené případy užití:** PU-2.2 Přiřazení kontaktů

**Zvláštní požadavky:** Nejsou.

### **2.2.2. Přiřazení kontaktů**

**Případ užití číslo:** PU-2.2

**Název:** Přiřazení kontaktů

**Aktéři:** Editor

**Popis:** Uživatel je ve stavu zadávání informací o skupině kontaktů. Z nabídky může zvolit kontakty, které mají být přiřazeny do této skupiny.

- Vstupní podmínky:**
1. Uživatel je přihlášený do systému.
  2. Systém je ve stavu zadávání nebo úpravy záznamu skupiny kontaktů.

**Výstupní podmínky:** 1. Ke skupině kontaktů je přiřazen žádný, jeden nebo více kontaktů.

- Běžná cesta:**
1. Systém zobrazí seznam kontaktů vedených v systému.
  2. Uživatel vybere kontakt.
  3. Systém uloží vztah skupiny a vybraného kontaktu do databáze.

**Alternativní cesty:** Nejsou.

- Výjimky:**
1. V systému nejsou vedeny žádné záznamy kontaktů.
    - 1.1. Konec případu užití.

**Vložené případy užití:** Nejsou.

**Zvláštní požadavky:** Nejsou.

### **2.2.3. Upravení skupiny kontaktů**

**Případ užití číslo:** PU-2.3

**Název:** Upravení skupiny kontaktů

**Aktéři:** Editor

**Popis:** Uživatel může změnit informace uložené v záznamu skupiny kontaktů.

- Vstupní podmínky:**
1. Uživatel je přihlášený do systému.
  2. Skupina kontaktů je vedena v systému.

**Výstupní podmínky:**

1. V databázi je uložen změněný záznam skupiny kontaktů.
2. Aktualizuje se seznam změn připravených k aktualizaci s propojenými systémy.

3. V případě zrušení úpravy skupiny kontaktů (viz alternativní cesta číslo 1) nebude v databázi vytvořen ani změněný žádný záznam.

- Běžná cesta:**
1. Uživatel zvolí akci k úpravě stávající skupiny kontaktů.

2. Systém zobrazí formulář záznamu skupiny kontaktů.
3. Uživatel provede změny ve formuláři.
  - 3.1. Uživatel upraví informace o skupině kontaktů.
  - 3.2. Systém provede případ užití PU-2.2 Přiřazení kontaktů.
4. Uživatel systému sdělí, že záznam je hotový.
5. Systém validuje zadané informace o skupině kontaktů.
6. Validní záznam systém uloží do databáze.
7. Systém zobrazí editační formulář skupiny kontaktů.

- Alternativní cesty:**
1. Uživatel se rozhodne zrušit úpravu záznamu (kdykoli po kroku 1 a před krokem 6).
    - 1.1. Uživatel zvolí akci ke zrušení úpravy záznamu.
    - 1.2. Systém zruší editaci a zobrazí stránku přehledu skupin kontaktů.

**Výjimky:** Nejsou.

**Vložené případy užití:** PU-2.2 Přiřazení kontaktů

**Zvláštní požadavky:** Nejsou.

#### **2.2.4. Smazání skupiny kontaktů**

**Případ užití číslo:** PU-2.4

**Název:** Smazání skupiny kontaktů

**Aktéři:** Editor

**Popis:** Uživatel může smazat záznam skupiny kontaktů.

- Vstupní podmínky:**
1. Uživatel je přihlášený do systému.
  2. Skupina kontaktů je vedena v systému.

- Výstupní podmínky:**
1. Skupina kontaktů je označena ke smazání.
  2. Skupina je odebrána ze záznamů přidružených kontaktů.
  3. Aktualizuje se seznam změn připravených k aktualizaci s propojenými systémy.
  4. V případě zrušení požadavku na smazání záznamu nebude v databázi smazán ani změněn žádný záznam.

- Běžná cesta:**
1. Uživatel zvolí akci ke smazání v systému vedeného záznamu skupiny kontaktů.
  2. Systém se zeptá uživatele, zda chce záznam opravdu smazat.
  3. Uživatel požadavek potvrdí.
  4. Systém označí záznam ke smazání.

- Alternativní cesty:**
1. Uživatel se rozhodne zrušit smazání záznamu (po kroku 2).
    - 1.1. Uživatel zvolí akci ke zrušení smazání záznamu.
    - 1.2. Systém zruší požadavek na smazání záznamu.

**Výjimky:** Nejsou.

**Vložené případy užití:** Nejsou.

**Zvláštní požadavky:** Nejsou.

### 2.2.5. Zobrazení přehledu skupin kontaktů

<b>Případ užití číslo:</b>	PU-2.5
<b>Název:</b>	Zobrazení přehledu skupin kontaktů
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel může zobrazit přehled skupin kontaktů, výsledky na tomto přehledu může filtrovat a řadit podle zvolených atributů.
<b>Vstupní podmínky:</b>	1. Uživatel je přihlášený do systému.
<b>Výstupní podmínky:</b>	Nejsou.
<b>Běžná cesta:</b>	<ol style="list-style-type: none"><li>1. Uživatel zvolí akci k zobrazení přehledu skupin kontaktů.</li><li>2. Systém zobrazí seznam skupin kontaktů, textové pole pro zadání řetězce pro filtrování výsledků a ovládání řazení u atributů Název a Název anglicky.</li><li>3. Pokud je v systému vedeno více než 50 skupin kontaktů, systém na stránce zobrazí navigaci mezi stránkami skupin kontaktů.</li></ol>
<b>Alternativní cesty:</b>	<ol style="list-style-type: none"><li>1. Uživatel chce filtrovat výsledky v přehledu (po kroku 2 nebo 3).<ol style="list-style-type: none"><li>1.1. Uživatel vyplní řetězec pro filtrování výsledků.</li><li>1.2. Systém zobrazí výsledky odpovídající zadanému filtru.</li></ol></li><li>2. Uživatel chce seřadit výsledky v přehledu (po kroku 2 nebo 3).<ol style="list-style-type: none"><li>2.1. Uživatel zvolí atribut a typ řazení (vzestupně, sestupně).</li><li>2.2. Systém zobrazí výsledky seřazené podle zvoleného atributu a typu řazení.</li></ol></li><li>3. Uživatel chce přejít na jinou stránku (po kroku 3 běžné cesty a 1.2 nebo 2.2 alternativních cest).<ol style="list-style-type: none"><li>3.1. Uživatel zvolí akci přechodu na předchozí nebo následující stránku, nebo stránku podle čísla jejího pořadí.</li><li>3.2. Systém zobrazí záznamy odpovídající stránce, na které se uživatel nachází.</li></ol></li></ol>
<b>Výjimky:</b>	Nejsou.
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

### 2.3. Funkční požadavky

Id požadavku	Priorita	Popis
Skupina.Model	Nutné	Systém bude vést databázi skupin kontaktů a s nimi spojených údajů (viz <a href="#">datový model skupiny kontaktů</a> ).
Skupina.Vytvořit	Nutné	Systém umožní vytvořit novou skupinu kontaktů.
Skupina.Náhled	Nutné	Systém umožní zobrazit náhled skupiny kontaktů.
Skupina.Upravit	Nutné	Systém umožní upravit skupinu kontaktů.
Skupina.Upravit.Kontakty	Nutné	Systém umožní přidávat a odebírat kontakty přidružené ke skupině kontaktů.
Skupina.Upravit.Zrušit	Nutné	Pokud uživatel upravuje skupinu kontaktů, systém mu umožní ukončit editaci bez zachování

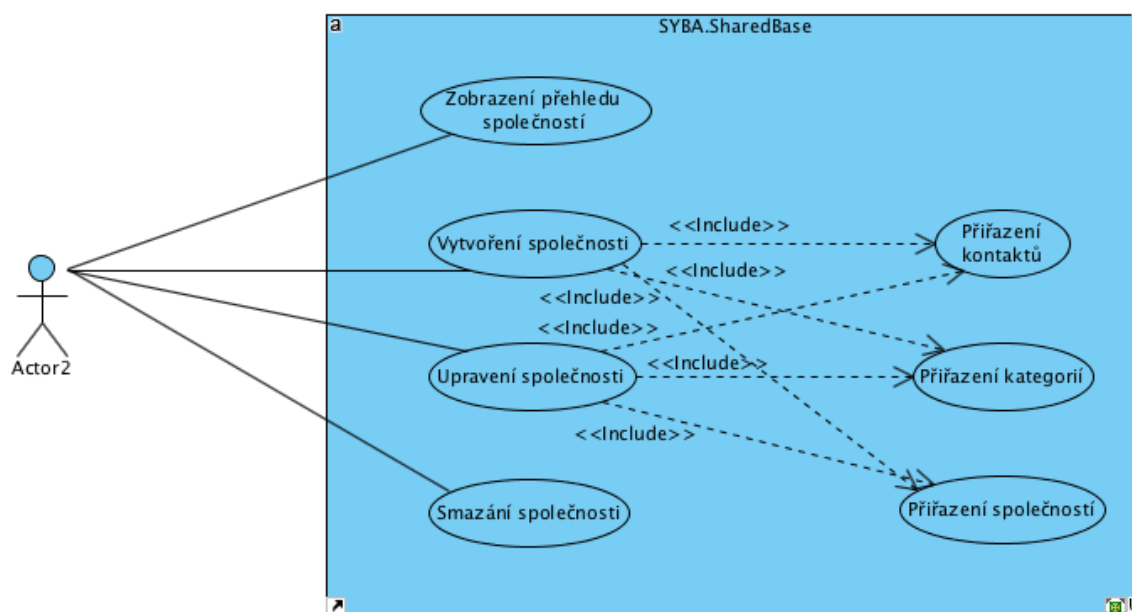
Skupina.Smazat	Nutné	provedených změn.
Skupina.Smazat.Potvrzení	Nutné	System umožní smazat skupinu kontaktů. Pokud uživatel požaduje smazání skupiny kontaktů, systém požaduje potvrzení požadavku.
Skupina.Smazat.Zrušit	Nutné	Uživatel může zrušit požadavek na smazání skupiny kontaktů.
Skupina.Přehled	Nutné	System umožní zobrazit přehled všech skupin kontaktů.
Skupina.Přehled.Filtr	Mohl by mít	System bude filtrovat přehled skupin kontaktů na základě uživatelem zadaného textového filtru.
Skupina.Přehled.Stránkování	Nutné	System bude stránkovat přehled skupin kontaktů v počtu 50 skupin na stránku.
Skupina.Přehled.Řazení	Mohl by mít	System bude řadit skupiny kontaktů v přehledu podle uživatelem zvoleného atributu (název, počet kontaktů) a typu řazení (vzestupně, sestupně).

### 3. Správa organizací

#### 3.1. Popis

System bude vést databázi společnosti. Autorizovaný uživatel může vytvořit nový záznam společnosti, aktualizovat informace ve stávajícím záznamu, spravovat přiřazené kontakty a kategorie společností nebo záznam smazat. V případě vytvoření nového záznamu, úpravy nebo smazání je třeba označit všechny přidružené kontakty jako změněné, aby se změny projevíly při následné synchronizaci s vnějšími systémy.

#### 3.2. Případy užití



Obrázek 4 – PU: Správa společností

### **3.2.1. Vytvoření společnosti**

<b>Případ užití číslo:</b>	PU-3.1
<b>Název:</b>	Vytvoření společnosti
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel se prostřednictvím Internetu přihlásí k systému. Zvolí akci vedoucí k vytvoření nového záznamu, vyplní informace o společnosti a systém tento záznam uloží do databáze.
<b>Vstupní podmínky:</b>	1. Uživatel je přihlášený do systému.
<b>Výstupní podmínky:</b>	1. V systému je veden nový záznam společnosti. 2. V případě zrušení zadávání nového záznamu (viz alternativní cesta číslo 1) nebude v databázi vytvořen žádný záznam.
<b>Běžná cesta:</b>	1. Uživatel zvolí akci k vytvoření nového záznamu společnosti. 2. Systém zobrazí formulář záznamu společnosti. 3. Uživatel vyplní formulář. 3.1. Uživatel vyplní informace o společnosti. 3.2. Systém provede případ užití PU-3.2 Přiřazení kontaktů. 3.3. Systém provede případ užití PU-3.3 Přiřazení kategorií. 3.4. Systém provede případ užití PU-3.4 Přiřazení společností. 4. Uživatel systému sdělí, že záznam je hotový. 5. Systém validuje zadané informace o společnosti. 6. Validní záznam systém uloží do databáze. 7. Systém zobrazí editační formulář nově vytvořené společnosti.
<b>Alternativní cesty:</b>	1. Uživatel se rozhodne zrušit vytváření nového záznamu (kdykoli po kroku 1 a před krokem 6). 1.1. Uživatel zvolí akci ke zrušení nového záznamu. 1.2. Systém zruší zadávání a zobrazí stránku přehledu společností.
<b>Výjimky:</b>	1. Zadané informace o společnosti nejsou validní (po kroku 5). 1.1. Systém zobrazí chybová hlášení u špatně vyplněných údajů. 1.2. Uživatel opraví tyto údaje. 1.3. Systém se vrátí zpět na krok 5.
<b>Vložené případy užití:</b>	PU-3.2 Přiřazení kontaktů PU-3.3 Přiřazení kategorií PU-3.4 Přiřazení společností
<b>Zvláštní požadavky:</b>	Nejsou.

### **3.2.2. Přiřazení kontaktů**

<b>Případ užití číslo:</b>	PU-3.2
<b>Název:</b>	Přiřazení kontaktů
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel je ve stavu zadávání informací o společnosti. Z nabídky může zvolit kontakty zaměstnanců, které mají být přiřazeny k této společnosti.

<b>Vstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Uživatel je přihlášený do systému.</li> <li>2. Systém je ve stavu zadávání nebo úpravy záznamu společnosti.</li> </ol>
<b>Výstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Ke společnosti je přiřazen žádný, jeden nebo více kontaktů.</li> </ol>
<b>Běžná cesta:</b>	<ol style="list-style-type: none"> <li>1. Systém zobrazí seznam kontaktů vedených v systému.</li> <li>2. Uživatel vybere kontakty.</li> <li>3. Systém uloží vztah společnosti a vybraných kontaktů do databáze.</li> </ol>
<b>Alternativní cesty:</b>	Nejsou.
<b>Výjimky:</b>	<ol style="list-style-type: none"> <li>1. V systému nejsou vedeny žádné záznamy kontaktů. <ol style="list-style-type: none"> <li>1.1. Konec případu užití.</li> </ol> </li> </ol>
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

### **3.2.3. Přiřazení kategorií**

<b>Případ užití číslo:</b>	PU-3.3
<b>Název:</b>	Přiřazení kategorií
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel je ve stavu zadávání informací o společnosti. Z nabídky může zvolit kategorie, které mají být přiřazeny k této společnosti.
<b>Vstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Uživatel je přihlášený do systému.</li> <li>2. Systém je ve stavu zadávání nebo úpravy záznamu společnosti.</li> </ol>
<b>Výstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Ke společnosti je přiřazena žádná, jedna nebo více kategorií.</li> </ol>
<b>Běžná cesta:</b>	<ol style="list-style-type: none"> <li>4. Systém zobrazí seznam kategorií vedených v systému.</li> <li>5. Uživatel vybere kategorie.</li> <li>6. Systém uloží vztah společnosti a vybraných kategorií do databáze.</li> </ol>
<b>Alternativní cesty:</b>	Nejsou.
<b>Výjimky:</b>	<ol style="list-style-type: none"> <li>1. V systému nejsou vedeny žádné záznamy kategorií. <ol style="list-style-type: none"> <li>1.1. Konec případu užití.</li> </ol> </li> </ol>
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

### **3.2.4. Přiřazení společností**

<b>Případ užití číslo:</b>	PU-3.4
<b>Název:</b>	Přiřazení společností
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel je ve stavu zadávání informací o společnosti. Z nabídky může zvolit společnosti, které mají být přiřazeny k této společnosti jako organizace, kterých je tato společnost členem.
<b>Vstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Uživatel je přihlášený do systému.</li> <li>2. Systém je ve stavu zadávání nebo úpravy záznamu společnosti.</li> </ol>
<b>Výstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Ke společnosti je přiřazena žádná, jedna nebo více společností.</li> </ol>
<b>Běžná cesta:</b>	<ol style="list-style-type: none"> <li>1. Systém zobrazí seznam společností vedených v systému.</li> </ol>

2. Uživatel vybere společnosti.
3. Systém uloží vztah upravované společnosti a vybraných společností do databáze.

**Alternativní cesty:** Nejsou.

- Výjimky:**
1. V systému nejsou vedeny žádné záznamy společností.
    - 1.1. Konec případu užití.
  2. Systém nenabízí v seznamu společností k přiřazení právě upravovanou společnost.

**Vložené případy užití:** Nejsou.

**Zvláštní požadavky:** Nejsou.

### **3.2.5. Upravení společnosti**

**Případ užití číslo:** PU-3.5

**Název:** Upravení společnosti

**Aktéři:** Editor

**Popis:** Uživatel může změnit informace uložené v záznamu společnosti.

- Vstupní podmínky:**
1. Uživatel je přihlášený do systému.
  2. Společnost je vedena v systému.

- Výstupní podmínky:**
1. V databázi je uložen změněný záznam společnosti.
  2. Aktualizuje se seznam změn připravených k aktualizaci s propojenými systémy.
  3. V případě zrušení úpravy záznamu (viz alternativní cesta číslo 1) nebude v databázi vytvořen ani změněn žádný záznam.

- Běžná cesta:**
1. Uživatel zvolí akci k úpravě v systému vedeného záznamu společnosti.
  2. Systém zobrazí formulář záznamu.
  3. Uživatel provede změny ve formuláři.
    - 3.1. Uživatel upraví informace o společnosti.
    - 3.2. Systém provede případ užití PU-3.2 Přiřazení kontaktů.
    - 3.3. Systém provede případ užití PU-3.3 Přiřazení kategorií.
    - 3.4. Systém provede případ užití PU-3.4 Přiřazení společností.
  4. Uživatel systému sdělí, že záznam je hotový.
  5. Systém validuje zadané informace o společnosti.
  6. Validní záznam systém uloží do databáze.
  7. Systém zobrazí editační formulář společnosti.

- Alternativní cesty:**
1. Uživatel se rozhodne zrušit úpravu záznamu (kdykoli po kroku 1 a před krokem 6).
    - 1.1. Uživatel zvolí akci ke zrušení úpravy záznamu.
    - 1.2. Systém zruší editaci a zobrazí stránku přehledu společností.

**Výjimky:** Nejsou.

**Vložené případy užití:** PU-3.2 Přiřazení kontaktů

PU-3.3 Přiřazení kategorií  
 PU-3.4 Přiřazení společností

**Zvláštní požadavky:** Nejsou.

### **3.2.6. Smazání společnosti**

**Případ užití číslo:** PU-3.6

**Název:** Smazání společnosti

**Aktéři:** Editor

**Popis:** Uživatel může smazat záznam společnosti.

**Vstupní podmínky:**

1. Uživatel je přihlášený do systému.
2. Společnost je vedena v systému.

**Výstupní podmínky:**

1. Záznam společnosti je vymazán.
2. Společnost je odebrána ze záznamů přidružených kontaktů.
3. Společnost je odebrána ze záznamů přidružených kategorií.
4. Společnost je odebrána ze záznamů přidružených společností.
5. Aktualizuje se seznam změn připravených k aktualizaci s propojenými systémy.
6. V případě zrušení požadavku na smazání záznamu nebude v databázi smazán ani změněn žádný záznam.

**Běžná cesta:**

5. Uživatel zvolí akci ke smazání v systému vedeného záznamu společnosti.
6. Systém se zeptá uživatele, zda chce záznam opravdu smazat.
7. Uživatel požadavek potvrdí.
8. Systém záznam vymaže.

**Alternativní cesty:**

2. Uživatel se rozhodne zrušit smazání záznamu (po kroku 2).
  - 2.1. Uživatel zvolí akci ke zrušení smazání záznamu.
  - 2.2. Systém zruší požadavek na smazání záznamu.

**Výjimky:** Nejsou.

**Vložené případy užití:** Nejsou.

**Zvláštní požadavky:** Nejsou.

### **3.2.7. Zobrazení přehledu společností**

**Případ užití číslo:** PU-3.7

**Název:** Zobrazení přehledu společností

**Aktéři:** Editor

**Popis:** Uživatel může zobrazit přehled společností, výsledky na tomto přehledu může filtrovat a řadit podle zvolených atributů.

**Vstupní podmínky:**

1. Uživatel je přihlášený do systému.

**Výstupní podmínky:**

1. Výsledky na stránce odpovídají zadanému filtru a typu řazení.

**Běžná cesta:**

1. Uživatel zvolí akci k zobrazení přehledu společností.
2. Systém zobrazí seznam společností, textové pole pro zadání řetězce pro filtrování výsledků a ovládání řazení u atributu Název.



3. Pokud je v systému vedeno více než 50 společností, systém na stránce zobrazí navigaci mezi stránkami společností.
- Alternativní cesty:**
1. Uživatel chce filtrovat výsledky v přehledu (po kroku 2 nebo 3).
    - 1.1. Uživatel vyplní řetězec pro filtrování výsledků.
    - 1.2. Systém zobrazí výsledky odpovídající zadanému filtru.
  2. Uživatel chce seřadit výsledky v přehledu (po kroku 2 nebo 3).
    - 2.1. Uživatel zvolí atribut a typ řazení (vzestupně, sestupně).
    - 2.2. Systém zobrazí výsledky seřazené podle zvoleného atributu a typu řazení.
  3. Uživatel chce přejít na jinou stránku (po kroku 3 běžné cesty a 1.2 nebo 2.2 alternativních cest).
    - 3.1. Uživatel zvolí akci přechodu na předchozí nebo následující stránku, nebo stránku podle čísla jejího pořadí.
    - 3.2. Systém zobrazí záznamy odpovídající stránce, na které se uživatel nachází.

**Výjimky:** Nejsou.

**Vložené případy užití:** Nejsou.

**Zvláštní požadavky:** Nejsou.

### 3.3. Funkční požadavky

Id požadavku	Priorita	Popis
Firma.Model	Nutné	Systém bude vést databázi společností a s nimi spojených údajů (viz <a href="#">datový model společnosti</a> )
Firma.Vytvořit	Nutné	Systém umožní vytvořit nový záznam společnosti.
Firma.Vytvořit.Společnosti	Nutné	Systém umožní přiřadit ke společnosti žádnou, jednu nebo více společností (asociací společností), jichž je členem.
Firma.Vytvořit.Zaměstnanci	Nutné	Systém umožní přiřadit ke společnosti žádný, jeden nebo více kontaktů.
Firma.Vytvořit.Kategorie	Nutné	Systém umožní přiřadit společnost do žádné, jedné nebo více kategorií společností.
Firma.Vytvořit.Validace	Nutné	Systém bude validovat e-mailovou adresu, telefonní číslo, webovou stránku a povinná pole.
Firma.Náhled	Nutné	Systém umožní zobrazit náhled společnosti.
Firma.Upravit	Nutné	Systém umožní upravit stávající záznam společnosti.
Firma.Upravit.Zrušit	Nutné	Pokud uživatel upravuje záznam společnosti, systém mu umožní ukončit editaci bez zachování provedených změn.
Firma.Smazat	Nutné	Systém umožní smazat záznam společnosti.
Firma.Smazat.Potvrzení	Nutné	Pokud uživatel požaduje smazání záznamu, systém požaduje potvrzení požadavku.
Firma.Smazat.Zrušit	Nutné	Uživatel může zrušit požadavek na smazání záznamu.

