



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Webová aplikace pro objednávky a plánování výroby
Student:	Jan Prášil
Vedoucí:	Ing. Jan Hořejší
Studijní program:	Informatika
Studijní obor:	Web a multimédia
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2016/17

Pokyny pro vypracování

Cílem práce je analyzovat, navrhnout a následně implementovat serverovou stranu aplikace pro správu objednávek, vytváření plánu výroby a zadávání výsledků výroby pro malou společnost. Dále pak implementovat klientskou stranu pro webový prohlížeč.

Aplikace bude umožňovat zadávat data, která se často mění (objednávky, šarže materiálu, plán výroby). Ostatní téměř neměnná data budou uložena ve firemním systému SharePoint (data o výrobcích, formách, materiálech, strojích). Tyto data je třeba aktualizovat do vlastní databáze. Základní podporované uživatelské role budou: administrátor, vedoucí výroby a další.

Pokyny:

- analyzujte dosavadní řešení,
- navrhnete aplikaci dle požadavků,
- zvolte vhodné technologie a zvolíte technologii,
- implementujte serverovou stranu,
- navrhnete grafickou podobu webového rozhraní,
- implementujte klientskou stranu,
- aplikaci otestujte,
- navrhnete model nasazení.

Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdlík, CSc.
ředitel katedry

V Praze dne 16. prosince 2015

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

Webová aplikace pro objednávky a plánování výroby

Jan Prášil

Vedoucí práce: Ing. Jan Hořejší

15. května 2016

Poděkování

Tímto bych velmi rád poděkoval rodině a přátelům, kteří mě v průběhu studia podporovali. Nesmím opomenout společnost HORZ s.r.o. a pana Ing. Pavla Korčáka, kteří mi umožnili pojmout tento projekt jako bakalářskou práci. Dále bych chtěl velmi poděkovat vedoucímu této bakalářské práce, panu Ing. Janu Hořejšímu, za cenné rady a nasměrování při vypracování této práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 15. května 2016

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2016 Jan Prášil. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Prášil, Jan. *Webová aplikace pro objednávky a plánování výroby*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

Abstrakt

Tato bakalářská práce se zabývá vytvořením webové aplikace pro správu objednávek a plánování výroby, která vznikla na základě požadavků zákazníka, společnosti HORZ s.r.o. Součástí této práce je výběr vhodného serverového prostředí tak, aby mohla výsledná webová aplikace spolupracovat se stávajícími informačními systémy. Produktem této bakalářské práce je funkční a otestovaná webová aplikace.

Klíčová slova Webová aplikace, plánování výroby, objednávky, C#, .NET, SharePoint

Abstract

This bachelor thesis deals with a creation of an order and production planning web application which arose from customer requirements, HORZ s.r.o. company. The next part of the work is a choice of a suitable server environment to work with existing information systems. The product of this bachelor thesis is a functional and tested web application.

Keywords Web application, production planning, orders, C#, .NET, SharePoint

Obsah

Úvod	1
1 Cíle práce	3
2 Současné řešení	5
2.1 Přístupnost	5
2.2 Microsoft SharePoint	5
2.3 Nevýhody současného řešení	6
3 Analýza	9
3.1 Procesy ve společnosti	9
3.2 Funkční požadavky	11
3.3 Nefunkční požadavky	16
3.4 Případy užití	16
3.5 Uživatelské role	20
3.6 Doménová analýza	24
3.7 Ostatní entity	27
4 Návrh	29
4.1 Použité technologie	29
4.2 Databázový model	35
4.3 Grafické rozhraní	39
5 Realizace	41
5.1 Model nasazení	41
5.2 Struktura projektu	41
6 Testování	47
6.1 Jednotkové testování	47
6.2 Uživatelské testování	47

Závěr	49
Literatura	51
A Seznam použitých zkratk	55
B Instalační příručka	57
B.1 Instalační požadavky	57
B.2 Spuštění aplikace ve webovém prohlížeči	58
C Přílohy	59
D Obsah přiloženého CD	63

Seznam obrázků

2.1	Obrazovka s dokumentem pro export průvodek výrobou.	7
3.1	Diagram procesu přijetí objednávky	10
3.2	Diagram procesu zpracování objednávky	11
3.3	Diagram případů užití pro uživatele.	17
3.4	Diagram případů užití pro správu produkce.	19
3.5	Diagram případu užití pro nepřihlášeného uživatele	21
3.6	Diagram případů užití pro operátora výroby	21
3.7	Diagram případů užití pro vedení výroby	22
3.8	Diagram případu užití pro vedení společnosti	23
3.9	Diagram entit sekce Kontakty	24
3.10	Diagram entit sekce SharePoint	25
3.11	Diagram entit sekce Evidence produkce	26
4.1	Popis architektury MVC	30
4.2	Vnořování komponent v knihovně React	34
4.3	Průvodce generování zdvořitého kódu entit	38
4.4	Příklad modelu entit a jejich vztahů vytvořených pomocí návrhářského nástroje Microsoft Visal Studio	39
4.5	Návrh drátového modelu pro domovskou stranu WAHORZ	40
5.1	Návrh modelu nasazení	42
C.1	Databázový diagram	60
C.2	Úvodní obrazovka WAHORZ pro nepřihlášeného uživatele	61
C.3	Obrazovka s oznámením o úspěšné synchronizaci	61
C.4	Obrazovka s formulářem pro založení nové objednávky	62

Úvod

Společnost HORZ s.r.o. je podnik, který se zabývá, kromě jiných obchodních aktivit, také výrobou a zpracováním plastových výrobků[1].

Pro zadávání objednávek a pro plánování výroby používá společnost v současné době systém Microsoft SharePoint Foundation 2013, (SPF2013), společně s kancelářským softwarem Microsoft Office 2010 (MSO2010). Tyto systémy však, vzhledem k plánovanému růstu společnosti, neposkytují dostatek možností, aby uspokojily firemní potřeby. Setkávají se s mnoha chybami, které způsobují problémy při interní komunikaci mezi jednotlivými zaměstnanci a vedením společnosti, a při firemních auditech. Uživatelé dále nejeví zájem o používání SPF2013 kvůli jeho nedostatečné přehlednosti.

Společnost HORZ s.r.o. je malá firma[2][3], která nesouhlasí s používáním drahých informačních systémů, které by nedokázala dostatečně a plnohodnotně využít. Ty se totiž vyplatí používat především ve velkých výrobních závodech s mnoha zaměstnanci.