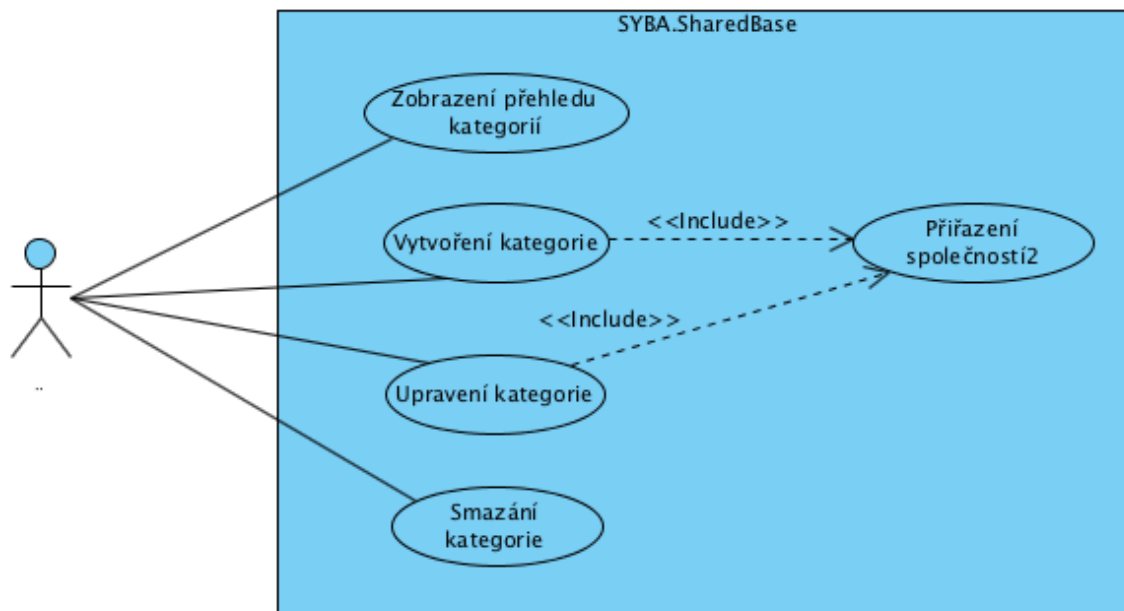
Firma.Smazat.Společnosti	Nutné	Smazání společnosti ponechá v systému záznam případných přidružených společností.
Firma.Smazat.Zaměstnanci	Nutné	Smazání společnosti ponechá v systému záznam případných přidružených kontaktů.
Firma.Smazat.Kategorie	Nutné	Smazání společnosti ponechá v systému záznamy případných přidružených kategorií společností.
Firma.Přehled	Nutné	Systém umožní zobrazit přehled všech společností.
Firma.Přehled.Filtr	Měl by mít	Systém bude filtrovat přehled společností na základě uživatelem zadaného textového filtru.
Firma.Přehled.Filtr.Kategorie	Mohl by mít	Systém zobrazí v přehledu pouze společnosti v uživatelem zvolené kategorii.
Firma.Přehled.Stránkování	Nutné	Systém bude stránkovat přehled společností v počtu 50 záznamů na stránku.
Firma.Přehled.Řazení	Měl by mít	Systém bude řadit společnosti v přehledu podle uživatelem zvoleného atributu (název, zkratka) a typu řazení (vzestupně, sestupně).

## 4. Správa kategorií společností

### 4.1. Popis

Systém bude vést databázi kategorií společností. Autorizovaný uživatel může vytvořit nový záznam kategorie, aktualizovat informace ve stávajícím záznamu, spravovat přiřazené společnosti nebo záznam smazat.

### 4.2. Případy užití



Obrázek 5 – PU: Správa kategorií společností

#### 4.2.1. Vytvoření kategorie společnosti

Případ užití číslo: PU-4.1

<b>Název:</b>	Vytvoření kategorie společností
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel se prostřednictvím Internetu přihlásí k systému. Zvolí akci vedoucí k vytvoření nového záznamu, vyplní informace o kategorii a systém tento záznam uloží do databáze.
<b>Vstupní podmínky:</b>	1. Uživatel je přihlášený do systému.
<b>Výstupní podmínky:</b>	1. V systému je veden nový záznam kategorie společností. 2. V případě zrušení zadávání nového záznamu (viz alternativní cesta číslo 1) nebude v databázi vytvořen žádný záznam.
<b>Běžná cesta:</b>	1. Uživatel zvolí akci k vytvoření nového záznamu kategorie. 2. Systém zobrazí formulář záznamu kategorie. 3. Uživatel vyplní formulář. 3.1. Uživatel vyplní informace o kategorii. 3.2. Systém provede případ užití PU-4.2 Přiřazení společností. 4. Uživatel systému sdělí, že záznam je hotový. 5. Systém validuje zadané informace o kategorii. 6. Validní záznam systém uloží do databáze. 7. Systém zobrazí editační formulář nově vytvořené kategorie.
<b>Alternativní cesty:</b>	1. Uživatel se rozhodne zrušit vytváření nového záznamu (kdykoli po kroku 1 a před krokem 6). 1.1. Uživatel zvolí akci ke zrušení nového záznamu. 1.2. Systém zruší zadávání a zobrazí stránku přehledu kategorií.
<b>Výjimky:</b>	1. Zadané informace o kategorii nejsou validní (po kroku 5). 1.1. Systém zobrazí chybová hlášení u špatně vyplněných údajů. 1.2. Uživatel opraví tyto údaje. 1.3. Systém se vrátí zpět na krok 5.
<b>Vložené případy užití:</b>	PU-4.2 Přiřazení společností
<b>Zvláštní požadavky:</b>	Nejsou.

#### **4.2.2. Přiřazení společností**

<b>Případ užití číslo:</b>	PU-4.2
<b>Název:</b>	Přiřazení společností
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel je ve stavu zadávání informací o kategorii. Z nabídky může zvolit společnosti, které mají být přiřazeny do této kategorie.
<b>Vstupní podmínky:</b>	1. Uživatel je přihlášený do systému. 2. Systém je ve stavu zadávání nebo úpravy záznamu kategorie.
<b>Výstupní podmínky:</b>	1. Ke kategorii je přiřazena žádná, jedna nebo více společností.
<b>Běžná cesta:</b>	1. Systém zobrazí seznam společností vedených v systému. 2. Uživatel vybere společnosti. 3. Systém uloží vztah kategorie a vybraných společností do databáze.

<b>Alternativní cesty:</b>	Nejsou.
<b>Výjimky:</b>	1. V systému nejsou vedeny žádné záznamy společností. 1.1. Konec případu užití.
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

#### **4.2.3. Upravení kategorie**

<b>Případ užití číslo:</b>	PU-4.3
<b>Název:</b>	Upravení kategorie
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel může změnit informace uložené v záznamu kategorie.
<b>Vstupní podmínky:</b>	1. Uživatel je přihlášený do systému. 2. Kategorie je vedena v systému.
<b>Výstupní podmínky:</b>	1. V databázi je uložen změněný záznam kategorie. 2. V případě zrušení úpravy záznamu (viz alternativní cesta číslo 1) nebude v databázi vytvořen ani změněn žádný záznam.
<b>Běžná cesta:</b>	1. Uživatel zvolí akci k úpravě v systému vedeného záznamu kategorie. 2. Systém zobrazí formulář záznamu. 3. Uživatel provede změny ve formuláři. 3.1. Uživatel upraví informace o kategorii. 3.2. Systém provede případ užití PU-4.2 Přiřazení společností. 4. Uživatel systému sdělí, že záznam je hotový. 5. Systém validuje zadané informace o kategorii. 6. Validní záznam systém uloží do databáze. 7. Systém zobrazí editační formulář kategorie.
<b>Alternativní cesty:</b>	1. Uživatel se rozhodne zrušit úpravu záznamu (kdykoli po kroku 1 a před krokem 6). 1.1. Uživatel zvolí akci ke zrušení úpravy záznamu. 1.2. Systém zruší editaci a zobrazí stránku přehledu kategorií.
<b>Výjimky:</b>	Nejsou.
<b>Vložené případy užití:</b>	PU-4.2 Přiřazení společností
<b>Zvláštní požadavky:</b>	Nejsou.

#### **4.2.4. Smazání kategorie**

<b>Případ užití číslo:</b>	PU-4.4
<b>Název:</b>	Smazání kategorie
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel může smazat záznam kategorie.
<b>Vstupní podmínky:</b>	1. Uživatel je přihlášený do systému. 2. Kategorie je vedena v systému.
<b>Výstupní podmínky:</b>	1. Záznam kategorie je vymazán.

2. Kategorie je odebrána ze záznamů přidružených společností.
3. V případě zrušení požadavku na smazání záznamu (viz alternativní cesta číslo 1) nebude v databázi smazán ani změněn žádný záznam.

**Běžná cesta:**

1. Uživatel zvolí akci ke smazání v systému vedeného záznamu kategorie.
2. Systém se zeptá uživatele, zda chce záznam opravdu smazat.
3. Uživatel požadavek potvrdí.
4. Systém záznam vymaže.

**Alternativní cesty:**

1. Uživatel se rozhodne zrušit smazání záznamu (po kroku 2).
  - 1.1. Uživatel zvolí akci ke zrušení smazání záznamu.
  - 1.2. Systém zruší požadavek na smazání záznamu.

**Výjimky:**

Nejsou.

**Vložené případy užití:**

Nejsou.

**Zvláštní požadavky:**

Nejsou.

#### **4.2.5. Zobrazení přehledu kategorií**

**Případ užití číslo:**

PU-3.7

**Název:**

Zobrazení přehledu společností

**Aktéři:**

Editor

**Popis:**

Uživatel může zobrazit přehled kategorií, výsledky na tomto přehledu jsou zobrazeny ve stromové struktuře a uživatel může měnit vzájemnou pozici jednotlivých kategorií.

**Vstupní podmínky:**

1. Uživatel je přihlášený do systému.

**Výstupní podmínky:**

Nejsou.

**Běžná cesta:**

1. Uživatel zvolí akci k zobrazení přehledu kategorií.
2. Systém zobrazí stromovou strukturu kategorií.
3. Uživatel chce změnit vzájemnou pozici kategorií.
  - 3.1. Uživatel změní pozice kategorií.
  - 3.2. Systém zobrazí výzvu k uložení nových pozic.
  - 3.3. Uživatel potvrdí uložení nových pozic.
  - 3.4. Systém uloží změny pozic.

**Alternativní cesty:**

1. Uživatel se rozhodne zrušit požadavek na změnu pozic (po kroku 3.2).
  - 1.1. Uživatel zvolí akci k zrušení nastavení nových pozic.
  - 1.2. Systém smaže změny pozic a navrátí vzhled stromu kategorií do původního stavu.

**Výjimky:**

Nejsou.

**Vložené případy užití:**

Nejsou.

**Zvláštní požadavky:**

Nejsou.

#### **4.3. Funkční požadavky**

**Id požadavku**

**Priorita**

**Popis**

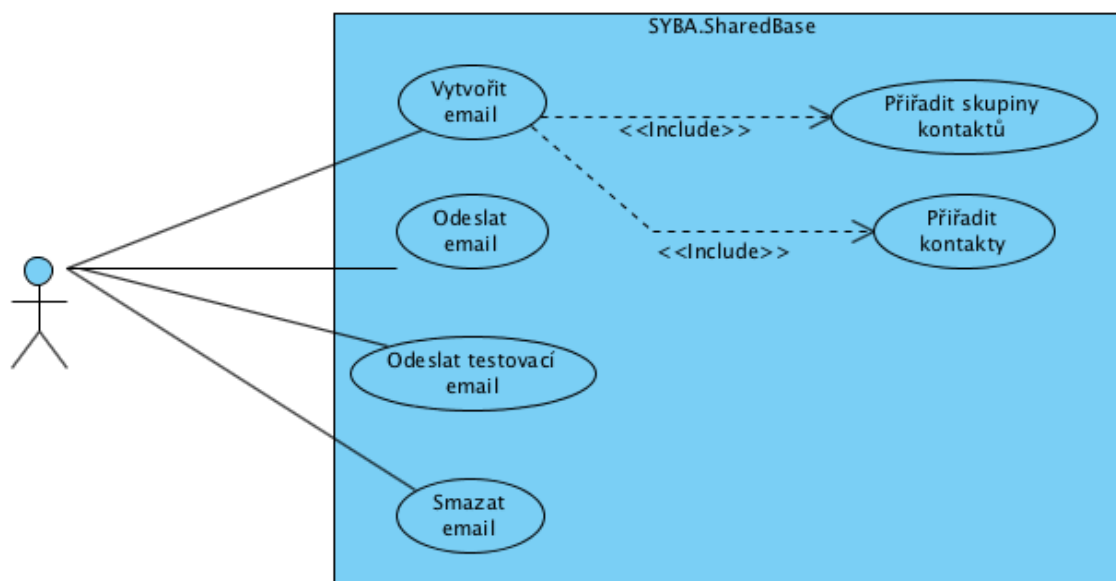
Kategorie.Model	Nutné	Systém bude vést databázi společností a s nimi spojených údajů (viz <u>datový model kategorie</u> )
Kategorie.Vytvořit	Nutné	Systém umožní vytvořit nový záznam kategorie.
Kategorie.Vytvořit.Společnosti	Nutné	Systém umožní přiřadit ke kategorii žádnou, jednu nebo více společností.
Kategorie.Vytvořit.Validace	Nutné	Systém bude validovat název kategorie.
Kategorie.Náhled	Nutné	Systém umožní zobrazit náhled kategorie.
Kategorie.Upravit	Nutné	Systém umožní upravit v systému vedený záznam kategorie.
Kategorie.Upravit.Zrušit	Nutné	Pokud uživatel upravuje záznam kategorie, systém mu umožní ukončit editaci bez zachování provedených změn.
Kategorie.Smazat	Nutné	Systém umožní smazat záznam kategorie.
Kategorie.Smazat.Potvrzení	Nutné	Pokud uživatel požaduje smazání záznamu, systém požaduje potvrzení požadavku.
Kategorie.Smazat.Zrušit	Nutné	Uživatel může zrušit požadavek na smazání záznamu.
Kategorie.Smazat.Společnosti	Nutné	Smazání kategorie ponechá v systému záznam případných přidružených společností.
Kategorie.Přehled	Nutné	Systém umožní zobrazit přehled všech kategorií v podobě stromu podle jejich vzájemných vztahů.
Kategorie.Přehled.Pozice	Nutné	Systém umožní změnu vzájemných pozic ve stromové struktuře kategorií.
Kategorie.Přehled.Pozice.Potvrď	Nutné	Pokud uživatel změní vzájemnou pozici kategorií ve stromové struktuře, systém zobrazí požadavek na potvrzení těchto změn.
Kategorie.Přehled.Pozice.Zrušit	Nutné	Pokud uživatel změní vzájemnou pozici kategorií ve stromové struktuře, systém uživateli umožní návrat k původní struktuře.

## 5. Odesílání hromadné korespondence

### 5.1. Popis

Systém bude napojen na externí službu na zasílání hromadné e-mailové korespondence. Pomocí této služby bude autorizovanému uživateli umožňovat rozesílání pošty na adresy označené pro rozesílání hromadné korespondence.

## 5.2. Případy užití



Obrázek 6 – PU: Odesílání hromadné korespondence

### 5.2.1. Vytvoření e-mailu

**Případ užití číslo:** PU-5.1

**Název:** Vytvoření e-mailu

**Akteři:** Editor

**Popis:** Uživatel se prostřednictvím Internetu přihlásí k systému. Zvolí akci vedoucí k vytvoření nového záznamu, vyplní informace o e-mailu a systém tento záznam uloží do databáze.

**Vstupní podmínky:** 1. Uživatel je přihlášený do systému.

**Výstupní podmínky:** 1. V systému je veden nový záznam e-mailu.  
2. V případě zrušení zadávání nového záznamu (viz alternativní cesta číslo 1) nebude v databázi vytvořen žádný záznam.

**Běžná cesta:** 1. Uživatel zvolí akci k vytvoření nového záznamu e-mailu.  
2. Systém zobrazí formulář záznamu.  
3. Uživatel vyplní formulář.  
3.1. Uživatel vyplní informace o e-mailu.  
3.2. Systém provede případ užití PU-5.2 Přiřazení skupin kontaktů.  
3.3. Systém provede případ užití PU-5.3 Přiřazení kontaktů.  
4. Uživatel systému sdělí, že záznam je hotový.  
5. Systém validuje zadané informace o e-mailu.  
6. Validní záznam systém uloží do databáze.

7. Systém zobrazí editační formulář nově vytvořeného e-mailu.

**Alternativní cesty:** 1. Uživatel se rozhodne zrušit vytváření nového záznamu (kdykoli po kroku 1 a před krokem 6).

1.1. Uživatel zvolí akci ke zrušení nového záznamu.

1.2. Systém zruší zadávání a zobrazí stránku přehledu kategorií.

- 2. Uživatel chce odeslat testovací e-mail.
    - 2.1. Systém provede případ užití PU-5.4 Odeslání testovacího e-mailu.
  - 3. Uživatel chce odeslat e-mail.
    - 3.1. Systém provede případ užití PU-5.5 Odeslání e-mailu.
- Výjimky:**
- 1. Zadané informace nejsou validní (po kroku 5).
    - 1.1. Systém zobrazí chybová hlášení u špatně vyplněných údajů.
    - 1.2. Uživatel opraví tyto údaje.
    - 1.3. Systém se vrátí zpět na krok 5.
- Vložené případy užití:** PU-5.2 Přiřazení skupin kontaktů  
PU-5.3 Přiřazení kontaktů
- Zvláštní požadavky:** Nejsou.

### 5.2.2. Přiřazení skupin kontaktů

- Případ užití číslo:** PU-5.2
- Název:** Přiřazení skupin kontaktů
- Aktéři:** Editor
- Popis:** Uživatel je ve stavu zadávání informací o e-mailu. Z nabídky může zvolit skupiny kontaktů, které mají být přiřazeny mezi příjemce tohoto e-mailu.
- Vstupní podmínky:**
- 1. Uživatel je přihlášený do systému.
  - 2. Systém je ve stavu zadávání, úpravy nebo přípravy k odeslání e-mailu.
- Výstupní podmínky:** 1. K e-mailu je přiřazena žádná, jedna nebo více skupin kontaktů.
- Běžná cesta:**
- 1. Systém zobrazí seznam skupin kontaktů vedených v systému.
  - 2. Uživatel vybere skupiny kontaktů.
  - 3. Systém uloží vztah e-mailu a vybraných skupin kontaktů do databáze.
- Alternativní cesty:** Nejsou.
- Výjimky:**
- 1. V systému nejsou vedeny žádné záznamy skupiny kontaktů.
    - 1.1. Konec případu užití.
- Vložené případy užití:** Nejsou.
- Zvláštní požadavky:** Nejsou.

### 5.2.3. Přiřazení kontaktů

- Případ užití číslo:** PU-5.3
- Název:** Přiřazení kontaktů
- Aktéři:** Editor
- Popis:** Uživatel je ve stavu zadávání informací o e-mailu. Z nabídky může zvolit kontakty, které mají být přiřazeny mezi příjemce tohoto e-mailu.
- Vstupní podmínky:**
- 1. Uživatel je přihlášený do systému.
  - 2. Systém je ve stavu zadávání, úpravy nebo přípravy k odeslání e-



	mailu.
<b>Výstupní podmínky:</b>	1. K e-mailu je přiřazen žádný, jeden nebo více kontaktů.
<b>Běžná cesta:</b>	1. Systém zobrazí seznam kontaktů vedených v systému. 2. Uživatel vybere kontakty. 3. Systém uloží vztah e-mailu a vybraných kontaktů do databáze.
<b>Alternativní cesty:</b>	Nejsou.
<b>Výjimky:</b>	1. V systému nejsou vedeny žádné záznamy kontaktů. 1.2. Konec případu užití.
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

#### **5.2.4. Odeslání testovacího e-mailu**

<b>Případ užití číslo:</b>	PU-5.4
<b>Název:</b>	Odeslání testovacího e-mailu
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel může odeslat testovací e-mail.
<b>Vstupní podmínky:</b>	1. Uživatel je přihlášený do systému. 2. E-mail je vytvořen v systému.
<b>Výstupní podmínky:</b>	1. E-mail je odeslán na vnější e-mailovou službu, výsledek odeslání je uložen v systému u záznamu e-mailu. 2. V případě zrušení odeslání testovacího e-mailu (viz alternativní cesta číslo 1) nebude v databázi vytvořen ani změněn žádný záznam.
<b>Běžná cesta:</b>	1. Uživatel zvolí akci k odeslání testovacího e-mailu. 2. Systém požádá o zadání e-mailové adresy příjemce testovacího e-mailu. 3. Uživatel zadá e-mailovou adresu příjemce. 4. Uživatel zvolí akci odeslání. 5. Systém odešle testovací e-mail.
<b>Alternativní cesty:</b>	1. Uživatel se rozhodne zrušit odeslání testovacího e-mailu (po kroku 2). 1.1. Uživatel zvolí akci ke zrušení odeslání testovacího e-mailu. 1.2. Systém zruší odeslání testovacího e-mailu a zobrazí detail testovacího e-mailu.
<b>Výjimky:</b>	Nejsou.
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

#### **5.2.5. Odeslání e-mailu**

<b>Případ užití číslo:</b>	PU-5.5
<b>Název:</b>	Odeslání e-mailu
<b>Aktéři:</b>	Editor

<b>Popis:</b>	Uživatel může odeslat e-mail.
<b>Vstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Uživatel je přihlášený do systému.</li> <li>2. E-mail je vytvořen v systému.</li> <li>3. E-mail má uveden jednoho nebo více příjemců.</li> </ol>
<b>Výstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. E-mail je odeslán na vnější e-mailovou službu, výsledek odeslání je uložen v systému u záznamu e-mailu.</li> <li>2. V případě zrušení odeslání e-mailu (viz alternativní cesta číslo 1) nebude v databázi vytvořen ani změněn žádný záznam.</li> </ol>
<b>Běžná cesta:</b>	<ol style="list-style-type: none"> <li>1. Uživatel zvolí akci k odeslání e-mailu.</li> <li>2. Systém zobrazí požadavek na potvrzení odeslání.</li> <li>3. Uživatel požadavek potvrdí.</li> <li>4. Systém odešle e-mail.</li> </ol>
<b>Alternativní cesty:</b>	<ol style="list-style-type: none"> <li>1. Uživatel se rozhodne zrušit odeslání e-mailu (po kroku 2). <ol style="list-style-type: none"> <li>1.1. Uživatel zvolí akci ke zrušení odeslání e-mailu.</li> <li>1.2. Systém zruší odeslání e-mailu a zobrazí detail e-mailu.</li> </ol> </li> </ol>
<b>Výjimky:</b>	Nejsou.
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

#### 5.2.6. Smazání e-mailu

<b>Případ užití číslo:</b>	PU-5.6
<b>Název:</b>	Smazání e-mailu
<b>Aktéři:</b>	Editor
<b>Popis:</b>	Uživatel může smazat záznam e-mailu.
<b>Vstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Uživatel je přihlášený do systému.</li> <li>2. E-mail je veden v systému.</li> </ol>
<b>Výstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Záznam e-mailu je vymazán.</li> <li>2. V případě zrušení požadavku na smazání záznamu (viz alternativní cesta číslo 1) nebude v databázi smazán ani změněn žádný záznam.</li> </ol>
<b>Běžná cesta:</b>	<ol style="list-style-type: none"> <li>1. Uživatel zvolí akci ke smazání v systému vedeného záznamu e-mailu.</li> <li>2. Systém se zeptá uživatele, zda chce záznam opravdu smazat.</li> <li>3. Uživatel požadavek potvrdí.</li> <li>4. Systém záznam vymaže.</li> </ol>
<b>Alternativní cesty:</b>	<ol style="list-style-type: none"> <li>1. Uživatel se rozhodne zrušit smazání záznamu (po kroku 2). <ol style="list-style-type: none"> <li>1.1. Uživatel zvolí akci ke zrušení smazání záznamu.</li> <li>1.2. Systém zruší požadavek na smazání záznamu.</li> </ol> </li> </ol>
<b>Výjimky:</b>	Nejsou.
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

### 5.3. Funkční požadavky

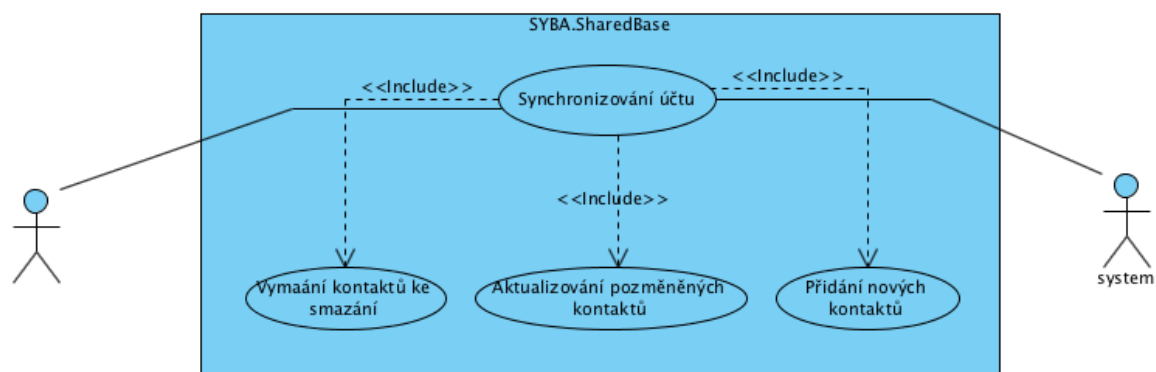
Id požadavku	Priorita	Popis
Email.Model	V dalších verzích	Systém bude vést databázi společností a s nimi spojených údajů (viz <a href="#">datový model e-mailu</a> )
Email.Vytvořit	V dalších verzích	Systém umožní vytvořit nový záznam e-mailu.
Email.Vytvořit.Skupiny	V dalších verzích	Systém umožní přiřadit k e-mailu žádnou, jednu nebo více skupin kontaktů.
Email.Vytvořit.Kontakty	V dalších verzích	Systém umožní přiřadit k e-mailu žádný, jeden nebo více kontaktů.
Email.Vytvořit.Náhled	V dalších verzích	Systém umožní zobrazit náhled e-mailu před odesláním.
Email.Test	V dalších verzích	Systém umožní odeslat testovací e-mail na uživatelem zadanou e-mailovou adresu.
Email.Rozeslat	V dalších verzích	Systém umožní rozeslat e-mail na zadané kontakty a skupiny kontaktů prostřednictvím vnější e-mailové služby.
Email.Zobrazit	V dalších verzích	Systém umožní zobrazit detail dříve vytvořeného e-mailu.
Email.Smazat	V dalších verzích	Systém umožní smazání dříve vytvořeného e-mailu.
Email.Smazat.Potvrzení	V dalších verzích	Pokud uživatel požaduje smazání záznamu, systém požaduje potvrzení požadavku.
Kategorie.Smazat.Zrušit	V dalších verzích	Uživatel může zrušit požadavek na smazání záznamu.
Email.Přehled	V dalších verzích	Systém umožní zobrazení přehledu vytvořených e-mailů.

## 6. Synchronizace se službou Google Contacts

### 6.1. Popis

Systém bude zaznamenávat změny v záznamech a periodicky spouštět automatickou synchronizaci kontaktů a skupin kontaktů v systému s účty ve službě Google Contacts. Autorizovanému uživateli umožní provést manuální synchronizaci.

## 6.2. Případy užití



Obrázek 7 – PU: Synchronizace se službou Google Contacts

### 6.2.1. Synchronizování účtu

**Případ užití číslo:** PU-6.1

**Název:** Synchronizování účtu

**Akteři:** Editor, Systém

**Popis:** Uživatel zvolí akci vedoucí k synchronizaci dat v systému se zvoleným účtem ve službě Google Contacts. Systém provede synchronizaci kontaktů.

**Vstupní podmínky:**

1. Uživatel je přihlášený do systému.
2. V systému je veden účet ke službě Google Contacts.

**Výstupní podmínky:**

1. V systému je veden nový stav synchronizace s daným účtem.
  - 1.1. V případě nekompletního provedení synchronizace je u účtu veden přehled doposud nesyndronizovaných kontaktů a skupin kontaktů.

**Běžná cesta:**

1. Uživatel zvolí akci k provedení synchronizace s daným účtem ve službě Google Contacts.
2. Systém spustí provádění synchronizace.
  - 2.1. Systém provede případ užití PU-6.2 Vymazání kontaktů.
  - 2.2. Systém provede případ užití PU-6.3 Aktualizování kontaktů.
  - 2.3. Systém provede případ užití PU-6.4 Přidání kontaktů.
3. Systém zobrazí editační formulář nově vytvořeného e-mailu.

**Alternativní cesty:**

1. Uživatel se rozhodne zrušit synchronizaci (během kroku 2).
  - 1.1. Uživatel zvolí akci ke zrušení synchronizace.
  - 1.2. Systém přeruší zruší synchronizaci a zobrazí stav.

**Výjimky:** Nejsou.

**Vložené případy užití:** PU-6.2 Vymazání kontaktů  
PU-6.3 Aktualizování kontaktů  
PU-6.4 Přidání kontaktů

**Zvláštní požadavky:** Nejsou.

### **6.2.2. Vymazání kontaktů**

<b>Případ užití číslo:</b>	PU-6.2
<b>Název:</b>	Vymazání kontaktů
<b>Aktéři:</b>	System
<b>Popis:</b>	System provádí synchronizaci smazaných kontaktů s účtem na službě Google Contacts.
<b>Vstupní podmínky:</b>	<ol style="list-style-type: none"><li>1. Je spuštěna synchronizace s účtem na službě Google Contacts.</li><li>2. Kontakty v systému označené ke smazání, které ještě nebyly vymazány z daného účtu.</li></ol>
<b>Výstupní podmínky:</b>	<ol style="list-style-type: none"><li>1. Z účtu na službě Google Contacts byly vymazány všechny kontakty označené v systému k vymazání.</li></ol>
<b>Běžná cesta:</b>	<ol style="list-style-type: none"><li>1. System vymazává kontakty z účtu Google.</li><li>2. System zobrazuje průběh vymazávání kontaktů.</li></ol>
<b>Alternativní cesty:</b>	<ol style="list-style-type: none"><li>1. Uživatel se rozhodne zrušit synchronizaci.<ol style="list-style-type: none"><li>1.1. Uživatel zvolí akci ke zrušení synchronizace.</li><li>1.2. System přeruší zruší synchronizaci a zobrazí stav.</li></ol></li></ol>
<b>Výjimky:</b>	Nejsou.
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

### **6.2.3. Aktualizace kontaktů**

<b>Případ užití číslo:</b>	PU-6.3
<b>Název:</b>	Aktualizace kontaktů
<b>Aktéři:</b>	System
<b>Popis:</b>	System provádí synchronizaci kontaktů změněných od poslední synchronizace s účtem na službě Google Contacts.
<b>Vstupní podmínky:</b>	<ol style="list-style-type: none"><li>1. Je spuštěna synchronizace s účtem na službě Google Contacts.</li><li>2. Kontakty v systému, které byly upraveny a které ještě nebyly synchronizovány s daným účtem, jsou aktualizovány na daném účtu.</li></ol>
<b>Výstupní podmínky:</b>	<ol style="list-style-type: none"><li>1. Na účtu na službě Google Contacts byly aktualizovány všechny kontakty upravené v systému.</li></ol>
<b>Běžná cesta:</b>	<ol style="list-style-type: none"><li>1. System aktualizuje kontakty na účtu Google.</li><li>2. System zobrazuje průběh aktualizace kontaktů.</li></ol>
<b>Alternativní cesty:</b>	<ol style="list-style-type: none"><li>1. Uživatel se rozhodne zrušit synchronizaci.<ol style="list-style-type: none"><li>1.1. Uživatel zvolí akci ke zrušení synchronizace.</li><li>1.2. System přeruší zruší synchronizaci a zobrazí stav.</li></ol></li></ol>
<b>Výjimky:</b>	Nejsou.
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

**6.2.4. Přidání kontaktů**

<b>Případ užití číslo:</b>	PU-6.4
<b>Název:</b>	Přidání kontaktů
<b>Aktéři:</b>	System
<b>Popis:</b>	System provádí synchronizaci kontaktů přidaných do systému od poslední synchronizace s účtem na službě Google Contacts.
<b>Vstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Je spuštěna synchronizace s účtem na službě Google Contacts.</li> <li>2. V systému jsou nové kontakty, které ještě nebyly synchronizovány s daným účtem.</li> </ol>
<b>Výstupní podmínky:</b>	<ol style="list-style-type: none"> <li>1. Na účet na službě Google Contacts byly přidány všechny nové kontakty.</li> </ol>
<b>Běžná cesta:</b>	<ol style="list-style-type: none"> <li>1. System nahrává nové kontakty na účet Google.</li> <li>2. System zobrazuje průběh nahrávání kontaktů.</li> </ol>
<b>Alternativní cesty:</b>	<ol style="list-style-type: none"> <li>1. Uživatel se rozhodne zrušit synchronizaci. <ol style="list-style-type: none"> <li>1.1. Uživatel zvolí akci ke zrušení synchronizace.</li> <li>1.2. System přeruší zruší synchronizaci a zobrazí stav.</li> </ol> </li> </ol>
<b>Výjimky:</b>	Nejsou.
<b>Vložené případy užití:</b>	Nejsou.
<b>Zvláštní požadavky:</b>	Nejsou.

**6.3. Funkční požadavky**

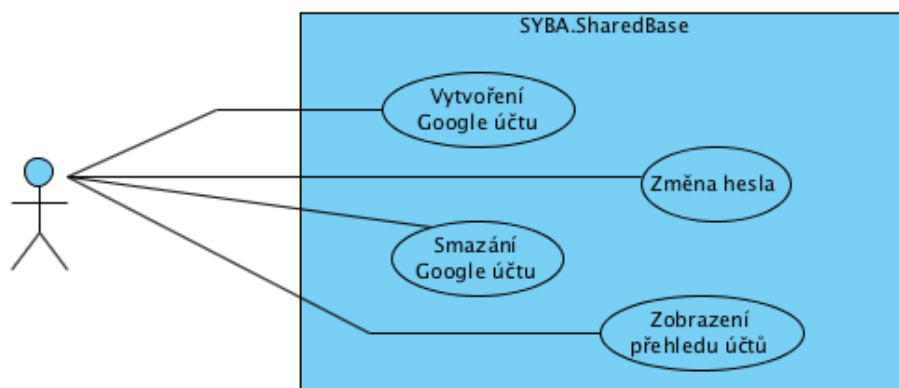
<b>Id požadavku</b>	<b>Priorita</b>	<b>Popis</b>
Synchronizace	V příštích verzích	System bude automaticky v časových intervalech hlídat změny a spouštět synchronizace.
Synchronizace.Spustit	Nutné	System umožní uživateli manuálně vyvolat synchronizaci s jednotlivými účty.
Synchronizace.Průběh	Měl by mít	System bude zobrazovat průběh synchronizace u jednotlivých účtů.
Synchronizace.Zrušit	Měl by mít	System umožní zrušit probíhající synchronizaci.
Synchronizace.Zrušit.Potvrzení	Měl by mít	Pokud uživatel požaduje zrušení probíhající synchronizace, system požaduje potvrzení požadavku.
Synchronizace.Smazat	Měl by mít	System umožní smazat veškeré kontakty na straně účtu Google.
Synchronizace.Smazat.Potvrzení	Měl by mít	Pokud uživatel požaduje smazání záznamu, system požaduje potvrzení požadavku.
Synchronizace.Smazat.Zrušit	Měl by mít	Uživatel může zrušit požadavek na smazání záznamu.

## 7. Správa Google účtů

### 7.1. Popis

Systém bude udržovat databázi účtů Google pro synchronizaci kontaktů a skupin kontaktů. Autorizovanému uživateli umožní vytvořit záznam účtu Google, změnit heslo k účtu nebo smazat záznam.

### 7.2. Případy užití



Obrázek 8 – PU: Správa společností

#### 7.2.1. Vytvoření Google účtu

**Případ užití číslo:** PU-7.1

**Název:** Vytvoření Google účtu

**Aktéři:** Editor

**Popis:** Uživatel se prostřednictvím Internetu přihlásí k systému. Zvolí akci vedoucí k vytvoření nového záznamu, vyplní přihlašovací jméno a heslo a systém tento záznam uloží do databáze.

**Vstupní podmínky:** 1. Uživatel je přihlášený do systému.

**Výstupní podmínky:** 1. V systému je veden nový záznam Google účtu.  
2. V případě zrušení zadávání nového záznamu (viz alternativní cesta číslo 1) nebude v databázi vytvořen žádný záznam.

**Běžná cesta:** 1. Uživatel zvolí akci k vytvoření nového záznamu Google účtu.

2. Systém zobrazí formulář záznamu.
3. Uživatel vyplní přihlašovací jméno a heslo k účtu.
4. Uživatel systému sdělí, že záznam je hotový.
5. Systém validuje zadané informace o Google účtu.
6. Validní záznam systém uloží do databáze.
7. Systém zobrazí přehled Google účtů v systému.

**Alternativní cesty:** 1. Uživatel se rozhodne zrušit vytváření nového záznamu (kdykoli po kroku 1 a před krokem 6).

- 1.1. Uživatel zvolí akci ke zrušení nového záznamu.
- 1.2. Systém zruší zadávání a zobrazí stránku přehledu společností.

**Výjimky:** 1. Zadané přihlašovací údaje nejsou validní (po kroku 5).

- 1.1. Systém zobrazí chybová hlášení u špatně vyplněných údajů.
- 1.2. Uživatel opraví tyto údaje.
- 1.3. Systém se vrátí zpět na krok 5.

**Vložené případy užití:** Nejsou.

**Zvláštní požadavky:** Nejsou.

### 7.2.2. Změna hesla

**Případ užití číslo:** PU-7.2

**Název:** Změna hesla

**Aktéři:** Editor

**Popis:** Uživatel může změnit heslo Google účtu.

**Vstupní podmínky:**

1. Uživatel je přihlášený do systému.
2. V systému existuje daný záznam Google účtu.

**Výstupní podmínky:** 1. Heslo účtu bylo změněno.

**Běžná cesta:**

1. Uživatel vybere akci ke změně hesla účtu.
2. Systém zobrazí formulář.
3. Uživatel zadá dvakrát nové heslo k účtu.
4. Systém validuje zadaná hesla účtu.
5. Validní záznam systém uloží do databáze.
6. Systém zobrazí přehled Google účtů v systému.

**Alternativní cesty:** Nejsou.

**Výjimky:**

1. Zadané přihlašovací údaje nejsou validní (po kroku 4).
  - 1.1. Systém zobrazí chybová hlášení u špatně vyplněných údajů.
  - 1.2. Uživatel opraví tyto údaje.
  - 1.3. Systém se vrátí zpět na krok 4.

**Vložené případy užití:** Nejsou.

**Zvláštní požadavky:** Nejsou.

### 7.2.3. Smazání účtu

**Případ užití číslo:** PU-7.3

**Název:** Smazání účtu

**Aktéři:** Editor

**Popis:** Uživatel může smazat záznam Google účtu.

**Vstupní podmínky:**

1. Uživatel je přihlášený do systému.
2. V systému existuje daný záznam Google účtu.

**Výstupní podmínky:**

1. Záznam účtu je vymazán.
2. V případě zrušení požadavku na smazání záznamu (viz alternativní cesta číslo 1) nebude v databázi smazán ani změněn žádný záznam.

**Běžná cesta:**

1. Uživatel zvolí akci ke smazání v systému vedeného záznamu Google účtu.
2. Systém se zeptá uživatele, zda chce záznam opravdu smazat.



3. Uživatel požadavek potvrdí.
  4. Systém záznam vymaže.
- Alternativní cesty:**
1. Uživatel se rozhodne zrušit smazání záznamu (po kroku 2).
    - 1.1. Uživatel zvolí akci ke zrušení smazání záznamu.
    - 1.2. Systém zruší požadavek na smazání záznamu.

**Výjimky:** Nejsou.

**Vložené případy užití:** Nejsou.

**Zvláštní požadavky:** Nejsou.

#### **7.2.4. Zobrazení přehledu účtů**

**Případ užití číslo:** PU-7.4

**Název:** Zobrazení přehledu účtů

**Aktéři:** Editor

**Popis:** Uživatel může zobrazit přehled v systému vedených Google účtů.

**Vstupní podmínky:** 1. Uživatel je přihlášený do systému.

**Výstupní podmínky:** Nejsou.

**Běžná cesta:**

1. Uživatel zvolí akci k zobrazení přehledu účtů.
2. Systém zobrazí seznam účtů.

**Alternativní cesty:** Nejsou.

**Výjimky:** Nejsou.

**Vložené případy užití:** Nejsou.

**Zvláštní požadavky:** Nejsou.

### **7.3. Funkční požadavky**

<b>Id požadavku</b>	<b>Priorita</b>	<b>Popis</b>
ÚčetGoogle.Přidat	Nutné	Systém umožní přidat nový záznam účtu Google.
ÚčetGoogle.Přidat.Email	Nutné	Pokud uživatel zadá neplatnou e-mailovou adresu, systém zobrazí upozornění a požaduje nápravu.
ÚčetGoogle.Přidat.Heslo	Nutné	Systém bude požadovat po uživateli zadat heslo k účtu pro kontrolu dvakrát.
ÚčetGoogle.Přidat.Chyba	Nutné	Pokud se dvě zadaná hesla neshodují, systém zobrazí upozornění a požaduje nápravu.
ÚčetGoogle.Přidat.Kontrola	Měl by mít	Systém provede zkušební připojení k účtu Google pro kontrolu zadaných údajů.
ÚčetGoogle.Přidat.Kontrola. Oprava	Měl by mít	V případě chyby při kontrole připojení k účtu, systém požaduje opravu údajů.
ÚčetGoogle.Změnit	Měl by mít	Systém umožní změnit heslo účtu Google.
ÚčetGoogle.Změnit.Heslo	Měl by mít	Systém bude požadovat po uživateli zadat nové heslo k účtu pro kontrolu dvakrát.
ÚčetGoogle.Změnit.Chyba	Měl by mít	Pokud se dvě zadaná hesla neshodují, systém zobrazí upozornění a požaduje nápravu.

ÚčetGoogle.Změnit.Zrušit	Měl by mít	Pokud uživatel upravuje skupinu kontaktů, systém mu umožní ukončit editaci bez zachování provedených změn.
ÚčetGoogle.Smazat	Nutné	Systém umožní smazat účet.
ÚčetGoogle.Smazat.Potvrzení	Nutné	Pokud uživatel požaduje smazání účtu, systém požaduje potvrzení požadavku.
ÚčetGoogle.Smazat.Zrušit	Nutné	Uživatel může zrušit požadavek na smazání účtu.

## Datový model

### Kontakt

Každý záznam kontaktu bude obsahovat alespoň tyto vlastnosti:

Vlastnost	Typ	Popis
křestní jméno	řetězec znaků	-
příjmení	řetězec znaků	-
druhé jméno	řetězec znaků	-
titul před jménem	řetězec znaků	-
titul za jménem	řetězec znaků	-
narozeniny	datum	-
poznámka	řetězec znaků	-
název banky	řetězec znaků	-
číslo bankovního účtu	řetězec znaků	-
mezinárodní číslo bankovního účtu	řetězec znaků	Číslo bankovního účtu ve tvaru IBAN.
zaměstnavatel	typ Společnost	Společnost, ve které je osoba kontaktu zaměstnána.
adresa	typ Adresa	-
adresa profilového obrázku	řetězec znaků	URL adresa obrázku.
e-mailové adresy	typ Email	-
telefonní čísla	typ Telefon	-
skupiny	typ Skupina	-

### Adresa

Každý záznam adresy bude obsahovat alespoň tyto vlastnosti:

Vlastnost	Typ	Popis
název ulice	řetězec znaků	-
město	řetězec znaků	-
země	řetězec znaků	-
region	řetězec znaků	-
poštovní směrovací číslo	řetězec znaků	-

P.O. box	řetězec znaků	-
zeměpisná šířka	řetězec znaků	-
zeměpisná délka	řetězec znaků	-

## Telefonní číslo

Každý záznam telefonního čísla bude obsahovat alespoň tyto vlastnosti:

Vlastnost	Typ	Popis
telefonní číslo	řetězec znaků	Vlastní hodnota telefonního čísla.
typ	řetězec znaků	Typ telefonního čísla: Mobilní, Do práce, Domů, Hlavní, Fax do práce, Fax domů, Google Voice, Pager, Jiný.

## E-mailová adresa

Každý záznam e-mailové adresy bude obsahovat alespoň tyto vlastnosti:

Vlastnost	Typ	Popis
e-mailová adresa	řetězec znaků	Vlastní hodnota e-mailové adresy.
typ	řetězec znaků	Typ e-mailové adresy: Osobní, Do práce, Jiný.
odesílat hromadné e-maily	boolean	True/false hodnota označující, zda je tato adresa určena k rozesílání hromadné korespondence.

## Skupina kontaktů

Každý záznam skupiny kontaktů bude obsahovat alespoň tyto vlastnosti:

Vlastnost	Typ	Popis
název skupiny	řetězec znaků	-
kontakty ve skupině	typ Kontakt	Seznam všech přidružených kontaktů.

## Společnost

Každý záznam společnosti bude obsahovat alespoň tyto vlastnosti:

Vlastnost	Typ	Popis
název společnosti	řetězec znaků	-
zkratka společnosti	řetězec znaků	-
popis společnosti	řetězec znaků	-
popis společnosti anglicky	řetězec znaků	-
výrobní program	řetězec znaků	-
výrobní program anglicky	řetězec znaků	-
identifikační číslo	řetězec znaků	-
daňové identifikační číslo	řetězec znaků	-
webová adresa	řetězec znaků	-
kontaktní telefon	řetězec znaků	-

kontaktní e-mail	řetězec znaků	-
datová schránka	řetězec znaků	-
adresa	typ Adresa	-
adresa loga	řetězec znaků	URL adresa obrázku loga společnosti.
zaměstnanci	typ Kontakt	-
organizace	typ Společnost	Seznam společností/asociací, jichž je tato společnost členem.
kategorie	typ Kategorie	Seznam přiřazených kategorií.

## Kategorie

Každý záznam kategorie bude obsahovat alespoň tyto vlastnosti:

Vlastnost	Typ	Popis
název kategorie	řetězec znaků	-
název kategorie anglicky	řetězec znaků	-
pozice	řetězec znaků	Řetězec určující kategorii ve stromu kategorií (př. 1, 1.1, 1.2, atp.).
společnosti	typ Společnost	Seznam společností v této kategorii.

## E-mail

Každý záznam e-mailu bude obsahovat alespoň tyto vlastnosti:

Vlastnost	Typ	Popis
datum odeslání	datum	-
jméno odesílatele	řetězec znaků	-
e-mail odesílatele	řetězec znaků	-
e-mail na odpověď	řetězec znaků	-
předmět zprávy	řetězec znaků	-
obsah zprávy	řetězec znaků	-
testovací adresa	řetězec znaků	-
ukazatel, zda byl test odeslán	řetězec znaků	-
skupiny kontaktů	typ Skupina	Skupiny kontaktů pro příjem e-mailu.
kontakty	typ Kontakt	Další kontakty pro příjem e-mailu.

## Požadavky na vnější rozhraní

### Uživatelská rozhraní

- UR-1 Uživatelské rozhraní systému musí být responzivní, aby byla zaručena jeho použitelnost na mobilních telefonech a tabletech.
- UR-2 Uživatelské rozhraní musí být uzpůsobené na dotyková zařízení.
- UR-3 Vzhled uživatelského rozhraní a user experience musí být podobný službě Google Kontakty, na kterou jsou zaměstnanci zadavatele zvyklí.

UR-4 Vzhled uživatelského rozhraní by měl odpovídat metodice Google Material Design<sup>1</sup>.

## Softwarová rozhraní

SR-1 Rozhraní ke službě Google Contacts:

- SR-1.1 Systém bude automaticky poskytovat službě Google Contacts přidané, upravené a smazané kontakty.
- SR-1.2 Systém bude automaticky poskytovat službě Google Contacts přidané, upravené a smazané skupiny kontaktů.
- SR-1.3 Před první synchronizací se službou Google Contacts systém vymaže všechny kontakty v daném účtu na této službě, aby zabránil vzniku duplicit.

SR-2 Rozhraní k e-mailové službě MailChimp:

- SR-2.1 Systém bude rozesílat prostřednictvím služby MailChimp hromadnou e-mailovou korespondenci.
- SR-2.2 Systém se bude služby MailChimp ptát na výsledky e-mailové kampaně.

SR-3 Rozhraní datové služby SYBA SharedBase:

- SR-3.1 Systém bude poskytovat správu veškerých informací vedených v systému prostřednictvím zabezpečeného API.

---

<sup>1</sup> Zásady a doporučení pro uživatelské rozhraní vyvíjené společností Google: <http://www.google.com/design/>.

# Návrh

Tato kapitola obsahuje softwarový návrh systému SYBA SharedBase upřesňující analytický model (Analýza) ve světle skutečného implementačního prostředí. V první části (Volba implementačního prostředí) jsou vyjmenovány frameworky a nástroje zvolené pro implementaci navrhovaného systému. Následuje architektonický návrh systému a detailní návrh jednotlivých modulů (Návrhový model).