Z tohoto důvodu si společnost přála vytvořit webovou aplikaci, která umožní spravovat proces od zadání objednávky a plánování její výroby až po expedici.

Tento proces spočívá v přijetí objednávky vedením společnosti nebo vedením výrobní sekce. Pokračuje zadáním výrobního plánu pro jednotlivé zaměstnance. Po výrobě následuje vyplnění výsledků výroby daným operátorem výroby a končí kontrolou výsledných produktů, expedicí objednávky a následně jejím uzavřením.

Webová aplikace by měla, dle přání vedení společnosti, poskytnout podobnou funkcionalitu, jakou v současné době nabízí stávající řešení. To se pravidelně setkává s technickými problémy a neumožňuje provádět možnosti, které společnost požaduje.

Prvním krokem, který jsem učinil před psaním této bakalářské práce, byla osobní návštěva společnosti a konzultace s jejím vedením. Zde jsem se seznámil s firemními procesy a prošel si jednotlivé sekce společnosti. Návštěva byla velmi užitečná pro pozdější analýzu a návrh Webové aplikace pro objednávky a plánování výroby (WAHORZ).

Cíle práce

Cílem této bakalářské práce je zhodnotit současný stav řešení automatizovaných postupů při zakládání objednávek, plánování, zápisu a kontroly výroby. Na základě těchto poznatků bude mým úkolem navrhnout řešení nové webové aplikace a pro implementaci tohoto řešení budou zvoleny vhodné nástroje tak, aby výsledný produkt odpovídal požadavkům zákazníka. Následně implementuji serverovou a klientskou stranu WAHORZ. V závěru bude třeba tuto aplikaci otestovat.

Mým prvním cílem bylo aplikaci navrhnout takovým způsobem, aby ji uživatelé používali a vyvarovat se příčinám nechtěného používání jako v případě současných informačních systémů zavedených ve společnosti. Bude proto třeba vytvořit takový produkt, který uživatele přivede k jeho používání. Proto bude velmi důležité celý systém konzultovat se zaměstnanci společnosti HORZ s.r.o. Zaměstnanci se dělí do tří skupin – vedení společnosti, vedení výroby a operátoři výroby. Každá z těchto tří skupin bude využívat tento systém trochu odlišným způsobem a bude důležité každému systém přizpůsobit.

Současné řešení

Pro ukládání informací o objednávkách, registru firemní infrastruktury, vyráběných produktech, materiálech, zákaznících, dodavatelích a zaměstnancích používá společnost informační systém SPF2013, který velmi úzce spolupracuje s balíkem kancelářských programů MSO2010. Dále používá účetní program POHODA, kterým se v této bakalářské práci nebudu zabývat, ale vzhledem k možným budoucím rozšířením WAHORZ je o tomto programu třeba uvažovat.

2.1 Přístupnost

K systému SPF2013 přistupuje uživatel pomocí webového prohlížeče. Společnost Microsoft doporučuje pro zachování co největší kompatibility používat prohlížeč Internet Explorer. Ten totiž pro práci s tímto systémem podporuje více funkcí[4]. Dále mezi podporované webové prohlížeče patří Mozilla Firefox, Google Chrome či Apple Safari[4].

MSO2010 je určen pro počítače s operačním systémem Microsoft Windows XP SP 3 a novější verze[5]. Společnost využívá MSO2010 k vytváření důležitých sdílených firemních dokumentů odkazujících se na informace, které jsou uloženy v SPF2013. Například pomocí programu MSO2010 Excel¹ je v současné době řešen systém objednávek a plánování výroby – do tabulkového procesoru jsou zapsány údaje o objednavce, ale informace o objednaném produktu je získána z databáze SPF2013.

2.2 Microsoft SharePoint

Microsoft SharePoint je webový informační systém společnosti Microsoft, který je určen pro ukládání a sdílení informací. Nabízí velké množství funkcí, mezi které patří správa obsahu, správa dokumentů, osobní úložiště dat, podnikové

¹Tabulkový procesor.

vyhledávání či podnikovou sociální síť. Dále potom správu pracovních postupů a podporu Business Intelligence (BI)[6].

Tento systém je možné poskytovat dvěma způsoby. První variantou je použití Microsoft SharePoint Online, druhou pak Microsoft SharePoint Server.

2.2.1 Microsoft SharePoint Online

Microsoft SharePoint Online je cloudová verze spravovaná společností Microsoft.

Největší výhodou této varianty je možnost absence jakékoliv serverové infrastruktury. Veškeré záležitosti týkající se fungování Microsoft SharePoint totiž zajišťuje společnost Microsoft. Další výhodou je pak poskytnutí aktuální verze Microsoft Office[7]. Tento systém však podléhá pravidelnému měsíčnímu licencování za jednoho uživatele[8].

2.2.2 Microsoft SharePoint Server

Varianta Microsoft SharePoint Server vyžaduje instalaci na server a dále splnění hardwarových a softwarových požadavků dle dané verze. Jedná se o požadavky na minimální verzi nainstalovaného operačního systému a Microsoft SQL Serveru, další požadavky se týkají velikosti paměti RAM či procesoru[9]. Každý přistupující uživatel musí disponovat licencí CAL²[8].

Od verze SharePoint 2010 byl tento produkt výrazně modernizován a vylepšen. Vzniklo spoustu nových funkcí, které umožňují lépe využít tento informační systém. Jednou z nich je třeba podpora Office Web Apps³[10].

2.3 Nevýhody současného řešení

Systém SPF2013 sice poskytuje zákazníkovi širokou škálu možností, přesto jsou některé jeho požadavky téměř nedosažitelné.

Tabulkový procesor MSO2010 Excel je mimo jiné využíván pro generování různých dokumentů ve formátu PDF. Jedním z těchto dokumentů je průvodka výrobou – dokument, který pro daného zaměstnance a danou pracovní směnu určuje jaké výrobky, v jakém množství a na jakém stroji se mají vyrobit. Zaměstnanec pak do takto vytištěného dokumentu zapisuje výsledky výroby. Tento dokument dále slouží pro zjištění důvodů případných negativních událostí, mezi které patří vysoká zmetkovitost⁴, reklamace a podobně.