Předmětem této návrhové dokumentace je systém verze 1.0 vymezený v sekci specifikace požadavků (Specifikace požadavků). Uvedené návrhové diagramy jsou zamýšleny pouze jako popis fungování a architektury systému, nikoli jako podklad pro poloautomatický přepis do kódu. Není jich proto tolik a v některých částech jsou modelovány spíše na vyšší úrovni abstrakce.

## Volba implementačního prostředí

### Serverová část

#### ASP.NET Web API

ASP.NET Web API je framework pro vytváření webových služeb založených na protokolu HTTP na platformě .NET. Umožňuje snadno definovat rozhraní a data na poskytované službě (Microsoft, 2014).

#### Fluent NHibernate

NHibernate je open-source řešení na objektově-relační mapování<sup>1</sup> pro .NET framework. Poskytuje framework pro mapování objektově orientovaného doménového modelu do relační databáze. Fluent NHibernate je nadstavba pro Hibernate umožňující automatické, konvencemi řízené mapování (NHibernate, 2014).

#### Castle Windsor

Castle Windsor je nástroj pro vkládání závislostí<sup>2</sup> pro platformu .NET spravovaný open-source komunitou projektu Castle (Kozmic, 2014).

### Uživatelské rozhraní

#### AngularJS

AngularJS je javascriptový framework určený pro vytváření takzvaných single-page aplikací (SPA)<sup>1</sup>. Jedná se o klientský MVC framework, jehož cílem je usnadnění vývoje a testování SPA

---

<sup>1</sup> Softwarová technika zajišťující automatickou konverzi dat mezi objektově orientovaným doménovým modelem a relační databází (Objektově relační mapování, 2014).

<sup>2</sup> Anglicky Dependench injection – technika vkládání závislostí mezi komponentami programu tak, aby jedna nemusela mít na druhou referenci v době sestavování programu (Dependency injection, 2014).

aplikací mimo jiné prostřednictvím data-bindingu, dependency injection, snadným rozšiřováním syntaxe HTML (Google, 2014).

## HTML5

HTML je značkovací jazyk určený k tvorbě webových stránek. HTML5 přináší oproti svému předchůdci podstatné změny, mezi které patří zejména možnost přehrávání multimédií v prohlížeči a podpora pro aplikace, které fungují i bez připojení k Internetu (HTML5, 2014).

## CSS3

Kaskádové styly je pojmenování pro jazyk popisující formátování a vzhled dokumentů napsaných v některém ze značkovacích jazyků. Třetí verze kaskádových stylů CSS3 oproti svým předchůdcům přináší nové možnosti nebo ty stávající rozšiřuje s tím, že je zachována zpětná kompatibilita. Mezi nové vlastnosti patří například zaoblení rohů elementu, vržený stín, transformace a další (Cascading Style Sheets, 2014).

## Návrhový model

Model obsahuje architektonický popis architektury systému, komponent tvořících výsledný program, jejich vzájemných vztahů a zodpovědností a komunikaci s externími službami a systémy. Pro jednotlivé komponenty obsahuje popis návrhových tříd, přiřazení zodpovědností a popis spolupráce pro realizaci požadovaných funkcí. Pro grafickou reprezentaci statického modelu využívá UML diagram komponent, diagram nasazení a diagram tříd. Pro grafickou reprezentaci dynamického chování využívá UML sekvenční diagramy.

## Architektura systému

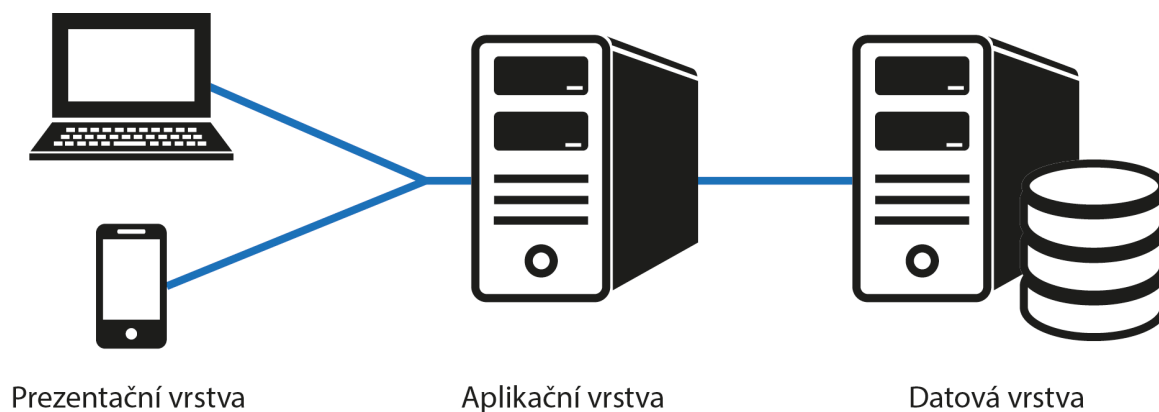
Diagram architektury systému (Obrázek 10 – Architektura systému) popisuje high-level přehled hlavních softwarových komponent systému SYBA SharedBase a zobrazení, jakým způsobem tyto celky spolupracují, přehled rozhraní a způsob interakce s externími systémy. Detailní popis jednotlivých komponent se nachází v sekci Návrh komponent.

Architektura řešení systému SYBA SharedBase je navržena podle principů třívrstvé architektury<sup>2</sup>. Tento typ architektury informačních systémů rozděluje systém na vrstvu prezentační (to, co uživatel vidí a používá) a vrstvy aplikační a datovou (to, co se odehrává na straně serveru). Výhodou vícevrstvé architektury je možnost rozdělení kompetencí jednotlivých částí a možnost jejich nezávislého vývoje, udržování i záměny za předpokladu, že rozhraní mezi nimi zůstává neměnné (Tonar, 15). Díky tomuto principu je možné, aby stejné rozhraní bylo používáno různými implementacemi pro odlišná zařízení a platformy. Rozhraní aplikační vrstvy může být používáno subsystémem SYBA.SharedBase.Frontend (Obrázek 10 – Architektura systému) – webové rozhraní na evidenci a správu kontaktů a zároveň i systémem SYBA.SharedBase.Mobile – mobilní aplikace pro správu kontaktů.

---

<sup>1</sup> Webová aplikace, jejíž veškerý kód – HTML, JavaScript, CSS – je klientovi poslán při prvním načtení stránky. V případě potřeby jsou dodatečné zdroje dynamicky načítány jako reakce na uživatelskou akci, nikdy však nedochází k reloadování stránky (Single-page application, 2014).

<sup>2</sup> Anglicky Three-tier architecture.



Obrázek 9 – Diagram třívrstvé architektury

Prezentáční vrstva (označena zeleným šrafováním na obrázku Obrázek 10 – Architektura systému) neboli vrstva uživatelského rozhraní je zodpovědná za interakci s uživatelem. Ten jejím prostřednictvím může zadávat informace, vyvolávat akce a jsou mu zobrazovány reakce systému. V systému SYBA SharedBase probíhá komunikace prezentační vrstvy s vrstvou aplikační prostřednictvím asynchronních požadavků na aplikační rozhraní s využitím protokolu HTTP.

Aplikační vrstva (označena červeným šrafováním na obrázku Obrázek 10 – Architektura systému) neboli aplikační server je prostřední vrstvou modelu (tzv. middleware) a zajišťuje přenos informací mezi prezentační vrstvou a datovou vrstvou, a to včetně provedení výpočtů a operací nad přenášenými daty nebo jejich transformaci. V systému SYBA SharedBase je aplikační vrstva od datové vrstvy oddělena prostřednictvím repository pattern (viz Vzor Repositář).

Datová vrstva (označena fialovým šrafováním na obrázku Obrázek 10 – Architektura systému) je nejnižší vrstvou modelu, zajišťuje práci s daty a implementuje persistentní úložiště dat s využitím SŘBD<sup>1</sup>. V systému SYBA SharedBase datová vrstva provádí datově-funkční operace nad relační SQL databází prostřednictvím ORM nástroje NHibernate.

## Návrhové vzory

V této sekci je uveden přehled hlavních návrhových vzorů využitých v návrhu systému SYBA SharedBase.

### Vzor Fasáda

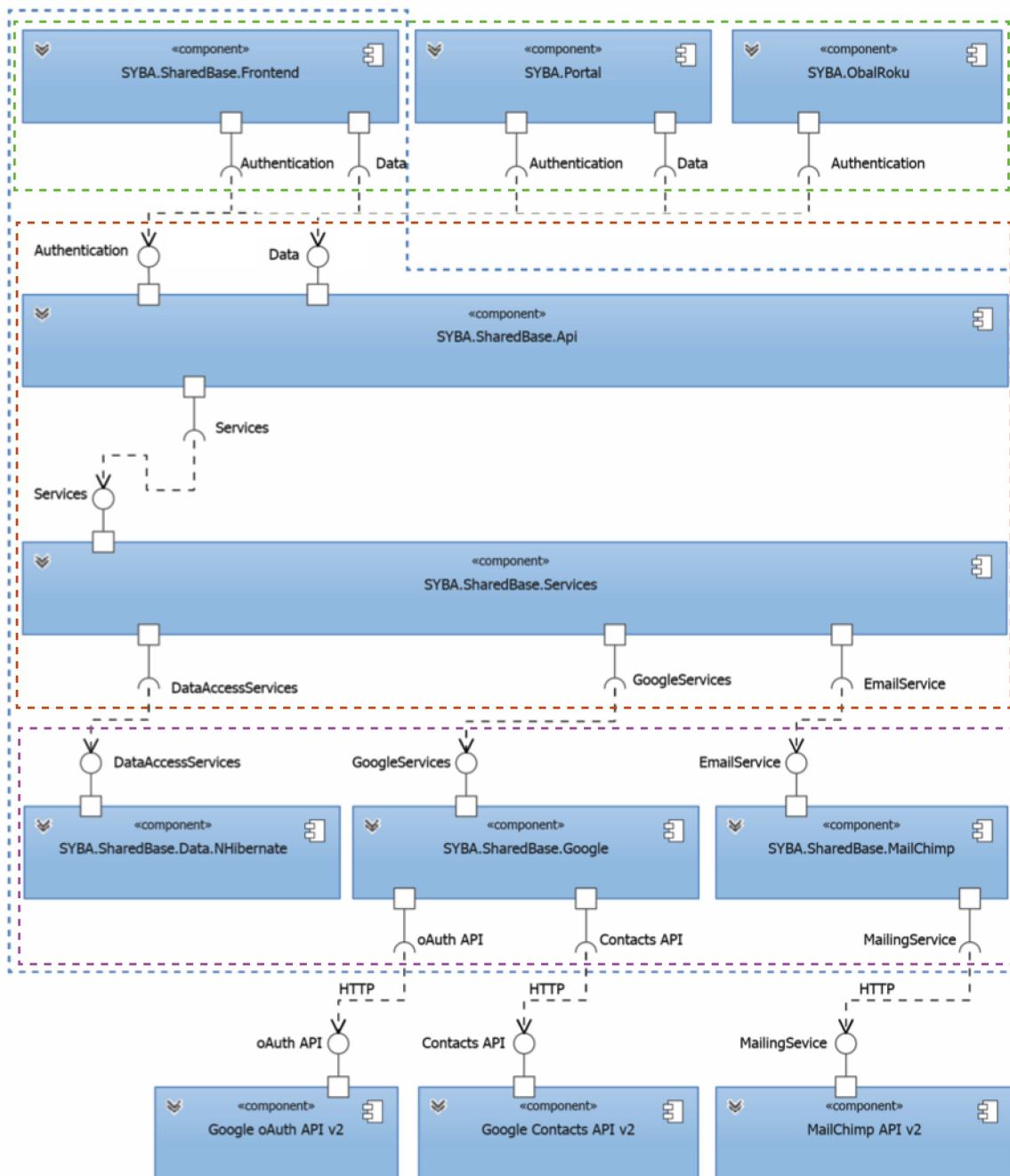
Návrhový vzor fasáda (anglicky Facade Pattern) je použit v aplikační vrstvě, kde servisní vrstvu tvoří množina fasád nad doménovým modelem a submoduly systému. Tyto fasády nabízejí jednotná rozhraní, množiny operací, pomocí nichž komunikují se subsystemy. Abstrahují tak tyto subsystemy od zbytku aplikace.

### Vzor Repositář

Návrhový vzor repositář (anglicky Repository Pattern) slouží k rozdělení zodpovědnosti za získání dat z databáze a jejich mapování na entity modelu od aplikační logiky, která pracuje s tímto modelem.

<sup>1</sup> Systém řízení báze dat.

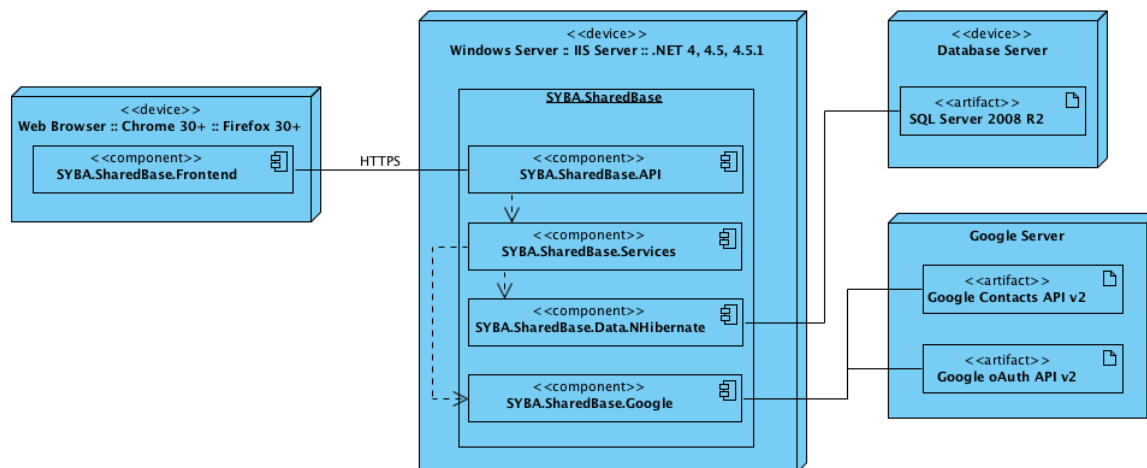




Obrázek 10 – Architektura systému

## Model nasazení

Diagram nasazení (Obrázek 11 – Diagram nasazení) specifikuje topologii technických prostředků určených ke spuštění a běhu systému SYBA SharedBase.



Obrázek 11 – Diagram nasazení

## Návrh komponent

Tato sekce se blíže zabývá jednotlivými komponentami systému SYBA SharedBase. U každé komponenty je uveden její popis, zodpovědnosti a návrhové diagramy dokumentující balíčky, třídy a metody. U každé třídy je uveden její krátký popis a přehled seznam funkcí.

### SYBA SharedBase Domain

Komponenta SYBA SharedBase Domain obsahuje definice POCO entit odpovídající datovému modelu a definuje rozhraní pro operace s nimi.

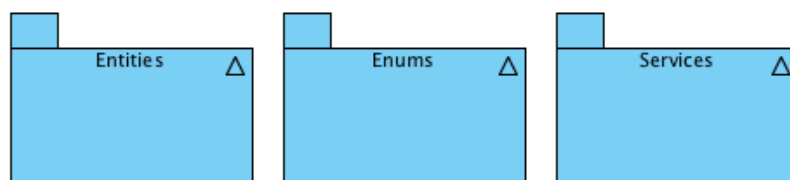
#### Hlavní vlastnosti a funkce

1. Definice datového modelu systému SYBA SharedBase.
2. Definice rozhraní pro práci s datovým modelem.

#### Návrh komponenty

Diagram na obrázku Obrázek 12 – SYBA SharedBase Domain zobrazuje rozdělení komponenty do tří balíčků:

1. Entities: Definice business entit v systému.
2. Enums: Definice výčtových typů používaných doménovým modelem.
3. Services: Definice rozhraní tříd pro operace s doménovým modelem.



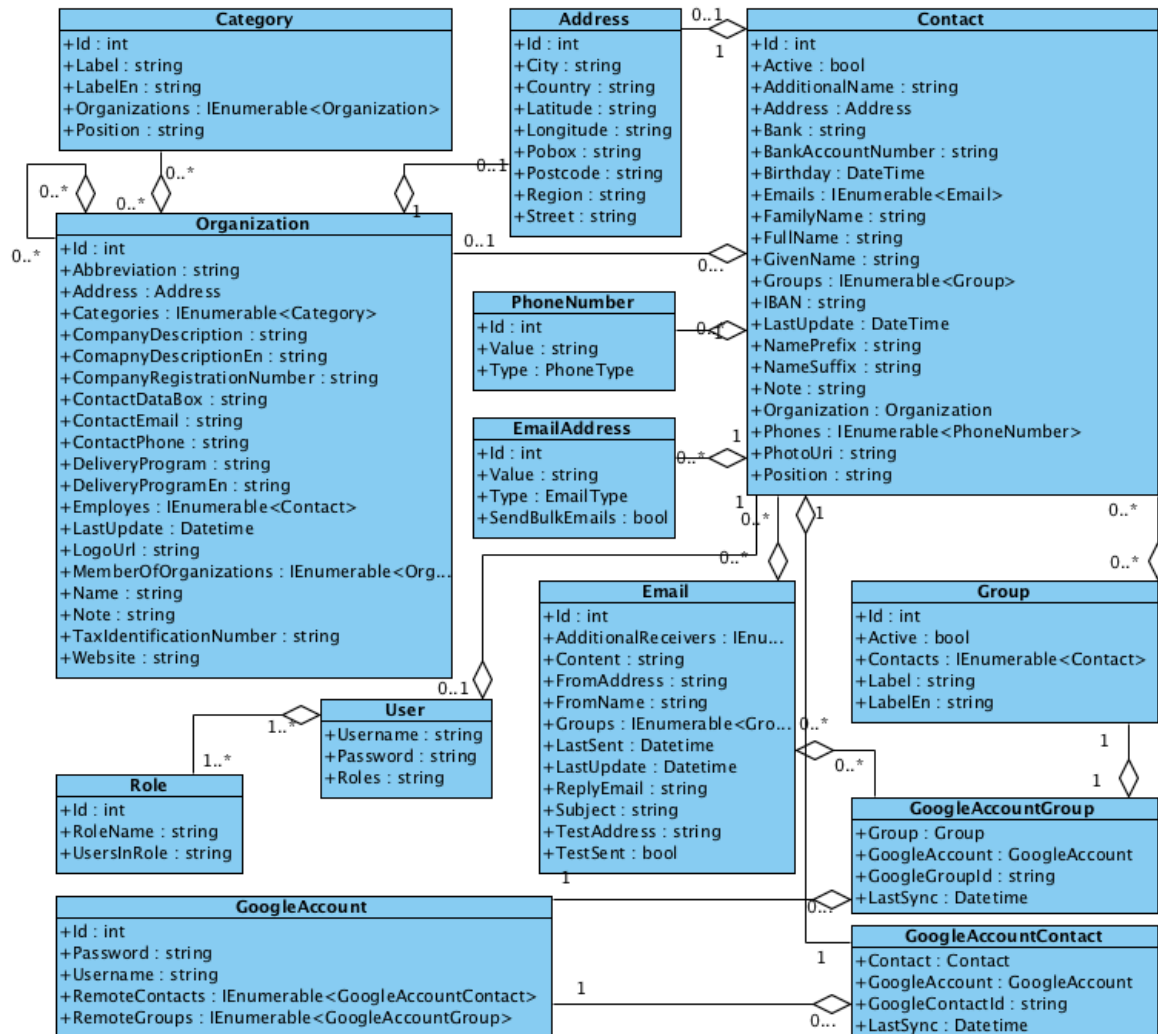
Obrázek 12 – SYBA SharedBase Domain: Přehled balíčků

#### Balíček Entities

Balíček Entities obsahuje business entity nebo obecněji business objekty, obaluje business logiku a data nezbytná k reprezentaci objektů ze skutečného světa – např. Kontakt, Společnost – uvnitř

systému. Business entity také validují data obsažená v entitách a zapouzdrují logiku zaručující konzistenci dat.

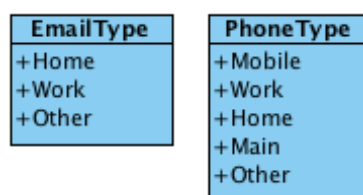
Atributy ve entitách Contact a Group odpovídají pojmenování atributů ve třídách Contact a Group ve službě Google Contacts.



Obrázek 13 – SYBA SharedBase Domain: Balíček Entities

### Balíček Enums

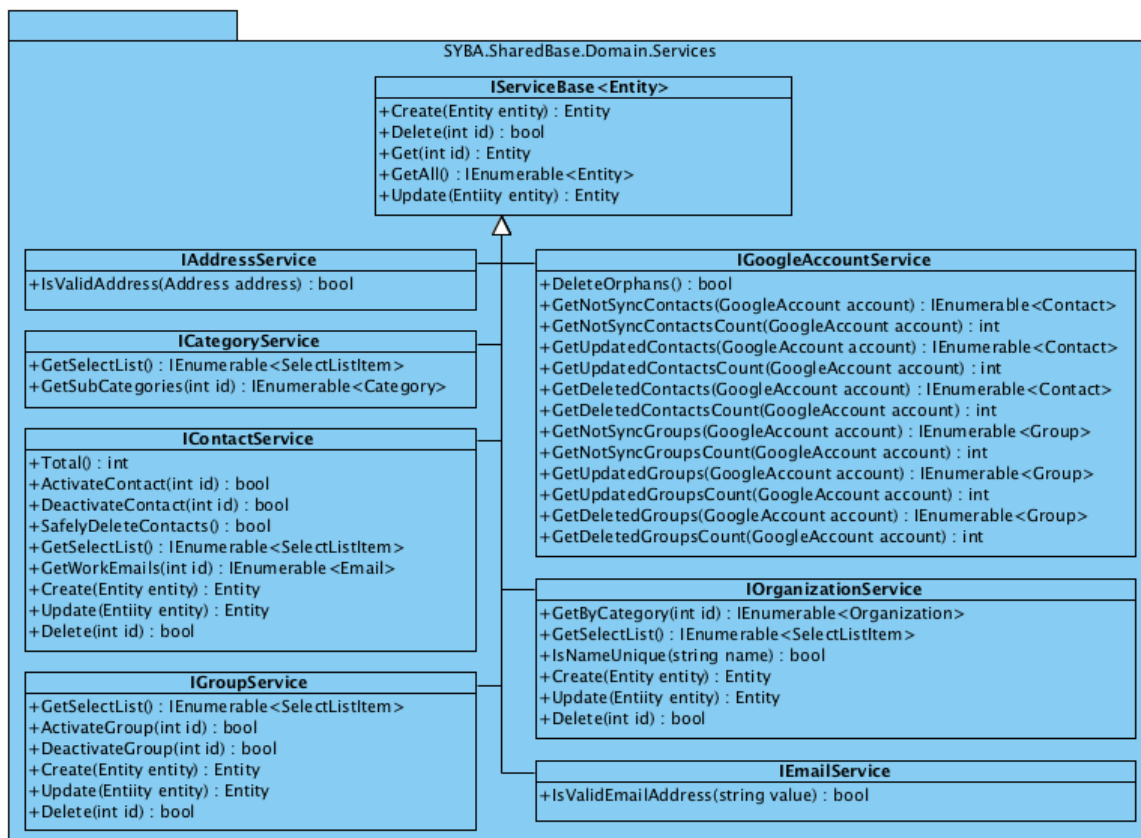
Balíček Enums obsahuje výčtové typy s konstantami pro určení typu telefonního čísla a e-mailové adresy. Hodnoty v těchto výčtových typech jsou podmnožinou hodnot, jichž mohou nabývat typy telefonních čísel a e-mailových adres v systému Google Contacts.



Obrázek 14 – SYBA SharedBase Domain: Balíček Enums

## Balíček Services

Balíček Services obsahuje definice rozhraní pro business logiku nad entitami v systému.



## SYBA SharedBase Data NHibernate

Komponenta SYBA SharedBase Data NHibernate představuje implementaci datové vrstvy systému. Obsahuje třídy, které centralizují veškerý přístup k datům uloženým v databázi, všechny CRUD operace – operace vytvoření, čtení, modifikace a smazání – jsou prováděny pouze prostřednictvím této komponenty. Pro samotný přístup k datům vrstva využívá ORM framework Fluent NHibernate.