Generování těchto průvodků je řešeno pomocí maker⁵ implementovaných v programovacím jazyce Visual Basic. Generování ale provázají časté problémy, které vznikají kvůli synchronizaci dokumentu na SPF2013, úpravou

²Client access licence, způsob licencování společnosti Microsoft.

³Doplňek, který podporuje otevírání Microsoft Office dokumentů webovém prohlížeči.

⁴Počet nepovedených výrobků.



⁵V kancelářských balících posloupnost funkcí usnadňující nějakou činnost.

2.3. Nevýhody současného řešení

dokumentu na jiném počítači, než na kterém byl vytvořen, nebo pokud zaměstnanci zapisují údaje ve špatném tvaru. Tento problém úplně znemožňuje vytvářet tyto dokumenty automatizovaně a společnost tak raději volí metodu ručně psaných průvodků výrobou.

Průvodku připravenou v aplikaci MSO2010 Excel můžete vidět na obrázku 2.1.

Uživatelé mezi další problémy zařadili mnohačetné požadavky na zadávání přihlašovacích údajů při otevírání dokumentů v MSO2010 a někdy není možné z neznámého důvodu dokumenty ukládat do systému SPF2013.

Horz TJ Stadion Teplice s.r.o.		Směnová průvodka		Operátor výroby		0		
Výrobek 1		Šarže	15_009	Výkres	---			NASTA
Control Plan		CP_univ	Výrobek	kryt 003 bílá	Stroj	CS 1025		1/
Uvolňovací kus		Počet kusů	Šarže materiálu					NASTA
OD	DO	Zmetky najžděcí	Zmetky operační	Prostoje psát do poznámky a označit.				
Výrobek 2		Šarže	Výkres					
Control Plan		Výrobek	Stroj					
Uvolňovací kus		Počet kusů	Šarže materiálu					NASTA
OD	DO	Zmetky	Zmetky operační	Prostoje psát do poznámky a označit.				
Výrobek 3		Šarže	Výkres					
Control Plan		Výrobek	Stroj					GENEROVANÉ

Obrázek 2.1: Obrazovka s dokumentem pro export průvodků výrobou.

Analýza

Během této části práce bylo velmi důležité rozeznat procesy ve společnosti, akceptovat a strukturovat požadavky na vytvoření WAHORZ, rozlišit různé role zaměstnanců v rámci společnosti a seznámit se s vizí fungování nového informačního systému.

3.1 Procesy ve společnosti

V této kapitole rozeberu dva základní procesy společnosti, kterých se bude přímo týkat funkčnost celé webové aplikace.

3.1.1 Proces zadávání objednávek

Objednávky přijímá společnost dvěma způsoby. První možností je, že objednávku obdrží vedení společnosti, které ji předává vedení výroby ke zpracování. Tímto způsobem se především obstarávají objednávky od nových zákazníků či od stávajících zákazníků, pokud se jedná o nové produkty či požadavky na změnu výroby produktů.

Druhým způsobem je, že objednávku obdrží vedení výroby, které ji zařadí do výrobního procesu. Jedná se především o periodicky zadávané zakázky.

Schéma tohoto procesu si můžete prohlédnout na obrázku 3.1.

3.1.2 Proces zpracování objednávek

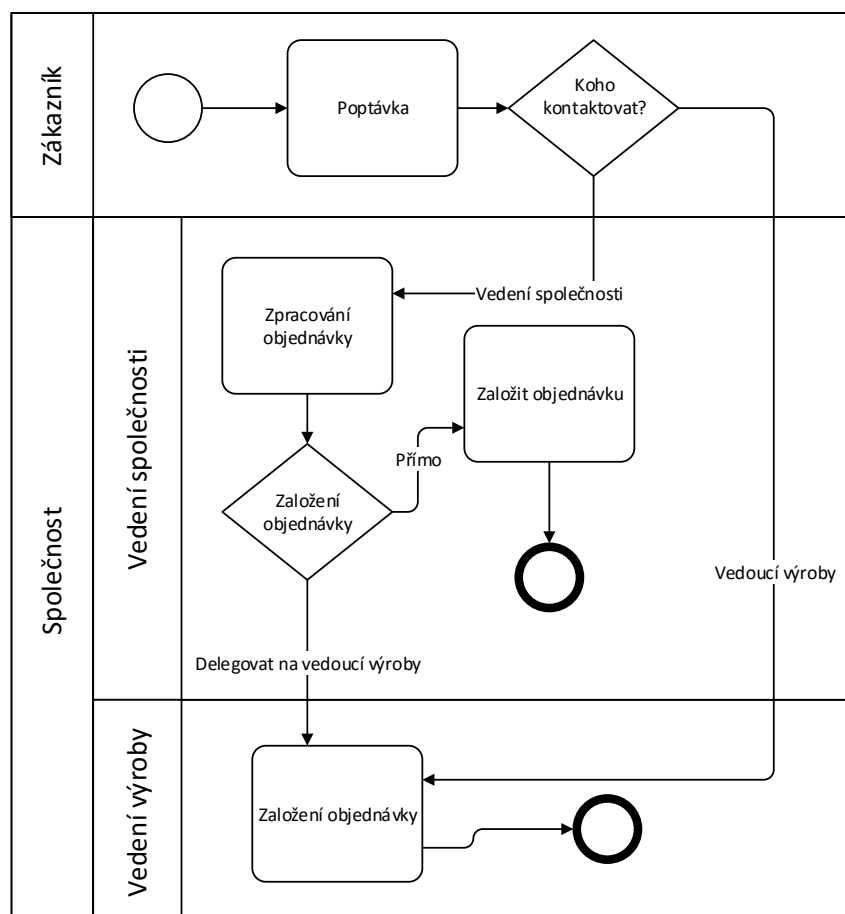
Objednávku zařazuje do výroby vedení výroby a určuje přitom datum provedení výroby a zaměstnance, který ji bude provádět.