### Hlavní vlastnosti a funkce

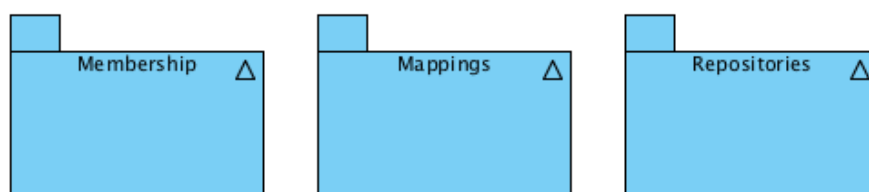
1. Prostřednictvím ORM Fluent NHibernate mapuje entity definované v SYBA SharedBase Domain na tabulky v relační SQL databázi.
2. Implementuje rozhraní typu IRepository pro CRUD operace nad daty v databázi. Ostatní komponenty systému provádí operace s daty v databázi pouze prostřednictvím těchto repository služeb.
3. Poskytuje implementaci třídy MembershipProvider a RoleProvider s využitím NHibernate. Jedná se o služby pro správu uživatelů a jejich rolí v systému, jejich autentizaci a autorizaci.

### Návrh komponenty

Diagram na obrázku Obrázek 15 – SYBA SharedBase Data NHibernate zobrazuje rozdělení komponenty do tří balíčků:

1. Mappings: Fluent NHibernate mapovací třídy.
2. Repositories: Implementace IRepository rozhraní.

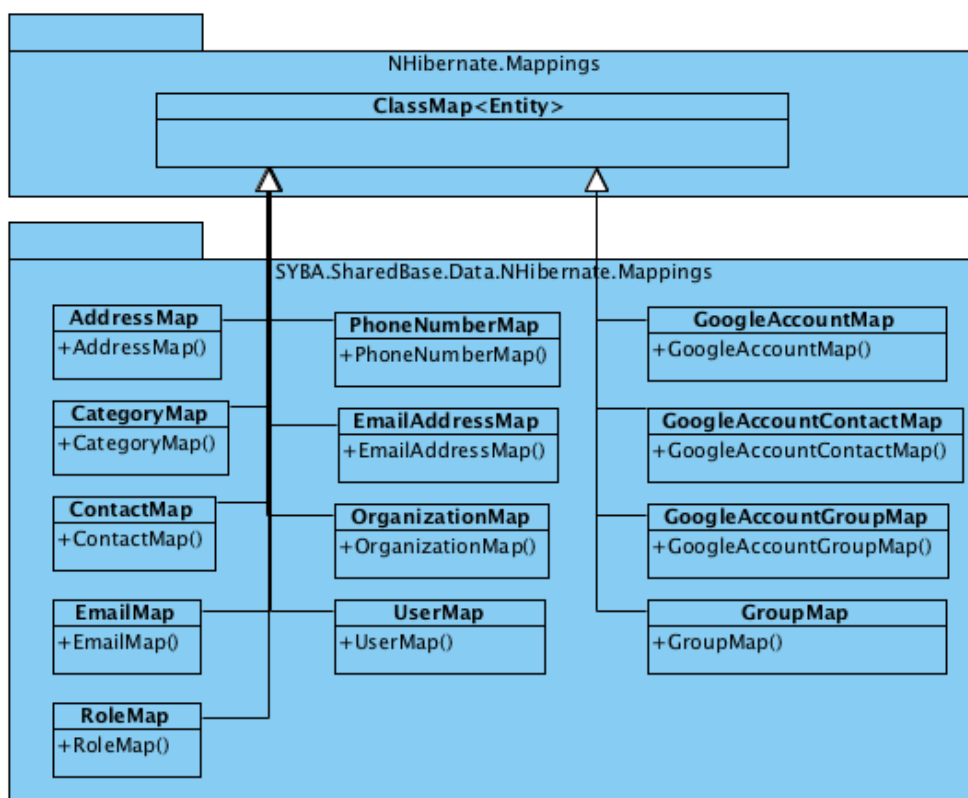
3. Membership: Vlastní implementace autentizačních a autorizačních ASP.NET služeb.



Obrázek 15 – SYBA SharedBase Data NHibernate: Přehled balíčků

### Balíček Mappings

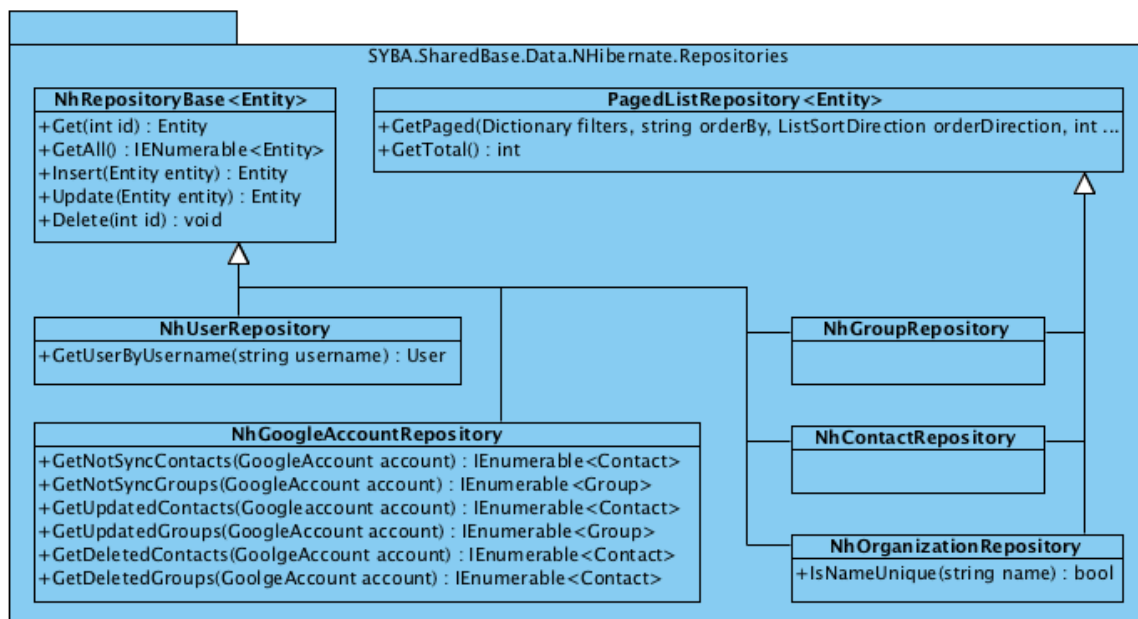
Balíček obsahuje třídy mapující všechny entity doménového modelu na tabulky v relační databázi. Mapování je uskutečněno prostřednictvím mapovacích tříd nástroje Fluent NHibernate.



Obrázek 16 – SYBA SharedBase Data NHiberante: Balíček Mappings

### Balíček Repositories

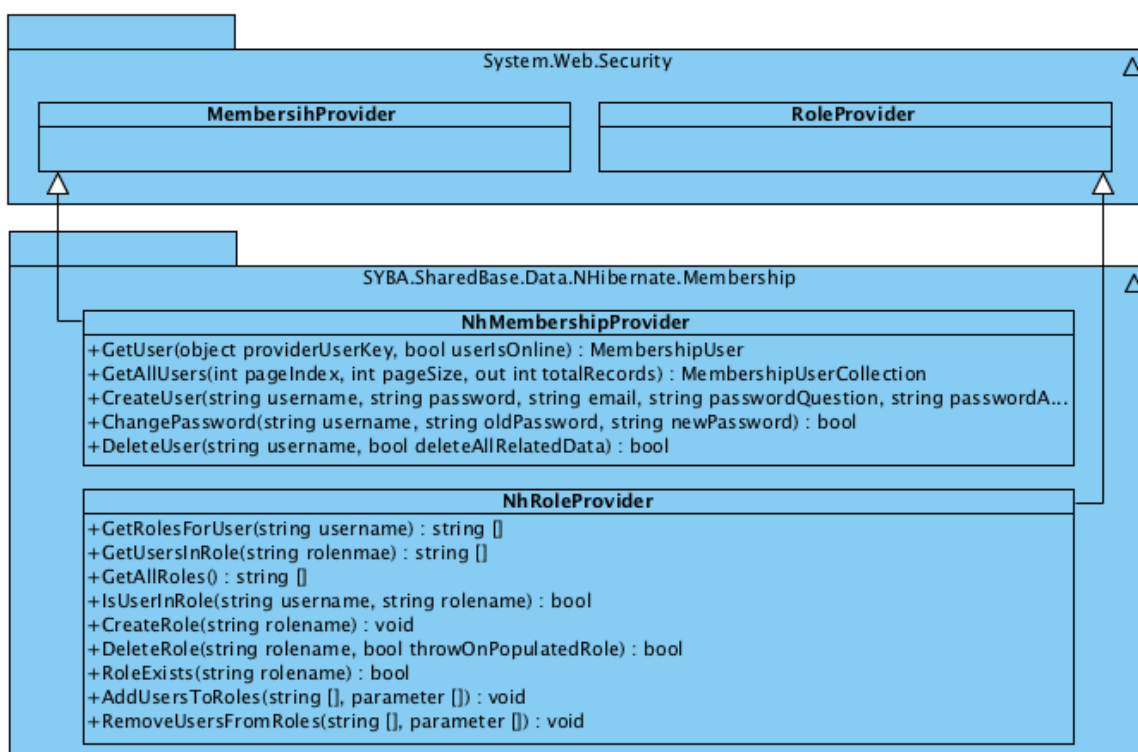
Třídy v balíčku Repositories implementují rozhraní typu IRepository pro přístup k datům. Slouží pro přímé CRUD operace s daty v databázi prostřednictvím nástroje NHibernate.



Obrázek 17 – SYBA SharedBase Data NHibernate: Balíček Repositories

### Balíček Membership

Vlastní implementace autentizačních a autorizačních ASP.NET služeb. Rozšiřuje autentizační služby vestavěné v ASP.NET o operace prováděné nad databází systému SYBA SharedBase s využitím NHibernate.



Obrázek 18 – SYBA SharedBase Data NHibernate: Balíček Membership

## SYBA SharedBase Services

Komponenta SYBA SharedBase Services obsahuje aplikační logiku systému. Definuje služby, které jsou odpovědné za vyhledání, zpracování, transformaci a správu dat v systému.

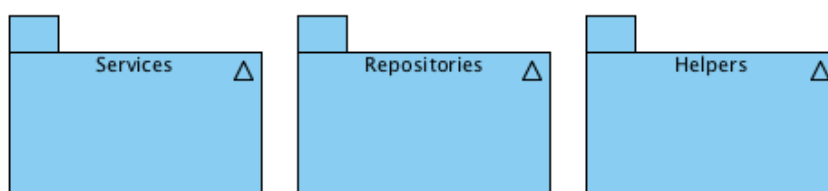
### Hlavní vlastnosti a funkce

1. Aplikační logika na vyhledání, zpracování, transformaci a správu dat v systému, provádění a koordinace dlouhých, více krokových business procesů.

### Návrh komponenty

Diagram na obrázku Obrázek 19 – SYBA SharedBase Services zobrazuje rozdělení komponenty do tří balíčků:

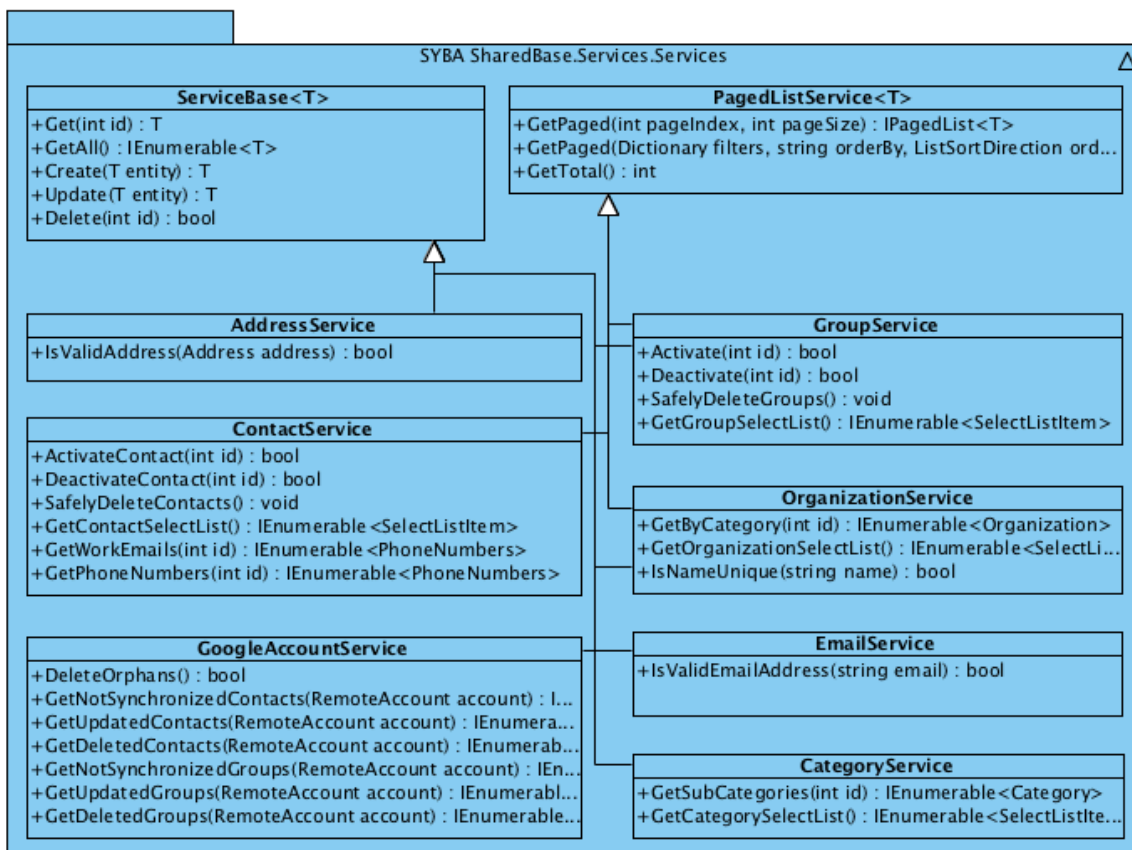
1. Services: Implementace servisních tříd obsahujících aplikační logiku.
2. Repositories: Definice rozhraní pro CRUD operace nad datovým modelem.
3. Helpers: Pomocné třídy.



Obrázek 19 – SYBA SharedBase Services: Přehled balíčků

### Balíček Services

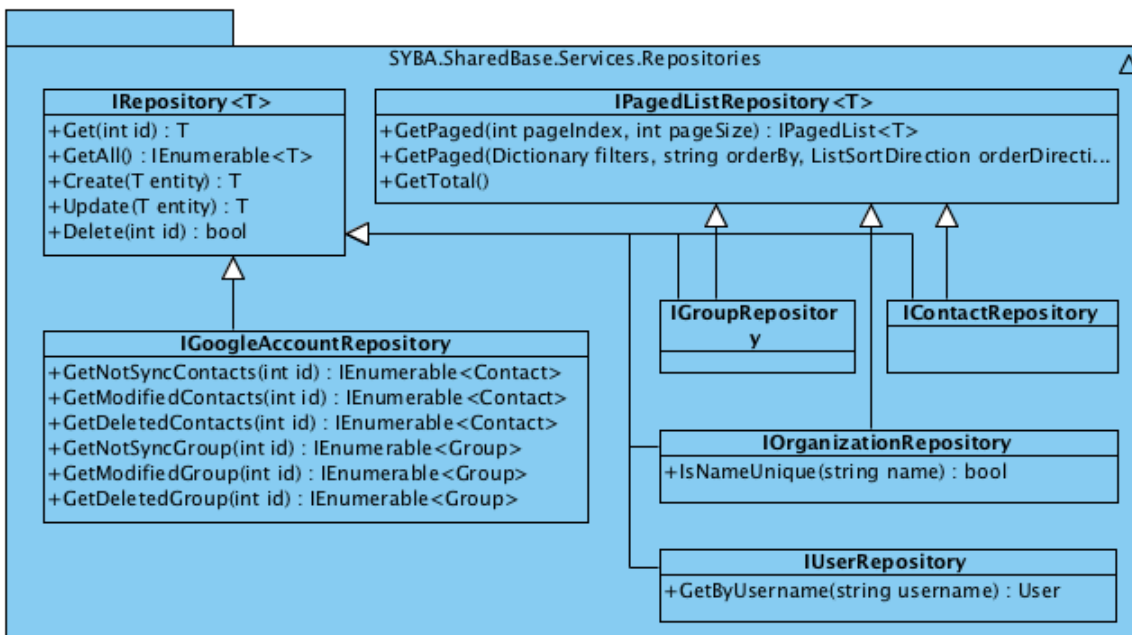
Balíček Services definuje aplikační logiku pro business procesy nad jednotlivými entitami.



Obrázek 20 - SYBA SharedBase Services: Balíček Services

**Balíček Repositories**

Balíček repositories definuje rozhraní pro operace v databázi.

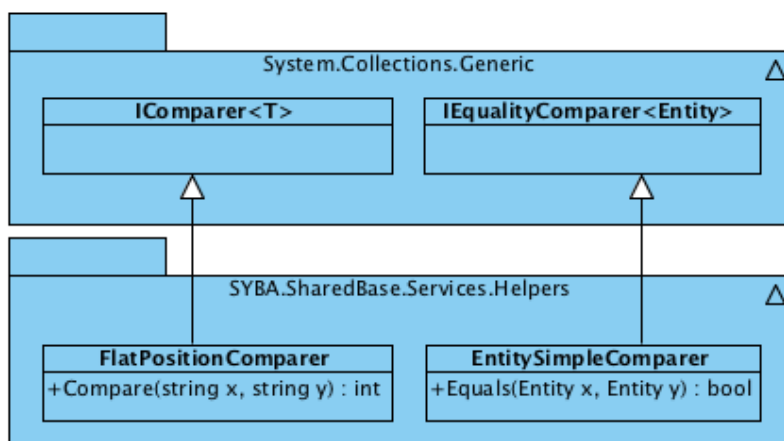


Obrázek 21 – SYBA SharedBase Services: Balíček Repositories



### Balíček Helpers

Balíček Helpers obsahuje definice pomocných tříd, jmenovitě třídy na porovnávání dvou entit z doménového modelu a porovnávání pozic jednotlivých objektů ve stromu kategorií společností.



Obrázek 22 – SYBA SharedBase Services: Balíček Helpers

### SYBA SharedBase Google

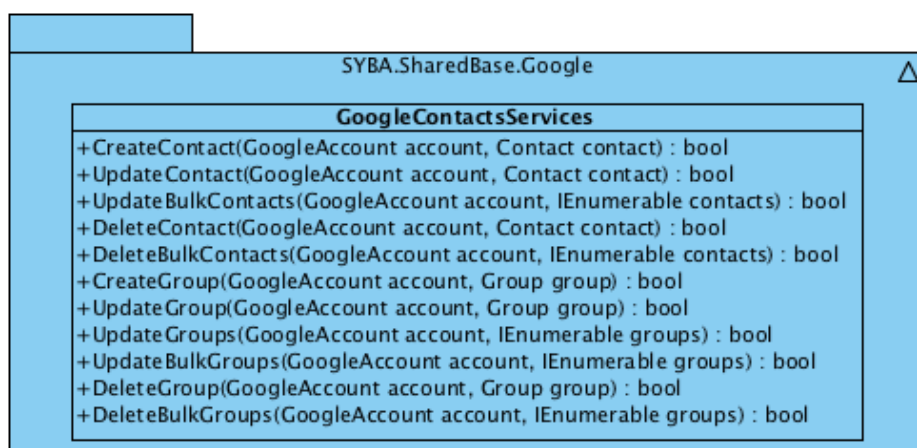
Komponenta SYBA SharedBase Google implementuje prostřednictvím Google GData komunikaci s Google Contact API.

#### Hlavní vlastnosti a funkce

1. Synchronizace dat s účtem ve službě Google prostřednictvím aplikačního rozhraní Google Contacts API v2.

#### Návrh komponenty

Komponenta SYBA SharedBase Google obsahuje pouze jednu třídu – `GoogleContactsService`. Tato služba prostřednictvím knihovny Google GData ke komunikaci s aplikačním rozhraním Google Contact API v2, synchronizuje změny v systému s účtem ve službě Google a zaznamenává stav synchronizace jednotlivých položek v systému SYBA SharedBase.



Obrázek 23 –SYBA SharedBase Google: Přehled komponenty

## SYBA SharedBase API

Komponenta SYBA SharedBase API implementuje aplikační rozhraní pro autentizaci a správu dat v systému SYBA SharedBase prostřednictvím protokolu HTTP.

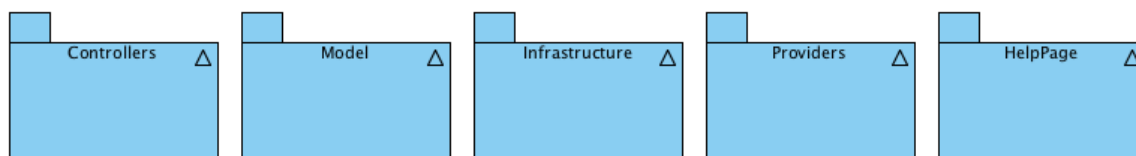
### Hlavní vlastnosti a funkce

1. Definuje koncové body aplikačního rozhraní a metody protokolu HTTP.
2. Definuje DTO<sup>1</sup> modely přenášených dat a poskytuje nástroje na překlad dat mezi business entitami a DTO objekty.
3. Automaticky z kódu generuje stránku specifikace API.

### Návrh komponenty

Diagram na obrázku Obrázek 24 – SYBA SharedBase API zobrazuje rozdělení komponenty do následujících balíčků:

1. Controllers: Definice přístupových bodů API.
2. Models: Definice DTO objektů.
3. Infrastructure: Dependency Injection.
4. Providers: Autentizace prostřednictvím OAuth protokolu.
5. HelpPage: ASP.NET Area obsahující automaticky generovanou specifikaci API.



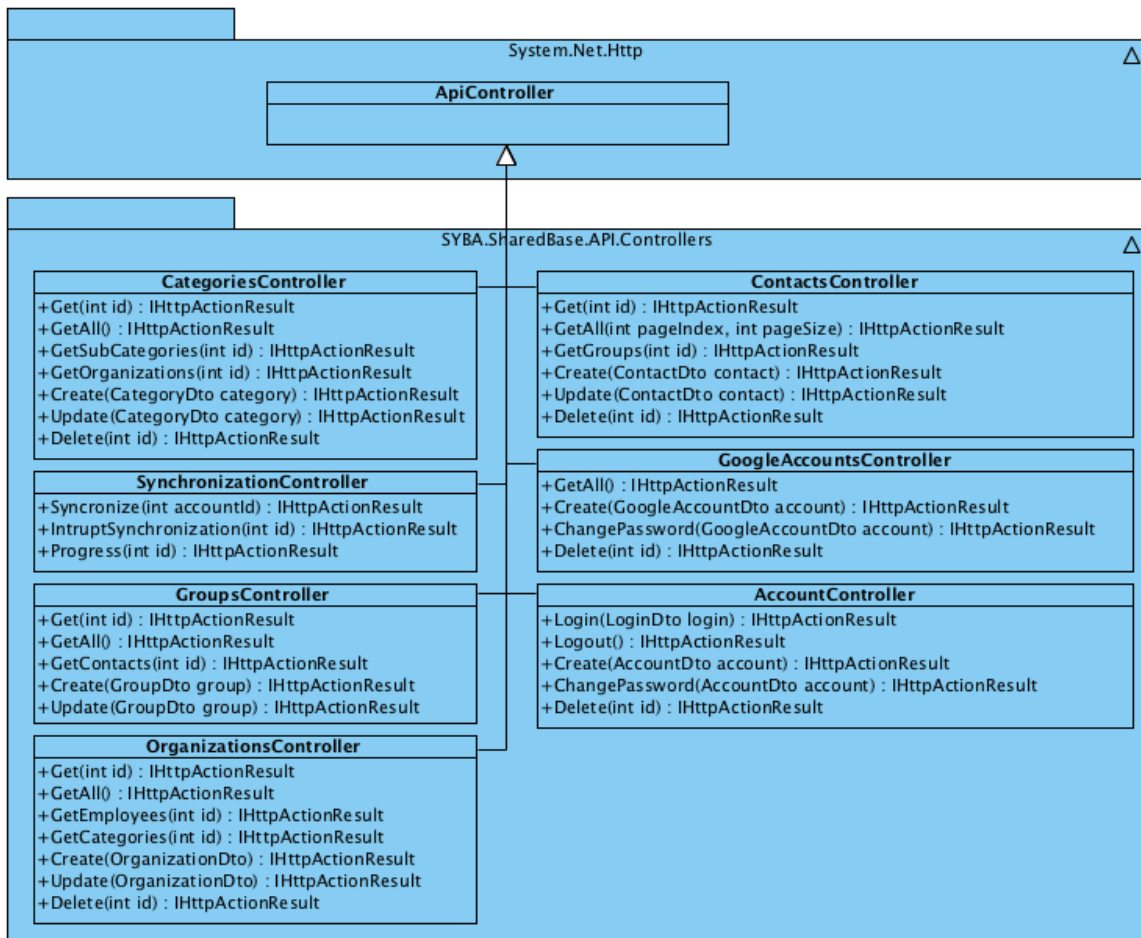
Obrázek 24 – SYBA SharedBase API: Přehled balíčků

#### Balíček Controllers

Balíček Controllers obsahuje implementace tříd typu ApiController. Jedná se o specializované třídy Controllers, které vrací serializovaná data podle formátu požadovaného klientem.