Zaměstnanec si v den výroby vyzvedne průvodku výrobou. V této průvodce se nachází informace o úkonech, které jsou daný den plánované. Průvodka zároveň obsahuje formulář k vyplnění výsledku výroby. Po splnění počtu vyrobených kusů produktu jsou výsledné výrobky nejdříve zkontrolovány a po doplnění případných nevyhovujících kusů je objednávka uzavřena a expedována

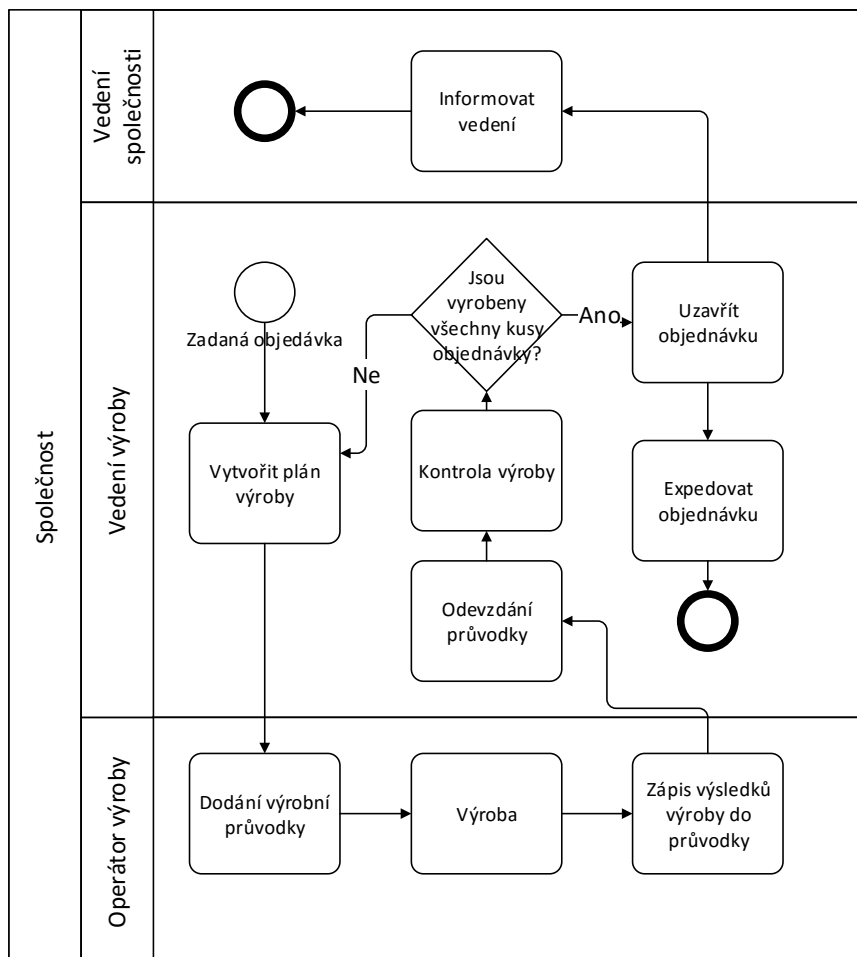
3. ANALÝZA

zákazníkovi. Podobu tohoto dokumentu před jeho vytištěním je k nahlédnutí na obrázku 2.1.

Schéma tohoto procesu si můžete prohlédnout na obrázku 3.2.



Obrázek 3.1: Diagram procesu přijetí objednávky.



Obrázek 3.2: Diagram procesu zpracování objednávky.

3.2 Funkční požadavky

3.2.1 Funkční požadavky pro uživatele

V této kapitole uvedu souhrn funkčních požadavků společnosti HORZ s.r.o. na vytvoření WAHORZ.

F1.1 Registrace

Nepřihlášený uživatel bude mít možnost jednoduché registrace. V registračním formuláři budou požadovány následující údaje: emailová adresa

3. ANALÝZA

uživatele a heslo. Emailová adresa bude zároveň sloužit jako přihlašovací, tudíž musí být jedinečná.

F1.2 **Zapomenuté heslo**

Aplikace umožní obnovit zapomenuté heslo pomocí formuláře, ve kterém bude zadána emailová adresa uživatele. Následně bude na daný email odeslána zpráva s odkazem sloužící pro změnu hesla.

F1.3 **Přihlášení uživatele**

Uživatel se bude přihlašovat zadáním registračního emailu a zvoleného hesla.

F1.4 **Odhlášení uživatele**

Aplikace umožní uživateli odhlášení z webové aplikace pomocí příslušného odkazu, který bude k dispozici na všech stránkách webové aplikace.

F1.5 **Změna údajů**

Aplikace umožní uživateli změnit své heslo pomocí příslušného formuláře, ve kterém zadá současné heslo, nové heslo a potvrzení nového hesla.

F1.6 **Schválit uživatele**

Aplikace umožní schválit uživatele žádajícího o přístup k portálu.

F1.7 **Odmítnout žádost uživatele**

Aplikace umožní odmítnout žádost uživatele žádajícího o přístup k portálu.

F1.8 **Odstranit uživatele**

Aplikace umožní odstranit uživatele.

F1.9 **Změnit uživatelskou roli**

Aplikace umožní uživateli změnit jeho uživatelskou roli.

F1.10 **Obnovit heslo uživatele**

Aplikace umožní nastavit nové heslo pro daného uživatele.

3.2.2 Funkční požadavky kontraktorů

F2.1 **Správa zákazníků**

F2.1.1 **Přidat zákazníka**

Aplikace umožní přidání nového zákazníka. Do příslušného formuláře se vyplňují základní údaje o společnosti – název, IČO, DIČ, adresa sídla společnosti a kontaktní údaje – telefon a email.

F2.1.2 Upravit zákazníka

Aplikace umožní editovat informace o zákazníkovi ve stejném rozsahu jako v požadavku F2.1.1 Přidat zákazníka.

F2.1.3 Smazat zákazníka

Aplikace umožní smazání zákazníka. Daný zákazník však nesmí mít žádné aktivní objednávky.

F2.2 Správa dodavatelů

F2.2.1 Přidat dodavatele

Aplikace umožní přidání nového dodavatele. Do příslušného formuláře se vyplňují základní údaje o společnosti: název, IČO, DIČ, adresa sídla společnosti a kontaktní údaje – telefon a email.

F2.2.2 Upravit dodavatele

Aplikace umožní editovat informace o dodavateli ve stejném rozsahu jako v požadavku F2.2.1.

F2.2.3 Smazat dodavatele

Aplikace umožní smazat dodavatele. Dodavatel však nesmí mít žádné aktivní dodávky – šarže materiálu.

3.2.3 Funkční požadavky zaměstnanců

F3.1 Přidat zaměstnance

Aplikace umožní vytvořit nového zaměstnance. Do příslušného formuláře se vyplňují informace s osobními údaji: výběr uživatele WAHORZ, jméno, příjmení, ulice, město, PSČ.