---

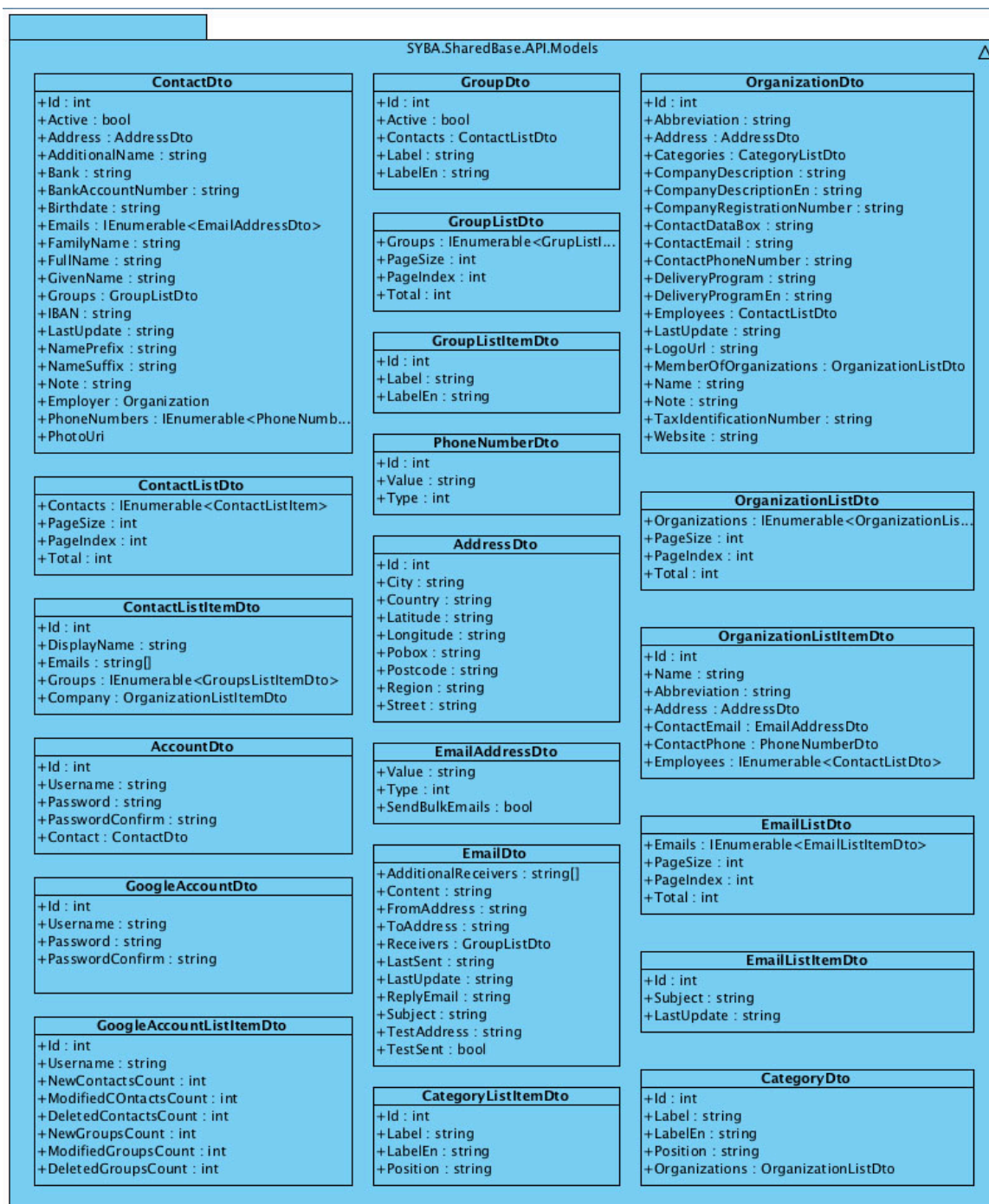
<sup>1</sup> Data Transfer Object.



Obrázek 25 - SYBA SharedBase API: Balíček Controllers

### Balíček Models

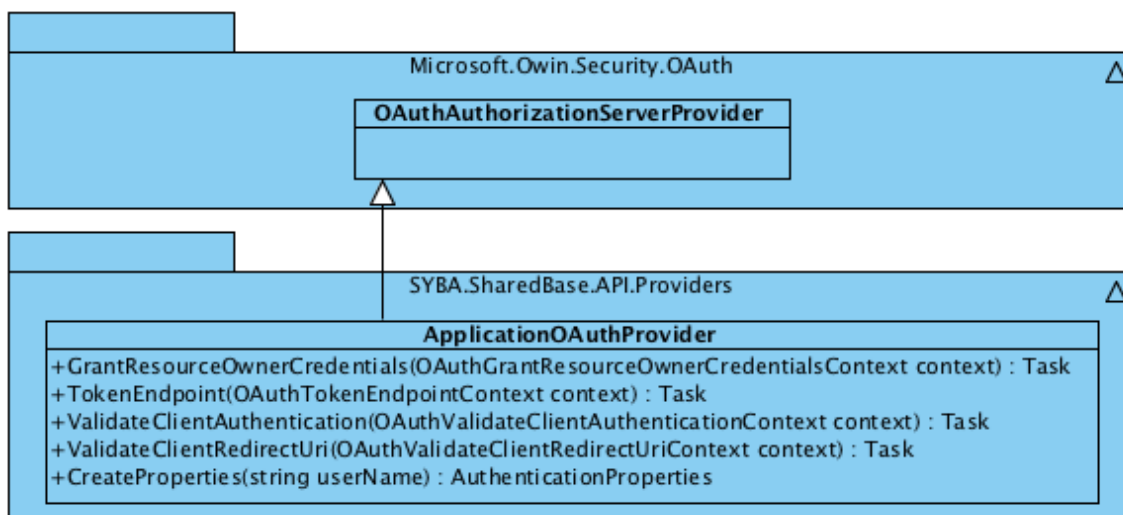
Balíček Models obsahuje definice DTO objektů dotazů a odpovědí API. Objekty jsou anotovány, aby byla umožněna jejich automatická serializace do formátu JSON/XML pro přenos prostřednictvím protokolu HTTP.



Obrázek 26 – SYBA SharedBase API: Balíček Models

**Balíček Providers**

Balíček Providers obsahuje vlastní implementaci autentizační služby prostřednictvím OAuth protokolu.



Obrázek 27 – SYBA SharedBase API: Balíček Providers

### Balíček HelpPage

ASP.NET Web API poskytuje knihovnu pro automatické generování specifikace API za běhu programu. Jedná se o jednoduchou MVC aplikaci, která generuje tabulku všech koncových bodů seskupených podle API řadičů. Každý záznam obsahuje odkaz na stránku s detailním popisem včetně ukázkového dotazu na API a následné odpovědi.

## SYBA SharedBase Frontend

Komponenta SYBA SharedBase Frontend obsahuje uživatelské rozhraní správy dat v systému SYBA SharedBase. Komunikace se systémem prostřednictvím API definovaném komponentou SYBA SharedBase Api. Její architektura je daná použitým MVC frameworkem AngularJS. Návrh uživatelských rozhraní systému ve formě náčrtů nízké úrovně je přiložen v dodatku (Příloha B: Návrh uživatelského rozhraní).

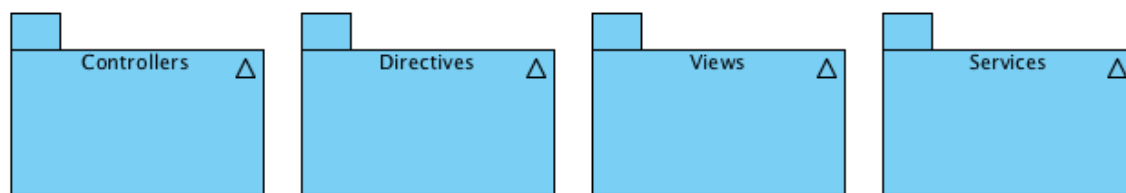
### Hlavní vlastnosti a funkce

1. Uživatelské rozhraní pro přihlášení uživatele do systému a jeho odhlášení. Udržování User Session v paměti prohlížeče pro „nekonečné přihlášení“.
2. Uživatelské rozhraní pro zobrazování, vytváření, úpravu a mazání kontaktů, skupin kontaktů, společností a jejich kategorií.
3. Uživatelské rozhraní pro synchronizaci kontaktů s účtem ve službě Google, zobrazování průběhu synchronizace.

### Návrh komponenty

Diagram na obrázku Obrázek 28 - SYBA SharedBase Frontend zobrazuje rozdělení komponenty do následujících balíčků:

1. Controllers: AngularJS Controllers.
2. Directives: Vlastní rozšíření HTML elementů prostřednictvím AngularJS Directives.
3. Services: AngularJS Factories a Providers.
4. Views: Šablony, části HTML kódu vkládané do SPA.



Obrázek 28 - SYBA SharedBase Frontend: Přehled balíčků



# Testování

Tato kapitola obsahuje testovací dokumentaci systému SYBA SharedBase verze 1.0 analyzovaného a navrženého v předešlých kapitolách (Analýza, Návrh). V dokumentu je navržena metodika testování odpovídající rozsahu systému a požadavkům zadavatele.

Dokument obsahuje celkový plán testování systému, definuje, jakým způsobem bude přistoupeno k testování daného systému, a určuje rozsah, zdroje a časový přehled testovacích aktivit. Zároveň jsou zde identifikovány části a funkce systému podstupující testování, úlohy související s daným procesem testování a rizika spojená se zde definovaným plánem.

## Úlohy a cíle

Při návrhu metodiky testování systému SYBA SharedBase je potřeba brát v úvahu, že testování je náročný proces, který zabírá nemalou část časových a finančních prostředků a na jehož konci si přesto nemůžeme být jisti, že je daný systém otestován dostatečně. Zároveň vedle časových a finančních omezení existují i omezení plynoucí z požadavků definovaných ve specifikaci požadavků (Specifikace požadavků). V neposlední řadě faktorem ovlivňujícím návrh testování jednotlivých částí systému je kritičnost neobjevené chyby v dané části systému ve fázi testování a cena chyby objevené až v reálném provozu.

Cílem této metodiky je ověřit všechny funkčnosti systému a ověření fungování systému na zadavatelem požadované produkční platformě.

## Řízení chyb

Zdrojový kód systému je udržován v GIT repositáři ve službě GitHub (požadavek omezení ON-6 v Omezení návrhu a implementace). Tato služba také nabízí vlastní issue tracking systém. Tento systém bude sloužit k zaznamenávání nalezených chyb a nastavení jejich priority a milníků, ve kterých je požadována náprava, k přiřazení osob zodpovědných za jejich řešení a k záznamu celého workflow tohoto řešení. GitHub issue tracking systém bude sloužit nejen k zaznamenání chyb nalezených ve fázi testování, ale i chyb, které se objeví až za reálného provozu systému.

## Spouštění testů

Bylo by možné použít server s kontinuální integrací (například TFS<sup>1</sup> s velice slušnou integrací do Visual Studia), kdy po každé ucelené změně kódů systému by byla nová verze nahrána na integrační server, kde se automaticky spustí definované testy. V tuto chvíli si však zadavatel přeje, aby zdrojové kódy byly v repositáři ve službě GitHub (výše zmíněný požadavek ON-6), který automatické sestavení aplikace a spouštění testů nepodporuje. Z tohoto důvodu se u jednotlivých typů testování zaměříme na jiné způsoby jejich automatizace.

---

<sup>1</sup> Team Foundation Server.



## Regresní testování

Po opravě objevené chyby v systému nebo po přidání nové funkcionality se spouštějí regresní testy. Tyto testy mají za úkol ověřit, zda-li tato změna v systému nenarušila funkci jiných částí systému, které tímto zásahem neměly být ovlivněny. Regresní testování se provádí automatickým spouštěním všech jednotkových, integračních a systémových testů po sestavení systému s provedenou změnou.

## Strategie testování

Tato část popisuje celkový přístup k testování systému. Jsou zde definovány úrovně testů aplikované při testování SYBA SharedBase. Pro jednotlivé architektonické vrstvy a hlavní komponenty systému je zde specifikován přístup k jejich testování, jsou určeny činnosti, techniky a používané nástroje. Strategie testování jednotlivých částí je popsána natolik podrobně, aby umožnila identifikaci hlavních testovacích úloh a odhad času potřebného k jejich provedení.

## Rozsah

Menší až středně velký rozsah systému SYBA SharedBase a omezené časové a finanční prostředky určené na fázi testování mají za následek omezení aplikovaných úrovní testů systému na jednotkové, systémové, integrační a akceptační testy.

## Jednotkové testování

Jednotkové testování slouží k ověření funkčnosti samostatných komponent. Zaobírá se ověřením systému na úrovni jednotlivých tříd a metod.

Měřítkem stupně otestování systému bude úroveň pokrytí kódu jednotkovými testy. Vyšší pokrytí kódu testy znamená důkladnější otestování systému a snižuje pravděpodobnost přehlédnuté chyby v kódu. Cílem je co nejvyšší, nejlépe stoprocentní pokrytí.

## Testování datové vrstvy

Datovou vrstvu systému reprezentuje komponenta SYBA SharedBase Data NHibernate. Ta pro mapování objektů na tabulky v databázi používá ORM nástroj NHibernate, jmenovitě jeho rozšíření Fluent NHibernate (viz SYBA SharedBase Data NHibernate). Testy pro tuto komponentu budou implementovány v projektu SYBA SharedBase Data NHibernate Test.

## Rozsah

Při použití objektově-relačního mapování NHibernate je potřeba otestovat tři základní věci:

1. zda jsou jednotlivé vlastnosti objektů správně uloženy do databáze,
2. zda jsou kaskádové procesy správně propagovány,
3. zda dotazy do databáze vrací správné výsledky.

## Nástroje

Framework Fluent NHibernate sám o sobě obsahuje knihovnu pro psaní testů ověřujících mapování vlastností objektů do relační databáze. Je tak například možné jednoduše vytvořit

novou instanci Kontaktu, vložit Kontakt do databáze, načíst záznam z databáze do nové instance Kontaktu a ověřit, že načtený Kontakt odpovídá originálu.

Aby bylo možné otestovat propagaci kaskádových procesů a ověřit správnost výsledků dotazů do databáze, je obvykle nezbytné komunikovat s reálnou databází. Snažit se na této úrovni nějakým způsobem databázi imitovat, je dosti komplikované. Komunikace s klasickou databází může být při velkém množství testů dost pomalá, proto je možné pro dosažení rychlého testování použít databázi v paměti počítače, jako je například databáze SQLite. Právě pomocí NHibernate lze poměrně snadno vytvořit a přistupovat k SQLite databázi v paměti a tak i provádět jednotkové testy dostatečně rychle v paměti počítače.

Samotné jednotkové testy budou psány s využitím frameworku NUnit. NUnit je framework pro jednotkové testování všech jazyků na platformě .NET.

### **Automatizace testů**

Přestože pro sdílení kódu nepoužíváme server s kontinuální integrací, ve Visual studiu lze nastavit, aby byly jednotkové testy spouštěny po každém sestavení aplikace. V případě, že je v systému provedena změna, systém je znovu sestaven a po každém takovém sestavení je ověřeno, zda-li provedená změna do systému nezavlekla chybu. Nutno poznamenat, že spouštění jednotkových testů po každém sestavení aplikace je funkce dostupná pouze ve Visual Studio Ultimate a Premium Edition.

### **Testování servisní vrstvy**

Datovou vrstvu systému reprezentuje komponenta SYBA SharedBase Services. Ta komunikuje s datovou vrstvou a s komponentami zodpovědnými za styk s externími systémy.

#### **Rozsah**

Cílem testování servisní vrstvy na úrovni jednotkových testů je ověření všech metod v servisních třídách balíčku Services a ověření správné funkčnosti pomocných tříd v balíčku Helpers.

Příklady:

1. Funkce GetContact(42) vrátí záznam s odpovídajícím identifikátorem.
2. Pro neexistující záznam s daným identifikátorem vrátí funkce GetContact(24) objekt null.
3. FlatPositionComparer správně porovná pozice „2.1“ a 21“.

#### **Nástroje**

Stejně jako jednotkové testy pro ověření datové vrstvy tak i tyto testy budou psány s využitím testovacího frameworku NUnit.

Pro oddělení testovaného kódu od zbytku aplikace (od repositářů datové vrstvy atp.) je nutné simulovat chování daným kódem využívaných tříd a komponent. K tomuto účelu slouží takzvané zástupné objekty (anglicky mock objects). Při testování systému SYBA SharedBase bude využíván Moq, framework pro vytváření zástupných objektů pracující pod AAA syntaxí<sup>1</sup>, rekursivní mockování, vytváření mockovacích repositářů atp.

---

<sup>1</sup> Arrange, Act, Assert.

## Testování aplikačního rozhraní

Komponenta SYBA SharedBase API definuje aplikační rozhraní pro správu dat v systému a autentizační služby (viz SYBA SharedBase API). Aplikační rozhraní je realizováno využitím ASP.NET Web API frameworku. Jednotlivé přístupové body rozhraní jsou definovány jako metody ve třídách typu ApiController. Právě tyto metody a třídy mohou být testovány pomocí klasických jednotkových testů.

### Rozsah

Cílem testování aplikačního rozhraní na úrovni jednotkových testů je ověření všech metod ve třídách typu ApiController a ověření správné funkčnosti metod ve službách v balíčku Providers.

Příklady:

1. Funkce GetContact(42) vrátí odpověď OkNegotiatedContentResult s obsahem nastaveným na záznam kontaktu s odpovídajícím identifikátorem.
2. Funkce GetContact(24) vrátí pro neexistující záznam s daným identifikátorem odpověď NotFoundResult.

### Nástroje

Nástroje potřebné k jednotkovému testování této vrstvy jsou stejné jako nástroje používané pro jednotkové testování servisní vrstvy.

## Testování uživatelského rozhraní

Uživatelské rozhraní neboli front-end systému SYBA SharedBase je navrženo v HTML a JavaScriptu a používá javascriptový MVC framework AngularJS. Stejně jako kód na back-endu, tak i základní jednotky na front-endu, které vytvářejí funkce systému, by měly být ověřeny unit testy. V javascriptových aplikacích jsou těmito základními jednotkami obvykle funkce – functions.

### Rozsah

Cílem je otestovat veškeré javascriptové funkce ve všech komponentách AngularJS – controllers, services, directives.

Příklady:

1. Funkce, která vyvolá zobrazení chybové hlášky.
2. Funkce validující správný formát e-mailové adresy.
3. Direktiva vykreslující HTML tag obsahující informace o kontaktu v seznamu kontaktů.

### Nástroje

Pro AngularJS existují frameworky a nástroje na testování, které umožňují definovat testy, automatizovat jejich spouštění a vyhodnocovat jejich průběh. Pro testování uživatelského rozhraní systému SYBA SharedBase jsou zvoleny následující nástroje:

#### *Jasmine*

Jasmine je testovací framework na definici a vyhodnocování kódu napsaného ve skriptovacím jazyce JavaScript. Jedná se o framework s jednoduchou syntaxí, který není závislý na žádném jiném javascriptovém frameworku.

**Karma**

Na samotné provádění nadefinovaných testů je potřeba takzvaný test runner, aplikace, která spouští webové prohlížeče a provádí oproti nim jednotlivé testy. Karma je test runner pro snadné spouštění a správu javascriptových testů, umožňující automatizované testování napříč webovými prohlížeči a připojenými zařízeními. Pro Karmu existuje množství zásuvných modulů, které rozšiřují jeho možnosti.

Karma umožňuje prostřednictvím zásuvných modulů spouštějících jednotlivé webové prohlížeče provádět simultánně testy proti více prohlížečům zároveň. V registru NPM jsou připraveny zásuvné moduly pro všechny hlavní webové prohlížeče. Zároveň umožňuje spouštět testy proti prohlížečům v zařízeních připojených k počítači, například v připojeném telefonu nebo tabletu.

Dále Karma umožňuje nejen spouštět jednorázové testy, ale funguje i v režimu sledujícím změny v kódu vyvíjeného systému a v případě, že programátor provede změnu v některém ze sledovaných javascriptových souborů, Karma tuto událost zachytí a automaticky znovu spustí jednotkové testy ve všech prohlížečích a připojených zařízeních.

**\$httpBackend**

Pro mockování HTTP dotazů na aplikační rozhraní na back-endové části systému je možné využít službu \$httpBackend. Při psaní jednotkových testů na uživatelské rozhraní tak nejsou tyto testy závislé na reálném back-endu aplikace.

**Integrační a systémové testování**

Na rozdíl od jednotkových testů se integrační testování neorientuje na testování malých částí kódu, ale na testování, jak integrované komponenty pracují dohromady. Integrované komponenty mohou být jak vnitřní (jednotlivé komponenty systému), tak i vnější (integrace s vnějšími systémy a službami). Systémové testování je v podstatě integrační testování, kdy testujeme, že všechny komponenty systému fungují jako celek.

**Testování aplikačního rozhraní**

Aby bylo zaručeno správné fungování aplikačního rozhraní a komunikace existujících klientů například s novou verzí tohoto rozhraní, je vhodné mít připraveny automatické integrační testy, které novou verzi rozhraní ověří.

**Rozsah**

Cílem je ověřit, že skutečné dotazy na aplikační rozhraní na straně serveru vrací správné odpovědi.

Příklady:

1. HTTP Get dotaz na adrese /api/contacts vrátí odpověď s kódem 200 Ok a seznamem kontaktů.
2. HTTP Get dotaz na adresu /api/contacts/42, kde záznam kontaktu s Id=42 neexistuje, vrátí odpověď s kódem 400 NotFound.

## Nástroje

Nástroje platformy .NET umožňují spouštění aplikačního rozhraní v paměti a posílání dotazů na koncové body dokonce bez nutnosti použití protokolu HTTP. Stačí vytvořit instanci `HttpServeru`, která se bude starat o vytvoření instance API aplikace a její hostování v unit testech.

Samotné testy pak mohou být definovány s využitím nástroje `NUnit` a automatizovány stejným způsobem jako jednotkové testy na této vrstvě.

## Testování uživatelského rozhraní

Při integračním/systémovém testování uživatelského rozhraní jsou psány tzv. end-to-end testy (E2E testy), které provádějí sekvence dílčích úloh, tvořících společně některou z komplexních funkcí systému, a které ověřují stav aplikace a uživatelského rozhraní v průběhu celého procesu.

### Rozsah

Cílem je ověřit, že veškeré funkce systému definované prostřednictvím případů užití ve specifikaci požadavků (Specifikace požadavků) jsou správně provedeny a uživatelské rozhraní systému v každém okamžiku odpovídá stavu systému.

Příklady:

1. Uživatel se přihlásí do systému a zobrazí se mu přehled kontaktů.
2. Uživatel zadá řetězec pro filtrování výsledků v přehledu kontaktů a systém zobrazí pouze výsledky odpovídající tomuto řetězci.
3. Uživatel smaže kontakt a systém v přehledu kontaktů zobrazí celkový počet kontaktů snížený o jeden.