F3.2 Upravit zaměstnance

Aplikace umožní editaci informací o zaměstnanci ve stejném rozsahu jako v požadavku F3.1.

F3.3 Smazat zaměstnance

Aplikace umožní smazat zaměstnance. V tomto případě nedojde k úplnému odstranění, pokud byl zaměstnanec součástí výrobních plánů společnosti, které jsou ve WAHORZ uloženy.

3.2.4 Funkční požadavky produkce společnosti

F4.1 Správa objednávek

F4.1.1 Přidat objednávku

Aplikace umožní založení nové objednávky vyplněním příslušného formuláře s následujícími prvky: výběr zákazníka a požadovaného

produktu, počet požadovaných kusů daného produktu, datum zadání a datum požadovaného termínu ke splnění objednávky. Volitelně bude možné přidat označení šarže objednávky – jedná se o interní identifikátor objednávek ve společnosti.

F4.1.2 Upravit objednávku

Aplikace umožní upravit objednávku v plném rozsahu jako v požadavku F4.1.1. Editace objednávky však bude umožněna pouze v případě, že nebyl vytvořen žádný výrobní plán k dané objednávce.

F4.1.3 Smazat objednávku

Aplikace umožní smazat objednávku. Stejně jako u úpravy objednávky bude tento krok možné učinit pouze v případě, kdy není zadán plán výroby dané objednávky.

F4.1.4 Ukončit objednávku

Aplikace umožní označit objednávku za dokončenou. S jednotlivými aspekty této objednávky již nebude možné pracovat – jedná se například o přidávání, editaci a mazání výrobních plánů, výsledků a kontrol z dané výroby.

F4.2 Správa šarží materiálu

F4.2.1 Přidat šarži materiálu

Aplikace umožní přidat šarži materiálu. Uživatel zvolí typ materiálu, dále vyplní datum dodání a dodavatele.

F4.2.2 Upravit šarži materiálu

Aplikace umožní upravit informace o šarži materiálu v plném rozsahu jako v požadavku F4.2.1.

F4.2.3 Smazat šarži materiálu

Aplikace umožní smazat šarži materiálu. Šarži nelze smazat, pokud je obsažena ve výrobních výsledcích.

F4.3 Správa plánů výroby

F4.3.1 Přidat plán výroby

Aplikace umožní vytvoření plánu výroby z objednávky dle příslušného formuláře: datum směny, typ směny (ranní, odpolední) a operátor výroby. Volitelně pak může obsahovat požadovaný stroj a množství.

F4.3.2 Upravit plán výroby

Aplikace umožní úpravu plánu výroby v plném rozsahu jako v požadavku F4.3.1. Plán však bude moci být upraven pouze v případě, že nebude zadán výsledek výroby.

F4.3.3 **Smazat plán výroby**

Aplikace umožní smazat plán výroby v případě, že nebude zadán výsledek výroby.

F4.4 **Správa výsledků výroby**

F4.4.1 **Přidat výsledek výroby**

Aplikace umožní zapsat výsledek výroby daného zaměstnance. Vypĺňují se informace o počtu vyrobených kusů, počtu zmetků najížděcích a operačních, souhrn deformací produktu a použitých šarží materiálů. Dále je možné přidat poznámku o výrobě.

F4.4.2 **Upravit výsledek výroby**

Aplikace umožní upravit výsledek výroby daného zaměstnance za podmínky, že nebyla provedena kontrola výrobků, v plném rozsahu jako v požadavku F4.4.1.

F4.4.3 **Smazat výsledek výroby**

Aplikace umožní odstranit výsledek výroby za podmínky, že nebyla provedena kontrola výrobků.

F4.5 **Správa kontroly výsledků výroby**

F4.5.1 **Přidat výsledek kontroly výroby**

Aplikace umožní zapsat výsledek z kontroly vyrobených produktů. Zapisuje se počet kusů dané výroby, které byly a nebyly v pořádku.

F4.5.2 **Upravit výsledek kontroly výroby**

Aplikace umožní upravit výsledek kontroly výroby za podmínky, že objednávka nebude uzavřena, v rozsahu jako v požadavku F4.5.1.

F4.5.3 **Smazat výsledek kontroly výroby**

Aplikace umožní odstranit výsledek kontroly výroby za podmínky, že objednávka nebude uzavřena.

3.2.5 Funkční požadavky globální nastavení

F5.1 **SharePoint**

F5.1.1 **Synchronizace se systémem SharePoint**

Aplikace umožní synchronizovat data spravovaná v systému SPF2013. Jedná se o informace o výrobcích, materiálech a firemní infrastruktuře.

F5.1.2 **Změna přihlašovacích údajů k SPF2013**

Aplikace umožní změnit přihlašovací údaje k systému SPF2013. Zadává se uživatelské jméno, heslo a URL adresa pro připojení k portálu SPF2013.

F5.1.3 Testování připojení k systému SharePoint

Aplikace umožní otestovat správnost nastavení pro připojení k systému SharePoint.

F5.2 Oznámení

F5.2.1 Přidat oznámení

Aplikace umožní přidat oznámení, které se skládá z titulku a textu.

F5.2.2 Upravit oznámení

Aplikace umožní upravit oznámení ve stejném rozsahu jako v požadavku F5.2.1.

F5.2.3 Smazat oznámení

Aplikace umožní smazat oznámení.

3.3 Nefunkční požadavky

N1 Systém je webová aplikace

WAHORZ je spuštěna na lokálním serveru společnosti HORZ s.r.o. a přístup je umožněn pouze v intranetu společnosti pod adresou <http://ishorz/>.

N2 Hostitelský server systému

Společnost disponuje serverem s operačním systémem Windows Server 2012 R2, na kterém vyžaduje spuštění webové aplikace.

N3 Databázový server

Aplikace bude používat databázový server, který je použitelný zdarma.

N4 Podporované prohlížeče

Aplikace bude podporována v nejnovějších verzích následujících prohlížečů:

P1 Internet Explorer

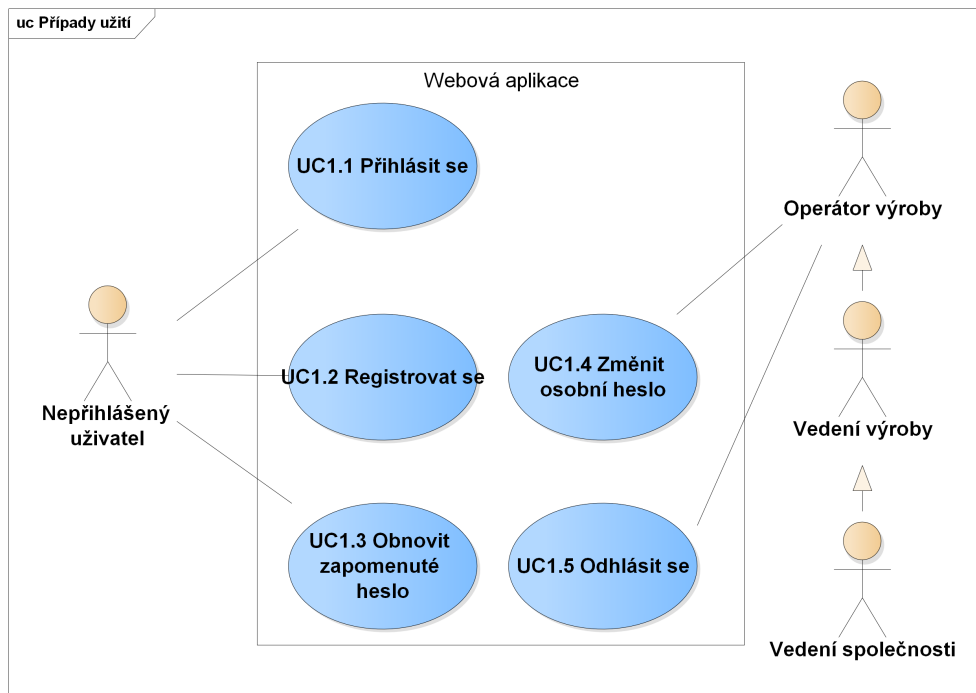
P2 Google Chrome

P3 Mozilla Firefox

P4 Safari

3.4 Případy užití

V této kapitole vypíši akce, které bude systémem umožněno provádět.



Obrázek 3.3: Diagram případů užití pro uživatele.

3.4.1 UC1 Uživatelské případy užití

Případy užití z této části můžete vidět na diagramu 3.3.

- UC1.1 Přihlásit se
- UC1.2 Registrovat se
- UC1.3 Obnovit zapomenuté heslo
- UC1.4 Změnit heslo
- UC1.5 Odhlásit se

3.4.2 UC2 Případy užití práce s kontakty

Případy užití v této části má právo vykonávat pouze vedení společnosti.

- UC2.1 Přidat dodavatele
- UC2.2 Upravit dodavatele
- UC2.3 Smazat dodavatele

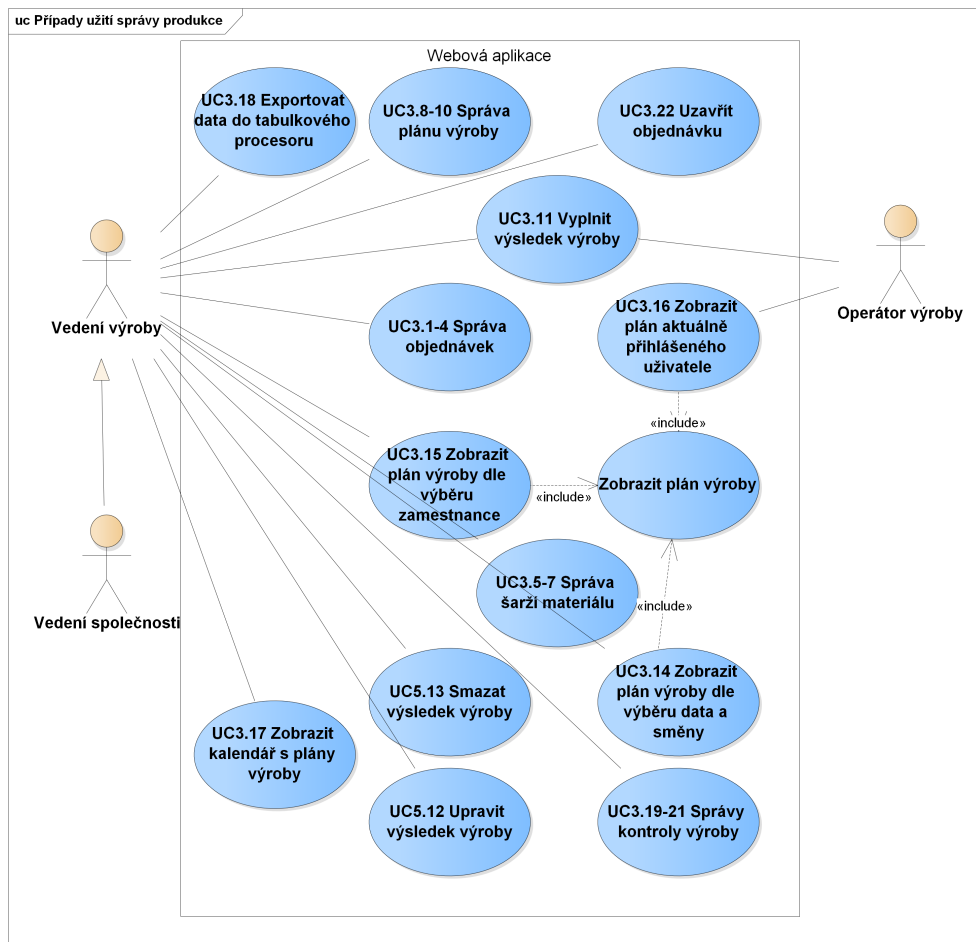
3. ANALÝZA

UC2.4	Zobrazit seznam dodavatelů
UC2.5	Přidat zákazníka
UC2.6	Upravit zákazníka
UC2.7	Odstranit zákazníka
UC2.8	Zobrazit seznam zákazníků
UC2.9	Přidat zaměstnance
UC2.10	Upravit zaměstnance
UC2.11	Smazat zaměstnance
UC2.12	Přiřadit uživatele k zaměstnanci

3.4.3 UC3 Případy užití správy produkce

Tyto případy popisuje diagram, který můžete vidět na obrázku 3.4.