## Nástroje

Pro end-to-end testování uživatelského rozhraní postaveného na frameworku `AngularJS` existuje nástroj `Protractor`. Ten umožňuje spouštět testy ve skutečných prohlížečích a interagovat s aplikací stejným způsobem, jako by s ní pracoval skutečný uživatel. Pomocí nástroje `Grunt` lze nastavit automatické spouštění end-to-end testů napsaných v `Protractoru` jako součást procesu sestavení aplikace.

Pro urychlení testování uživatelského rozhraní je možné odstínit UI od aplikačního rozhraní na serverové části aplikace využitím stejného principu zástupných objektů jako při jednotkovém testování UI – pomocí služby `$httpBackend`.

## Akceptační testování

Účelem akceptačního testování je ověření zadavatelem, že je systém opravdu připraven na nasazení do ostrého provozu. Během akceptačního testu koncoví uživatelé porovnají fungování systému se specifikací požadavků.

### Rozsah

Cíle akceptačního testování systému `SYBA SharedBase` můžeme rozdělit do dvou kategorií, podle typu účastníků používajících systém:

Zaměstnanec      Tester na sebe bere roli zaměstnance zadavatele, který má na starosti

udržování dat v systému.

Cílem testování je ověřit, že systém splňuje všechny specifikované požadavky a správně provede veškeré případy užití definované ve specifikaci požadavků (Specifikace požadavků).

Vývojář

Tester na sebe bere roli vývojáře, který vytváří aplikaci přistupující k datům v systému SYBA SharedBase prostřednictvím aplikačního rozhraní.

Cílem je ověřit všechny přístupové body aplikačního rozhraní, správnost odpovědí systému na zasláné dotazy a zabezpečení rozhraní.

## Nástroje

Pro akceptační testování systému SYBA SharedBase nejsou potřeba žádné zvláštní nástroje. Tester dostane k dispozici přehled požadavků na testovaný systém a popis případů užití, seznámí se s testovaným systémem a provede testovací úlohy.



## Závěr

V analytické části této práce se podařilo se zadavatelem vydiskutovat a detailně analyzovat požadavky na systém SYBA SharedBase. Tyto požadavky posloužily jako podklad pro vytvoření softwarového návrhu, na jehož základě byla úspěšně implementována první verze systému, která je v tuto chvíli nasazena v ostrém provozu. Zaměstnanci ČESKÉ OBALOVÉ ASOCIACE, s. r. o., již prostřednictvím webového rozhraní (<http://sharedbase.syba.cz>, snímky webového rozhraní jsou přiloženy v Příloha C: Snímky obrazovky SYBA SharedBase) spravují kontakty osob a společností v databázi a provádí synchronizaci kontaktů a skupin kontaktů se svými účty ve službě Google. Vývojáři zadavatele zároveň využívají dat poskytovaných prostřednictvím SYBA SharedBase API a vybrané společnosti z množiny společností vedených v systému jsou zobrazovány v katalogu dodavatelů obalů a obalových prostředků na stránkách obalového portálu ČESKÉ OBALOVÉ ASOCIACE (<http://syba.cz/dodavatele>, snímky katalogu jsou přiloženy v Příloha D: Snímky obrazovky Katalogu dodavatelů).

V poslední části byla na míru systému SYBA SharedBase navržena metodika testování, zaměřená zejména na jednotkové a integrační testování a jejich automatizaci. V průběhu vývoje systému SYBA SharedBase byl však zadavatelem vznesen požadavek na to, aby byl systém spuštěn co nejdříve. S heslem „better done than perfect“ – lépe alespoň něco hotové než čekat na dokonalý systém. Zadavatel rozhodl, že rizika spojená s chybou v systému nejsou kritická a že implementace všech hlavních funkcí systému verze 1.0 má větší prioritu než napsání testů. Muselo být ustoupeno od postupného, detailního testování celého systému, stoprocentního pokrytí jednotkovými testy, a i za cenu vyšší pravděpodobnosti chyby byl systém nasazen.

Aby však bylo dodrženo zadání diplomové práce, byla nadefinovaná metodika testování na systém přesto aplikována, alespoň ve zmenšené míře. Pro jednotlivé kategorie testů definované ve strategii testování (Strategie testování) bylo nastaveno prostředí a nakonfigurovány ve strategii definované testovací nástroje pro psaní a automatické spouštění testů při sestavování systému. V každé kategorii byly vytvořeny alespoň nějaké testy (Příloha F: Příklady testů), které ověřují správnou konfiguraci nástrojů na testování a které budou sloužit jako podklad pro testery, kteří se budou podílet na dalším vývoji systému. Díky použití moderních technologií a mezi vývojáři oblíbených a rozšířených frameworků při návrhu a následné implementaci systému, jakož i nástrojů zvolených pro testování, je vytváření testů i jejich automatizace velice snadné a na internetu je k dispozici dostatečné množství dokumentace a příkladů.

V tuto chvíli jsou systém SYBA SharedBase, zdrojové kódy a veškerá dokumentace předány zadavateli, který systém bez výhrad přijal a sám bude řídit jeho další vývoj.





## Literatura

- [1] *Cascading Style Sheets*. (2014). Získáno 12. 10. 2014, z Wikipedia: [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- [2] *Černá vs. bílá skříňka*. (2014). Získáno 12. 10. 2014, z SW Testování: [http://swtestovani.cz/index.php?option=com\\_content&view=article&id=23:erna-vs-bila-skika&catid=3:zaklady&Itemid=11](http://swtestovani.cz/index.php?option=com_content&view=article&id=23:erna-vs-bila-skika&catid=3:zaklady&Itemid=11)
- [3] ČESKÁ OBALOVÁ ASOCIACE SYBA. (10. 06. 2011). *O SYBĚ*. Získáno 15. 04. 2014, z Obalový portál: <http://www.syba.cz?inc=117>
- [4] Ash, S. (4. October 2007). *MoSCoW Prioritisation Briefing Paper*. Získáno 7. July 2009, z DSDM Consortium: <http://www.dsdm.org/knowledgebase/details/165/moscow-prioritisation-briefing-paper.html>
- [5] *Dependency injection*. (2014). Získáno 12. 10. 2014, z Wikipedia: [http://en.wikipedia.org/wiki/Dependency\\_injection](http://en.wikipedia.org/wiki/Dependency_injection)
- [6] Fournier, G. (2009). *Essential Software Testing*. Auerbach Publications.
- [7] Gilb, T. (1988). *Principles of Software Engineering Management*. Harlow, England: Addison-Wesley.
- [8] Google. (2014). *What Is Angular?* Získáno 12. 10. 2014, z AngularJS: <https://docs.angularjs.org/guide/introduction>
- [9] *HTML5*. (2014). Získáno 12. 10. 2014, z Wikipedia: <http://cs.wikipedia.org/wiki/HTML5>
- [10] Koźmic, K. (27. 05 2014). *Castle Windsor*. Získáno 12. 10. 2014, z Castle Project: <http://docs.castleproject.org/Default.aspx?Page=MainPage&NS=Windsor>
- [11] Microsoft. (2014). *ASP.NET Web API*. Získáno 12. 10. 2014, z ASP.NET: <http://www.asp.net/web-api>
- [12] *NHibernate*. (2014). Získáno 12. 10. 2014, z NHibernate: <http://nhforge.org/>
- [13] *Objektově relační mapování*. (2014). Získáno 12. 10. 2014, z Wikipedia: [http://cs.wikipedia.org/wiki/Objektov%C4%9B\\_rela%C4%8Dn%C3%AD\\_mapov%C3%A1n%C3%AD](http://cs.wikipedia.org/wiki/Objektov%C4%9B_rela%C4%8Dn%C3%AD_mapov%C3%A1n%C3%AD)
- [14] Oshero, R. (2014). *The Art of Unit Testing* (2. vyd.). Manning.
- [15] *Single-page application*. (2014). Získáno 12. 10. 2014, z Wikipedia: [http://en.wikipedia.org/wiki/Single-page\\_application](http://en.wikipedia.org/wiki/Single-page_application)
- [16] *Testování softwaru*. (2014). Získáno 12. 10. 2014, z Wikipedie: [http://cs.wikipedia.org/wiki/Testov%C3%A1n%C3%AD\\_softwaru](http://cs.wikipedia.org/wiki/Testov%C3%A1n%C3%AD_softwaru)

- [17] Tonar, J. (2013. 10 15). *Jak uplatnit principy třívrstvé architektury v rámci web integračního projektu*. Získáno 15. 10. 2014, z Vážně i nevážně blogujeme o webové integraci.: <http://www.web-integration.info/cs/blog/jak-uplatnit-principy-trivrstve-architektury-v-ramci-web-integracniho-projektu/>
- [18] Wiegers, K. E. (2008). *Požadavky na software*. Brno: Computer Press, a.s.

## Příloha A: Zkratky a akronymy

Zkratka	Definice
<b>API</b>	Application Programming Interface Programové rozhraní aplikace
<b>CSS</b>	Cascading Style Sheets Kaskádové styly
<b>CRUD</b>	Create Read Update Delete Vytvoření, čtení, modifikace, smazání
<b>DTO</b>	Data Transfer Object Objekt pro přenos dat
<b>HTML</b>	Hypertext Markup Language Hypertextový značkovací jazyk
<b>HTTP</b>	Hypertext Transfer Protocol Hypertextový přenosový protokol
<b>HTTPS</b>	Hypertext Transfer Protocol Secure Zabezpečený hypertextový přenosový protokol
<b>JSON</b>	JavaScript Object Notation Formát na výměnu dat
<b>MVC</b>	Model View Controller Architektonický vzor model, pohled, řadič
<b>NPM</b>	Node.js Package Manager Správce balíčků Node.js
<b>ORM</b>	Object Relational Mapping Objektově-relační mapování
<b>PDF</b>	Portable Document Format Přenostný formát dokumentu
<b>POCO</b>	Plain Old CLR Object
<b>SPA</b>	Single-page Application Single-page aplikace
<b>SQL CE</b>	SQL Compact Edition SQL Kompaktní edice
<b>SW</b>	Software
<b>UC</b>	Use Case Případ užití
<b>UI</b>	User Interface Uživatelské rozhraní
<b>URL</b>	Uniform Resource Locator Jednotný popis umístění zdroje
<b>XML</b>	eXtensible Markup Language Rozšiřitelný značkovací jazyk



## Příloha B: Návrh uživatelského rozhraní

Tento dodatek obsahuje návrhy nízké úrovně podrobnosti uživatelského rozhraní systému SYBA SharedBase. Slouží jako podklady o podobě UI pro grafika a vývojáře uživatelského rozhraní systému.

SYBA SharedBase

Přihlášení do systému

Uživatelské jméno:

Heslo:

Přihlásit trvale

Přihlásit

Příloha B: Návrh UI 1 – SYBA SharedBase: Přihlášení







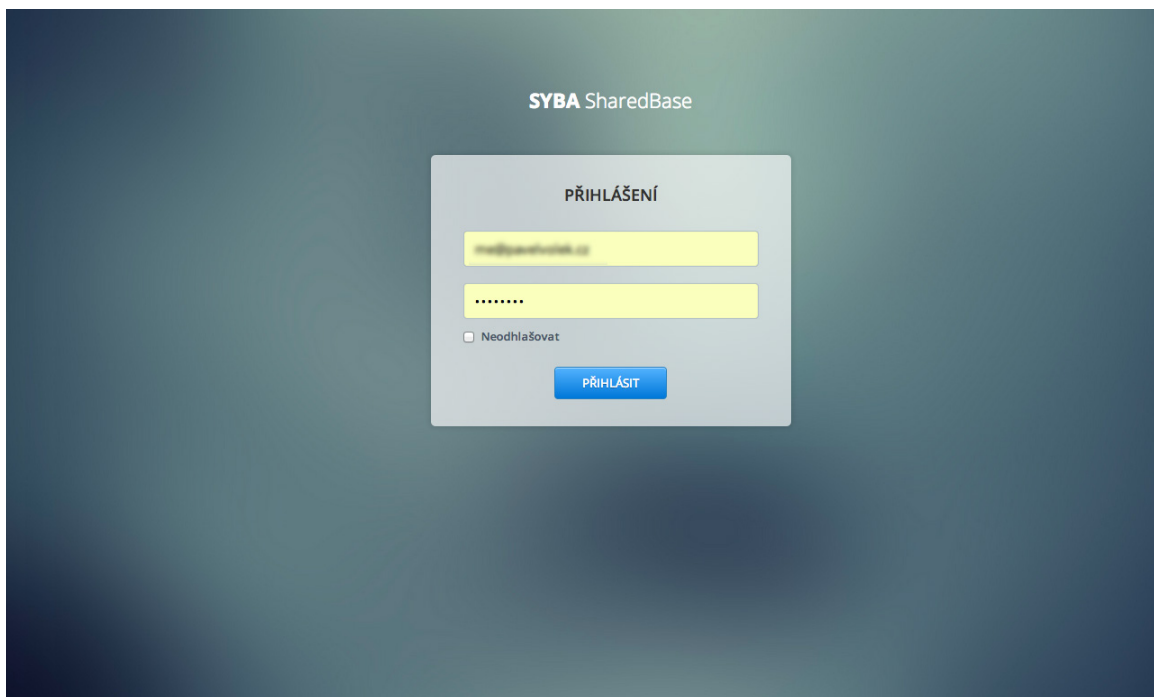
SYBA SharedBase
Odlásit

Kontakty	<a href="#">Přidat účet</a>								
Skupiny kontaktů	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; width: 25%;">Účet</th> <th style="text-align: left; width: 25%;">Stav kontaktů</th> <th style="text-align: left; width: 25%;">Stav skupin</th> <th style="text-align: left; width: 25%;">Akce</th> </tr> </thead> <tbody> <tr> <td>josef@gmail.com</td> <td>Nové: 23, upravené: 32, ke smazání 3</td> <td>Nové: 2, upravené: 0, ke smazání 0</td> <td><a href="#">Synchronizovat</a> <a href="#">Změnit heslo</a> <a href="#">Smazat</a></td> </tr> </tbody> </table>	Účet	Stav kontaktů	Stav skupin	Akce	josef@gmail.com	Nové: 23, upravené: 32, ke smazání 3	Nové: 2, upravené: 0, ke smazání 0	<a href="#">Synchronizovat</a> <a href="#">Změnit heslo</a> <a href="#">Smazat</a>
Účet	Stav kontaktů	Stav skupin	Akce						
josef@gmail.com	Nové: 23, upravené: 32, ke smazání 3	Nové: 2, upravené: 0, ke smazání 0	<a href="#">Synchronizovat</a> <a href="#">Změnit heslo</a> <a href="#">Smazat</a>						
Společnosti	<p>Probíhá synchronizace</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; border-bottom: 1px solid #ccc;">           Mazání kontaktů: 10 / 23  <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 10%, white 10%);"></div> </td> <td style="width: 50%; border-bottom: 1px solid #ccc;">           Mazání skupin: 0 / 2  <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 0%, white 0%);"></div> </td> </tr> <tr> <td style="border-bottom: 1px solid #ccc;">           Úpravy kontaktů: 0 / 32  <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 0%, white 0%);"></div> </td> <td style="border-bottom: 1px solid #ccc;">           Úpravy skupin: 0 / 0  <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 100%, white 100%);"></div> </td> </tr> <tr> <td style="border-bottom: 1px solid #ccc;">           Mazání kontaktů: 0 / 3  <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 0%, white 0%);"></div> </td> <td style="border-bottom: 1px solid #ccc;">           Mazání skupin: 0 / 0  <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 100%, white 100%);"></div> </td> </tr> </table> <p style="text-align: right; margin-top: 10px;"> <a href="#">Zrušit</a>   <a href="#">Pozastavit</a> </p>	Mazání kontaktů: 10 / 23 <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 10%, white 10%);"></div>	Mazání skupin: 0 / 2 <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 0%, white 0%);"></div>	Úpravy kontaktů: 0 / 32 <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 0%, white 0%);"></div>	Úpravy skupin: 0 / 0 <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 100%, white 100%);"></div>	Mazání kontaktů: 0 / 3 <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 0%, white 0%);"></div>	Mazání skupin: 0 / 0 <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 100%, white 100%);"></div>		
Mazání kontaktů: 10 / 23 <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 10%, white 10%);"></div>	Mazání skupin: 0 / 2 <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 0%, white 0%);"></div>								
Úpravy kontaktů: 0 / 32 <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 0%, white 0%);"></div>	Úpravy skupin: 0 / 0 <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 100%, white 100%);"></div>								
Mazání kontaktů: 0 / 3 <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 0%, white 0%);"></div>	Mazání skupin: 0 / 0 <div style="width: 100%; height: 10px; background: linear-gradient(to right, black 100%, white 100%);"></div>								
Kategorie společností									
Synchronizace	<table style="width: 100%; border-collapse: collapse;"> <tbody> <tr> <td style="width: 25%;">voprsal@gmail.com</td> <td style="width: 25%;">Nové: 23, upravené: 32, ke smazání 3</td> <td style="width: 25%;">Nové: 23, upravené: 32, ke smazání 3</td> <td style="width: 25%;"><a href="#">Synchronizovat</a> <a href="#">Změnit heslo</a> <a href="#">Smazat</a></td> </tr> <tr> <td>petr.novotny@gmail.com</td> <td>Nové: 23, upravené: 32, ke smazání 3</td> <td>Nové: 23, upravené: 32, ke smazání 3</td> <td><a href="#">Synchronizovat</a> <a href="#">Změnit heslo</a> <a href="#">Smazat</a></td> </tr> </tbody> </table>	voprsal@gmail.com	Nové: 23, upravené: 32, ke smazání 3	Nové: 23, upravené: 32, ke smazání 3	<a href="#">Synchronizovat</a> <a href="#">Změnit heslo</a> <a href="#">Smazat</a>	petr.novotny@gmail.com	Nové: 23, upravené: 32, ke smazání 3	Nové: 23, upravené: 32, ke smazání 3	<a href="#">Synchronizovat</a> <a href="#">Změnit heslo</a> <a href="#">Smazat</a>
voprsal@gmail.com	Nové: 23, upravené: 32, ke smazání 3	Nové: 23, upravené: 32, ke smazání 3	<a href="#">Synchronizovat</a> <a href="#">Změnit heslo</a> <a href="#">Smazat</a>						
petr.novotny@gmail.com	Nové: 23, upravené: 32, ke smazání 3	Nové: 23, upravené: 32, ke smazání 3	<a href="#">Synchronizovat</a> <a href="#">Změnit heslo</a> <a href="#">Smazat</a>						

**Příloha B: Návrh UI 6 – SYBA SharedBase: Průběh synchronizace s Google účtem**

## Příloha C: Snímky obrazovky SYBA SharedBase

Tento dodatek obsahuje snímky obrazovek ze systému SYBA SharedBase nasazeného do reálného provozu.



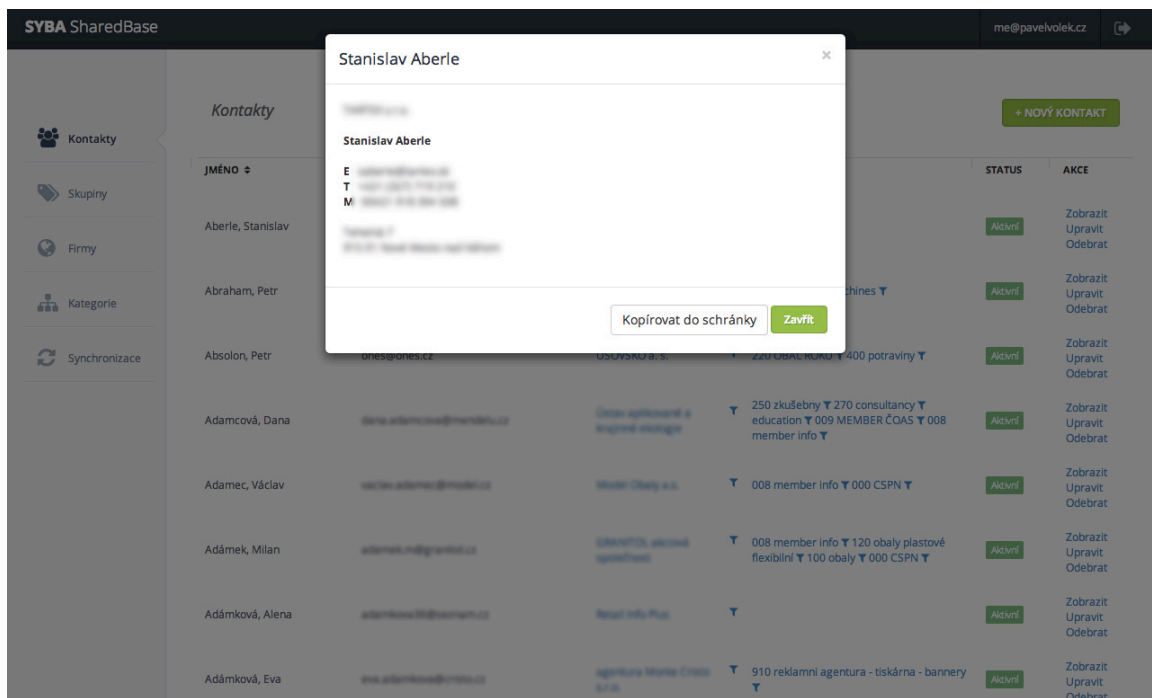
Příloha C: Snímek 1 – SYBA SharedBase: Přihlášení

JMÉNO	EMAIL	SPOLEČNOST	SKUPINY	STATUS	AKCE
Aberle, Stanislav			000 CSPN	Aktivní	Zobrazit Upravit Odebrat
Abraham, Petr			300 Packaging Machines	Aktivní	Zobrazit Upravit Odebrat
Absolon, Petr			220 OBAL ROKU 400 potraviny	Aktivní	Zobrazit Upravit Odebrat
Adamcová, Dana			250 zkušebny 270 consultancy education 009 MEMBER ČOAS 008 member info	Aktivní	Zobrazit Upravit Odebrat
Babušková, Dana			901 asociace svazy 181 palety black list	Aktivní	Zobrazit Upravit Odebrat

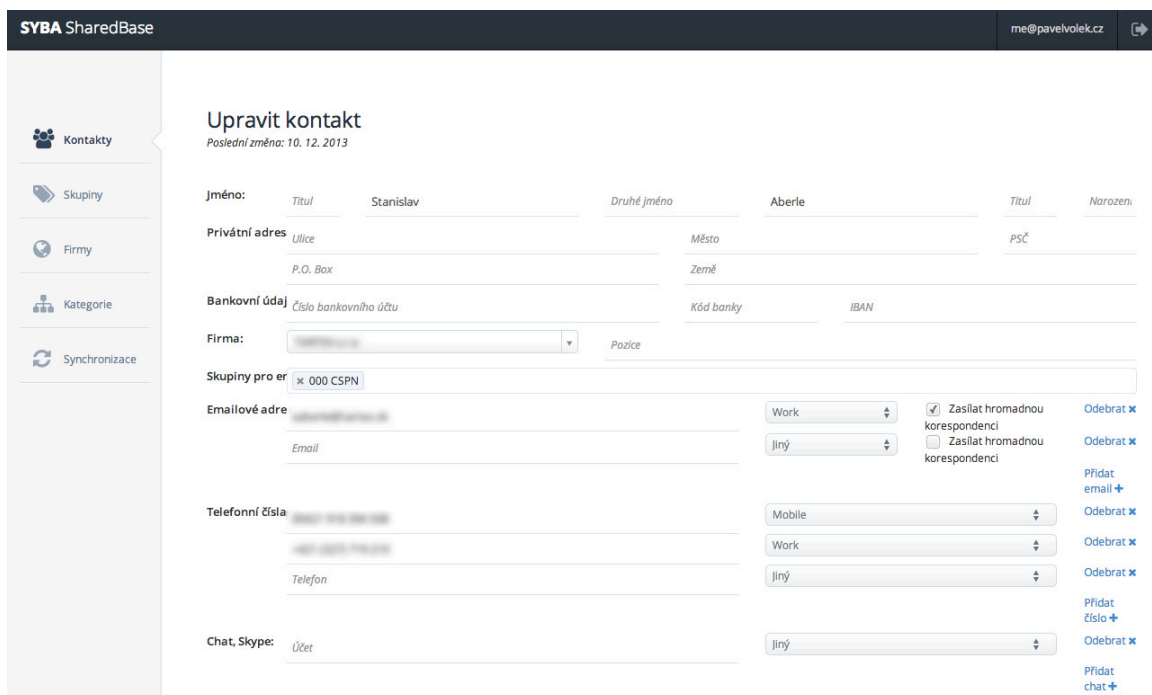
Total: 3655

1 2 3 4 5 6 7 8 9 10 > >>

Příloha C: Snímek 2 – SYBA SharedBase: Přehled kontaktů



Příloha C: Snímek 3 – SYBA SharedBase: Detail kontaktu



Příloha C: Snímek 4 – SYBA SharedBase: Formulář na úpravu kontaktu

SYBA SharedBase me@pavelvolek.cz

**Skupiny kontaktů** + NOVÁ SKUPINA

NÁZEV	POSLEDNÍ ZMĚNA	STATUS	
000 CSPN	12/3/2013 5:25:27 PM	Aktivní	Export do CSV Odebrat Upravit
000 Fachpack 2013	12/3/2013 5:25:38 PM	Aktivní	Export do CSV Odebrat Upravit
000 mailman	12/3/2013 5:25:26 PM	Aktivní	Export do CSV Odebrat Upravit
000 zero N	12/3/2013 5:30:31 PM	Aktivní	Export do CSV Odebrat Upravit
0000 PĚNY	12/3/2013 5:25:30 PM	Aktivní	Export do CSV Odebrat Upravit
0000test	12/3/2013 5:32:59 PM	Aktivní	Export do CSV Odebrat Upravit
008 member info	12/3/2013 5:25:27 PM	Aktivní	Export do CSV Odebrat Upravit
009 MEMBER ČOAS	12/3/2013 5:25:27 PM	Aktivní	Export do CSV Odebrat Upravit
010 member SYBA	12/3/2013 5:25:27 PM	Aktivní	Export do CSV Odebrat Upravit
011 member SYBA neosílat	12/3/2013 5:25:31 PM	Aktivní	Export do CSV Upravit