UC3.1	Zobrazit seznam objednávek
UC3.2	Zadat novou objednávku
UC3.3	Upravit objednávku
UC3.4	Smazat objednávku
UC3.5	Přidat šarži materiálu
UC3.6	Upravit šarži materiálu
UC3.7	Smazat šarži materiálu
UC3.8	Vytvořit plán výroby
UC3.9	Upravit plán výroby
UC3.10	Smazat plán výroby
UC3.11	Vyplnit výsledek výroby jakéhokoliv uživatele
UC3.12	Upravit výsledek výroby
UC3.13	Smazat výsledek výroby
UC3.14	Zobrazit plán výroby dle výběru data a směny
UC3.15	Zobrazit plán výroby dle výběru zaměstnance



Obrázek 3.4: Diagram případů užití pro správu produkce.

- UC3.16 Zobrazit plán aktuálně přihlášeného uživatele
- UC3.17 Zobrazit kalendář s plány výroby
- UC3.18 Exportovat data do tabulkového procesoru
- UC3.19 Vyplnit kontrolu výroby
- UC3.20 Upravit kontrolu výroby
- UC3.21 Smazat kontrolu výroby
- UC3.22 Uzavřít objednávku

3.4.4 UC4 Případy užití globálního natavení webové aplikace

Případy užití v této části má právo vykonávat pouze vedení společnosti.

- UC4.1 Schválit uživatele
- UC4.2 Zamítnout žádost o schválení uživatele
- UC4.2 Změnit roli uživatele
- UC4.3 Změnit heslo uživatele po požádání
- UC4.4 Odstranit schváleného uživatele
- UC4.5 Nastavit přihlašovací údaje pro SPF2013
- UC4.6 Testovat připojení k SPF2013
- UC4.7 Synchronizovat data z SPF2013
- UC4.8 Přidat oznámení
- UC4.9 Upravit oznámení
- UC4.10 Odstranit oznámení

3.5 Uživatelské role

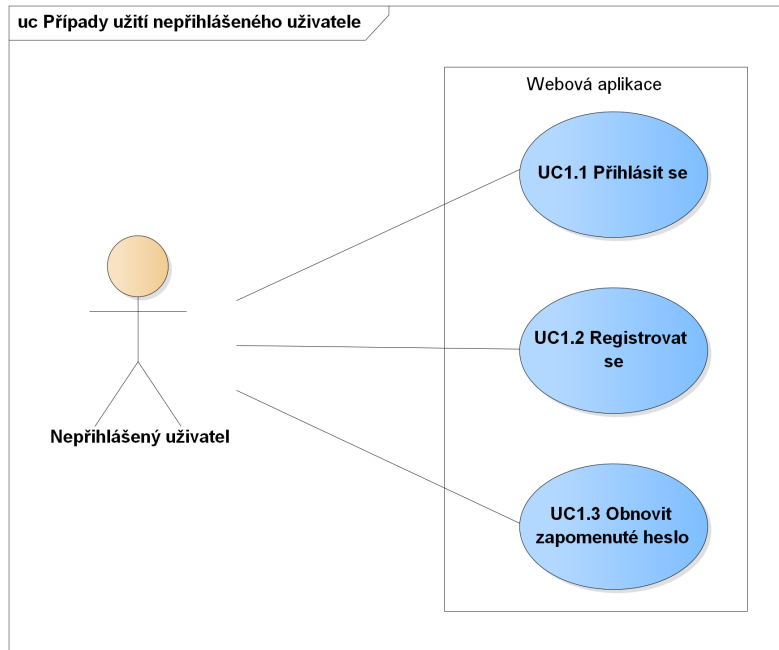
WAHORZ bude sloužit pro tři typy autorizovaných uživatelů. Každá z těchto rolí má specifické úkoly, ke kterým bude systém navržen.

3.5.1 Neautorizovaný uživatel

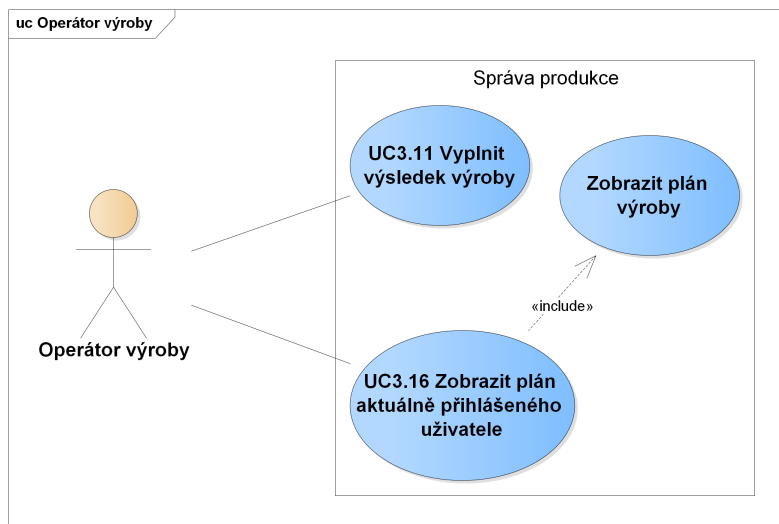
Neautorizovaný uživatel je role uživatele, který je nepřihlášený, stejně tak se jedná o přihlášeného uživatele, který nebyl schválen vedením společnosti. Tento uživatel má přístup pouze k domovské stránce, registraci a přihlášení. Diagram případu užití si můžete prohlédnout na obrázku 3.5.