Příloha C: Snímek 5 – SYBA SharedBase: Přehled skupin kontaktů

SYBA SharedBase me@pavelvolek.cz

**Firmy a organizace** + NOVÝ ZÁZNAM

Filtr

NÁZEV	KONTAKTY	ADRESA	KONTAKTNÍ EMAIL	TELEFON	
1. moravsko - česká vinařská s.r.o.	<a href="#">Kontakt</a>	Wenceslava 101, 712 01 Písek 8, Horní Počapky			Upravit Odebrat
3 M Česko, spol. s r.o.	<a href="#">Kontakt</a>	V Parku 200/20, 148 00 Praha 4, Česká republika		+420 281 280 111	Upravit Odebrat
3H creative studio, s. r. o.	<a href="#">Kontakt</a>	Spa. Travná 26, 601 01 Brno, Česká republika	<a href="#">kreativni@3hcreative.cz</a>		Upravit Odebrat
4P FLEXIBLES PRAHA, s.r.o.	<a href="#">Kontakt</a>	Janáčkova 26, 148 00 Praha 4, Česká republika			Upravit Odebrat
A ASKA GRAFIK s.r.o.	<a href="#">Kontakt</a>	Čestmírová 11, 610 00 Brno, Česká republika	<a href="#">info@aska.cz</a>	+420 543 241 470	Upravit Odebrat
A.A. Potištné lepicí pásy, s.r.o.	<a href="#">Kontakt</a>	Benátská 26, 148 00 Praha 4, Česká republika	<a href="#">info@aa.cz</a>	+420 281 280 111	Upravit Odebrat
A.B.ISACSON ASSOCIATES, Inc	<a href="#">Kontakt</a>	20 Broadway, Suite 1200, NY 10006 New York, USA			Upravit Odebrat
A.W. Faber-Castell spol. s r.o.	<a href="#">Kontakt</a>	U Hrádků 1011, 272 01 Hrádky nad Moravou, Česká republika	<a href="#">info@aw-castell.cz</a>	+420 281 622 700	Upravit Odebrat
A.W. spol. s r.o.	<a href="#">Kontakt</a>	Pravá 10, 700 00 Olomouc, Česká republika	<a href="#">aw@aw-castell.cz</a>		Upravit Odebrat

sharedbase.syba.cz/Organization

Příloha C: Snímek 6 – SYBA SharedBase: Přehled společností

SYBA SharedBase me@pavelvolek.cz

- Kontakty
- Skupiny
- Firmy
- Kategorie
- Synchronizace

### Kategorie firem + NOVÁ KATEGORIE

Klikněte na "Uložit změny" pro uložení vámi provedených změn.

obaly a obalové materiály → (Containers & Packaging Materials →)	Upravit   Smazat
ampule → (Ampoules →)	Upravit   Smazat
ampule plastové (Ampoules - Plastic)	Upravit   Smazat
ampule skleněné (Ampoules - Glass)	Upravit   Smazat
bedny → (Boxes, Crates →)	Upravit   Smazat
bedny dřevěné (Wooden Boxes, Crates)	Upravit   Smazat
bedny kovové ()	Upravit   Smazat
bio obaly (Bio Packaging/Containers)	Upravit   Smazat
dózy, dóžičky plastové ()	Upravit   Smazat
dutinky, ochranné hrany (Cores, Protective Edges)	Upravit   Smazat
etikety, sleeve etikety, IML, RFID → (Labels, Stickers, Shrink Sleeves, IML, RFID →)	Upravit   Smazat
samolepicí etikety (Self-adhesive Labels, Stickers)	Upravit   Smazat
etikety grafické ( Wet Labels)	Upravit   Smazat
wrap around etikety (Wrap Around Labels)	Upravit   Smazat
shrink sleeves (smrštitelné (rukávové) etikety, šrinky) (Shrink Sleeves)	Upravit   Smazat
IML (in-mould-labels) (IML)	Upravit   Smazat
RFID (radio frekvence Identification) (RFID)	Upravit   Smazat

Příloha C: Snímek 7 - SYBA SharedBase: Přehled kategorií společností

SYBA SharedBase me@pavelvolek.cz

- Kontakty
- Skupiny
- Firmy
- Kategorie
- Synchronizace

### Synchronizace se službou Google Kontakty

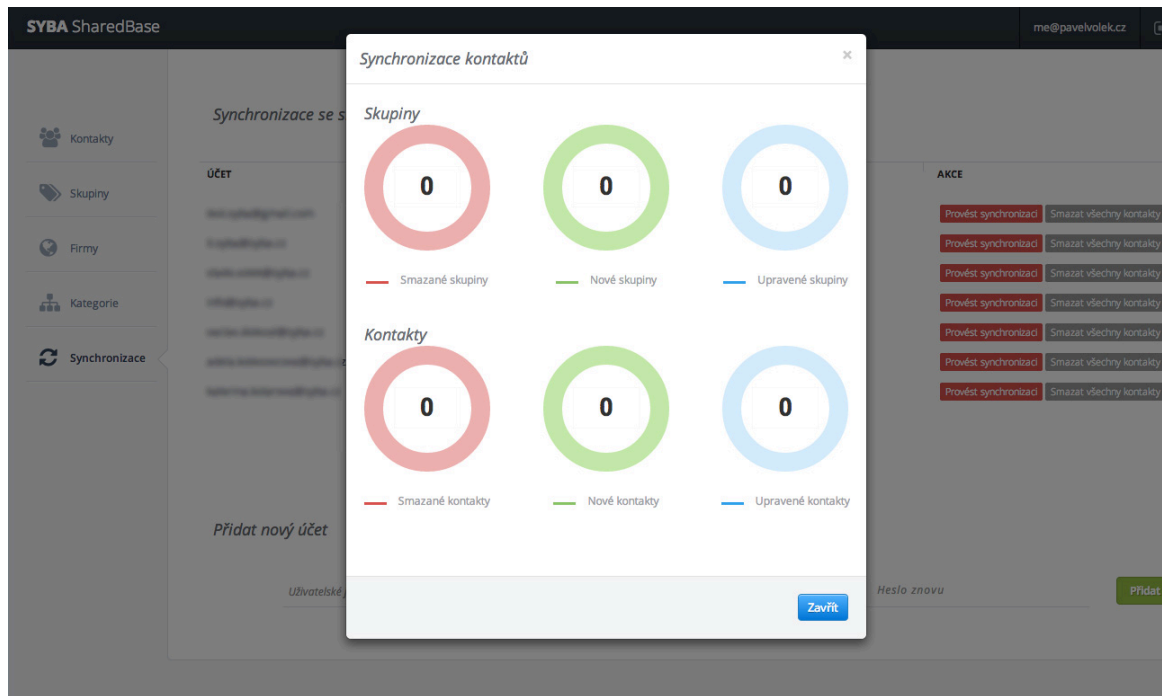
ÚČET	KONTAKTY	SKUPINY	AKCE
	20 nové, 60 upravené, 0 smazané	0 nové, 0 upravené, 0 smazané	<span style="background-color: #f44336; color: white; padding: 2px 5px;">Provést synchronizaci</span> <span style="background-color: #f44336; color: white; padding: 2px 5px;">Smazat všechny kontakty</span>
	3 nové, 22 upravené, 0 smazané	0 nové, 0 upravené, 0 smazané	<span style="background-color: #f44336; color: white; padding: 2px 5px;">Provést synchronizaci</span> <span style="background-color: #f44336; color: white; padding: 2px 5px;">Smazat všechny kontakty</span>
	15 nové, 33 upravené, 0 smazané	0 nové, 0 upravené, 0 smazané	<span style="background-color: #f44336; color: white; padding: 2px 5px;">Provést synchronizaci</span> <span style="background-color: #f44336; color: white; padding: 2px 5px;">Smazat všechny kontakty</span>
	3 nové, 22 upravené, 0 smazané	0 nové, 0 upravené, 0 smazané	<span style="background-color: #f44336; color: white; padding: 2px 5px;">Provést synchronizaci</span> <span style="background-color: #f44336; color: white; padding: 2px 5px;">Smazat všechny kontakty</span>
	39 nové, 144 upravené, 0 smazané	0 nové, 0 upravené, 0 smazané	<span style="background-color: #f44336; color: white; padding: 2px 5px;">Provést synchronizaci</span> <span style="background-color: #f44336; color: white; padding: 2px 5px;">Smazat všechny kontakty</span>

Přidat nový účet

Přidat

sharedbase.syba.cz/Synchronization

Příloha C: Snímek 8 – SYBA SharedBase: Stav synchronizace s Google účty



Příloha C: Snímek 9 – SYBA SharedBase: Aktuální průběh synchronizace



## Příloha D: Snímky obrazovky Katalogu dodavatelů

Tento dodatek obsahuje snímky obrazovek z Katalogu dodavatelů ve webové aplikaci Obalový portál (<http://syba.cz/dodavatele>). Záznamy kategorií a firem v tomto katalogu jsou získávány prostřednictvím aplikačního rozhraní SYBA SharedBase (<http://api.syba.cz>).

The screenshot shows the SYBA supplier catalog interface. The header includes the SYBA logo and navigation links. The main content area is titled "KATALOG DODAVATELŮ CZ+SK". Search filters are set to "balicí stroje a linky" and "etiketovací, dotiskovací, označovací zařízení". The results list includes:

- BOTTLING PRINTING, s.r.o.** (www.bottlingprinting.cz): laserový tisk, ink-jet a termotransfer technologie, aplikátory a tiskárny etiket
- EPRIN spol. s r.o.** (www.eprin.cz): samolepicí etikety a etikety RFID, snímače RFID a čárového kódu, tiskárny  
Adresa: Hybešova 38, Brno 602 00, Česká republika; Telefon: +420 538 707 200-2; E-mail: eprin@eprin.cz
- REKO PRINT, s.r.o.** (www.reko-print.cz): aplikátory etiket, inkjetové, termotransferové a laserové tiskárny
- AJP-tech spol. s r.o.** (www.ajptech.cz): tiskárny Ink-jet
- ALBERTINA Trading, spol. s r.o.** (www.albertina-trading.cz): kontrolní váhy, pásové váhy, podtlakové manipulátory, etiketovací stroje, paletizační linky, pinčící linky

Příloha D: Katalog 1 – Detail firmy

The screenshot shows the SYBA supplier catalog interface with search filters set to "obaly a obalové materiály" and "etikety, sleeve etikety, IML, RFID". A dropdown menu for "vyberte podkategorii" is open, showing options like "etikety grafické", "IML (in-mould-labels)", "RFID (radio frekvency Identification)", "samolepicí etikety", "shrink sleeves (smršťitelné (rukávové) etikety, šrinky)", "wrap around etikety", and "samoolepicí etikety". The results list includes:

- aluprint.gaya.sk**: papírové etikety, samolepicí etikety, samolepicí etikety
- www.eprin.cz**: samolepicí etikety, shrink sleeves (smršťitelné (rukávové) etikety, šrinky), wrap around etikety
- www.grafobalartypa.cz**: krabice z vlnité lepenky, skládačky, etikety, papírové sáčky
- www.grafobal.sk**: krabice z vlnité lepenky, skládačky, etikety, sáčky papírové, leporela, puzzle
- www.peroutka.cz**: samolepicí a smršťitelné etikety, etikety do tiskáren, laminované potíštěné fólie, papírové dutinky, hliníková víčka
- www.otk.cz**: grafické etikety, shrink sleeve, IML, flexibilní obaly
- www.aledeto.cz/**: ALEDETO s. r. o.

Příloha D: Katalog 2 - Výběr kategorie





# Příloha E: Snímky obrazovky SYBA SharedBase API

Tento dodatek obsahuje snímky obrazovek aplikačního rozhraní SYBA SharedBase API (<http://api.syba.cz>).

## ASP.NET Web API Help Page

### Introduction

Provide a general description of your APIs here.

### Organizations

API	Description
GET api/v2/organizations?offset={offset}&size={size}	Get paged list of organizations.
GET api/v2/organizations/{id}	Get organization details.
GET api/v2/organizations/{id}/employees	Get the employees list for organization.
POST api/v2/organizations	Create new organization.
PUT api/v2/organizations/{id}	Update organization.
DELETE api/v2/organizations/{id}	Delete organization.

### Contacts

API	Description
GET api/v2/contacts?offset={offset}&size={size}	Get paged list of contacts.
GET api/v2/contacts/{id}	Get contact details.
GET api/v2/contacts/{id}/groups	Get the group list for contact.
POST api/v2/contacts	Create new contact.
PUT api/v2/contacts/{id}	Updates contact.
DELETE api/v2/contacts/{id}	Delete contact.

### Categories

API	Description
-----	-------------

## Příloha E: API 1 – Přehled koncových bodů

[Help Page Home](#)

### GET api/v2/organizations/{id}/employees

Get the employees list for organization.

#### Request Information

##### URI Parameters

Name	Description	Type	Additional information
id	The organization id to get employees for.	Integer	Required

##### Body Parameters

None.

#### Response Information

##### Resource Description

List of all employees of the given organization.Resource not found error.

IHttpActionResult

None.

##### Response Formats

application/json, text/json, application/xml, text/xml

Sample:  
Sample not available.

## Příloha E: API 2 – Detail koncového bodu



## Příloha F: Příklady testů

Tento dodatek obsahuje přehled typových příkladů testů jednotlivých komponent systému SYBA SharedBase, které byly vytvořeny pomocí nástrojů definovaných v kapitole Testování, sekci Strategie testování.

### Jednotkový test mapování

Následující testovací třída z komponenty SYBA.SharedBase.Data.NHibernate.Tests ověřuje prostřednictvím knihovny NHibernate mapování atributů objektu Organization do relační databáze. Ukazuje, jak je vytvořit novou instanci Organization, vložit Organization do databáze, načíst záznam z databáze do nové instance Organization a ověřit, že načtený záznam odpovídá originálu.

```

1. // Test correctness of Organization mapping using NHibernate
2. [TestFixture]
3. public class OrganizationTests : AbstractInMemoryDataFixture
4. {
5.     var organizations = new List<Organization>() {
6.         new Organization {Name = "Test Member Organization 1"},
7.         new Organization {Name = "Test Member Organization 2"}
8.     };
9.
10.    [Test]
11.    public void CanCorrectlyMapOrganization()
12.    {
13.        new PersistenceSpecification<Organization>(Session,
14.            new CustomEqualityComparer())
15.            .CheckProperty(c => c.LastUpdate, DateTime.Now)
16.            .CheckProperty(c => c.Name, "Test Name")
17.            .CheckProperty(c => c.Abbreviation, "TN")
18.            .CheckProperty(c => c.IsPublic, true)
19.            .CheckProperty(c => c.CompanyDescription, "Description")
20.            .CheckProperty(c => c.CompanyDescription_en, "Descript. EN")
21.            .CheckProperty(c => c.DeliveryProgramme, "Programme")
22.            .CheckProperty(c => c.DeliveryProgramme_en, "Programme EN")
23.            .CheckProperty(c => c.CompanyRegistrationNumber, "IC")
24.            .CheckProperty(c => c.TaxIdentificationNumber, "DIC")
25.            .CheckProperty(c => c.Website, "http://www.test.com")
26.            .CheckProperty(c => c.ContactPhone, "+420 777 222 444")
27.            .CheckProperty(c => c.ContactFax, "+420 222 222 444")
28.            .CheckProperty(c => c.ContactDataBox, "+420 222 222 444")
29.            .CheckProperty(c => c.ContactEmail, "info@test.com")
30.            .CheckReference(c => c.Address, new Address() {
31.                Street = "St. Test 1242/10",
32.                City = "Washington",
33.                Country = "USA"
34.            })
35.            .CheckProperty(c => c.LogoUrl, "Logo Uri")
36.            .CheckList(
37.                c => c.MemberOrganizations,
38.                organizations,
39.                (association, member) =>
40.                    association.AddMemberOrganization(member))
41.            .VerifyTheMappings();
42.    }
43. }

```

## Jednotkový test servisní třídy

Následující testovací třída z komponenty SYBA.SharedBase.Services.Tests ověřuje správnou funkčnost servisní třídy ContactsService obsahující business logiku nad entitou Contact. Níže uvedený test ověřuje, že metoda GetContactSelectList správně sestavuje seznam pouze z kontaktů, které nebyly označeny ke smazání. Aby byla testovaná třída odstíněna od implementace objektu ContactRepository, jsou toto rozhraní nahrazeno zástupným objektem – mock objektem, vytvořeným pomocí knihovny Moq.

```
44. [TestClass]
45. public class ContactServiceTests
46. {
47.     [TestMethod]
48.     public void GetsSelectListFromContacts()
49.     {
50.         // Arrange
51.         var contacts = new List<Contact>() {
52.             new Contact { Id = 1, Active = true },
53.             new Contact { Id = 2, Active = true },
54.             new Contact { Id = 3, Active = false }
55.         };
56.         var mockRepository = new Mock<IContactRepository>();
57.         IQueryable<Contact> queryableContacts = contacts.AsQueryable();
58.         mockRepository
59.             .Setup(x => x.GetAll())
60.             .Returns(queryableContacts);
61.         var service = new ContactService(mockRepository.Object);
62.
63.         // Act
64.         var result = service.GetContactSelectList();
65.
66.         // Assert
67.         Assert.IsNotNull(result);
68.         Assert.That(result.Count(), Is.EqualTo(2));
69.         Assert.That(result,
70.             Has.Some.Matches<SelectListItem>(x => x.Value == "1"));
71.         Assert.That(result,
72.             Has.Some.Matches<SelectListItem>(x => x.Value == "2"));
73.     }
74.
75.     // ...
76.
77. }
```

## Jednotkový test aplikačního rozhraní

Následující testovací třída z komponenty SYBA.SharedBase.API.Test ověřuje správnou funkčnost třídy aplikačního rozhraní ContactsController. Testovaná třída je opět odstíněna od implementace závislostí použitím zástupných objektů.

```
78. [TestClass]
79. public class TestContactsController
80. {
81.     [TestMethod]
82.     public void GetReturnsContactsWithSameId()
83.     {
84.         // Arrange
85.         var mockService = new Mock<IContactService>();
86.         mockService
87.             .Setup(x => x.GetContact(42))
88.             .Returns(new Contact());
89.         var controller = new ContactsController(mockService.Object);
90.
```

```

91.     // Act
92.     IHttpActionResult actionResult = controller.Get(42);
93.
94.     // Assert
95.     var contentResult = actionResult as
96.         OkNegotiatedContentResult<Contact>;
97.     Assert.IsNotNull(contentResult);
98.     Assert.IsNotNull(contentResult.Content);
99.     Assert.AreEqual(42, contentResult.Content.Id);
100. }
101.
102. [TestMethod]
103. public void GetReturnsNotFound()
104. {
105.     // Arrange
106.     var mockService = new Mock<IContactService>();
107.     var controller = new ContactsController(mockService.Object);
108.
109.     // Act
110.     IHttpActionResult actionResult = controller.Get(42);
111.
112.     // Assert
113.     Assert.IsInstanceOfType(actionResult, typeof(NotFoundResult));
114. }
115.
116. // ...
117.
118. }

```

## Jednotkový test uživatelského rozhraní

Následující testovací skript z komponenty SYBA.SharedBase.Frontend ověřuje AngularJS řadič ContactListCtrl, jehož zodpovědností je dotazem na API získat pole kontaktů a to následně zobrazit. Níže uvedený test je napsaný v testovacím frameworku Jasmine a ověřuje schopnost načíst seznam kontaktů. Služba volající aplikační rozhraní prostřednictvím HTTP protokolu je nahrazena zástupným objektem.

```

119. describe('Controller: ContactListCtrl', function() {
120.     var scope, httpBackend, ContactListCtrl;
121.
122.     // load the controller's module
123.     beforeEach(module('sharedbaseApp'));
124.
125.     // Initialize the controller and a mock scope
126.     beforeEach(inject(function ($controller, $rootScope, $httpBackend)
127.     {
128.         httpBackend = $httpBackend;
129.         scope = $rootScope.$new();
130.         ContactListCtrl = $controller('ContactListCtrl', {
131.             $scope: scope
132.         });
133.     }));
134.
135.     // Ensure no outstanding expectaions or requests after
136.     // test has been run
137.     afterEach(function() {
138.         httpBackend.verifyNoOutstandingExpectation();
139.         httpBackend.verifyNoOutstandingRequest();
140.     });
141.
142.     it('should fetch list of contacts', function(){
143.         var controller = createController();
144.         httpBackend.expect('GET', '/contacts?offset=0&limit=2')
145.             .respond({
146.                 "total": 3400,

```

```

147.         "pageSize":2,
148.         "pageIndex":0,
149.         "items": [
150.             {"id": 1,"fullName": "John Deer"}
151.             {"id": 2,"fullName": "Samuel L. Jackson"}
152.         ]
153.     });
154.
155.     $httpBackend.flush();
156.     expect(scope.contacts.length).toBe(2);
157.     expect(scope.contacts[0].id).toBe(1);
158.     expect(scope.contacts[1].id).toBe(2);
159. });
160. });

```

## Integrační E2E test

Následující testovací skript je také z komponenty SYBA.SharedBase.Frontend. Popisuje end-to-end test, jehož cílem je ověřit, že po úspěšném přihlášení do správy kontaktů je stránka přeměrována na úvodní stránku. Test je napsaný ve frameworku Protractor (který je postavený na frameworku Jasmine, takže má stejnou syntax). Ten umožňuje definovat takové testy, které interagují se systémem stejným způsobem jako by s nimi pracoval skutečný uživatel.

```

161. describe('Homepage', function() {
162.     var ptor, username, password, loginButton;
163.
164.     ptor = protractor.getInstance();
165.
166.     beforeEach(function() {
167.         ptor.get('#/login');
168.         username = element(by.model('userName'));
169.         password = element(by.model('password'));
170.         loginButton = element(by.id('loginButton'));
171.     });
172.
173.     it('should navigate to homepage after login', function() {
174.         loginButton.click().then(function() {
175.             ptor.waitForAngular();
176.             expect(ptor.getCurrentUrl()).toContain('#/');
177.         });
178.     });
179. });

```

## Příloha G: Obsah přiloženého CD

```
CD
| Text
|   DP_Pavel-Volek_2015_SYBA-SharedBase.doc
|   DP_Pavel-Volek_2015_SYBA-SharedBase.pdf
|   Attachments
|     Screenshots
|     Models
Src
|   SYBA.SharedBase
|   SYBA.SharedBase.Frontend
| README.txt
```