3.5.2 Operátor výroby

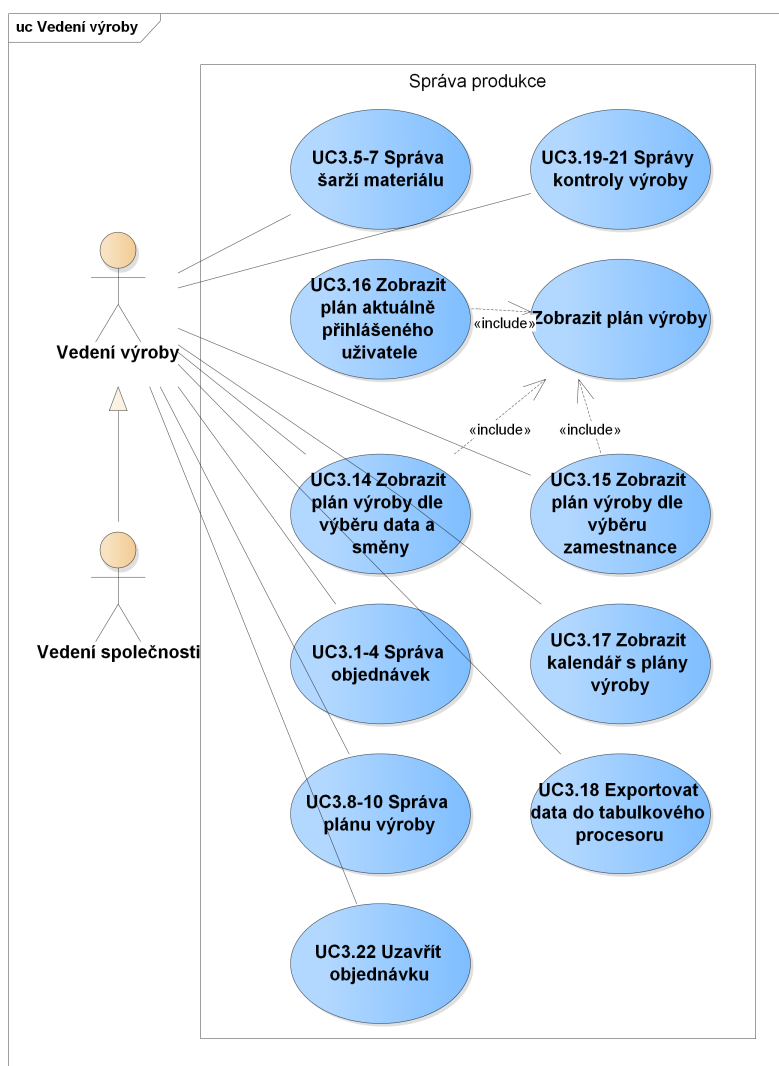
Operátor výroby je role uživatele, který spravuje pouze své výrobní plány. Zobrazuje svůj výrobní plán a má oprávnění vyplnit. Vůči WAHORZ nemá tento typ uživatele žádná další oprávnění. Diagram případu užití si můžete prohlédnout na obrázku 3.6.



Obrázek 3.5: Diagram případu užití pro nepřihlášeného uživatele.



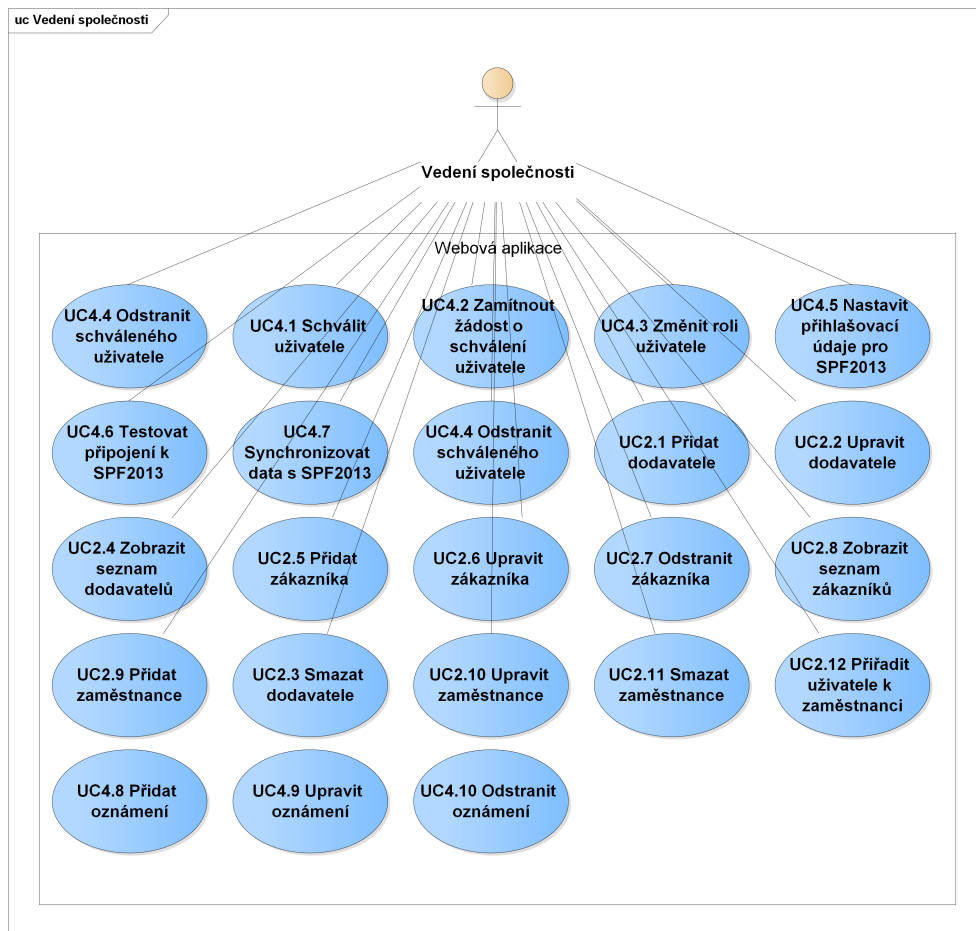
Obrázek 3.6: Diagram případu užití pro operátora výroby.



Obrázek 3.7: Diagram případů užití pro vedení výroby.

3.5.3 Vedení výroby

Vedení výroby je role uživatele, který má, co se týče správy produkce společnosti, největší oprávnění. Kompletně spravuje objednávky, řídí plánování výroby, zadává, upravuje a maže výsledky výroby jakéhokoliv zaměstnance. Role *vedení výroby* je nadřazena roli *operátor výroby*. Diagram případu užití si můžete prohlédnout na obrázku 3.7.



Obrázek 3.8: Diagram případu užití pro vedení společnosti.

3.5.4 Vedení společnosti

Vedení společnosti je uživatelská role, která nabízí nejširší možné využití portálu, které zahrnuje, stejně jako u role *vedení výroby*, kompletní správu produkce společnosti. Dále spravuje seznamy zákazníků, dodavatelů a zaměstnanců, a může měnit globální nastavení serveru, mezi které patří především synchronizace WAHORZ se systémem SPF2013. Role *vedení společnosti* je nadřazena roli *vedení výroby*. Diagram případů užití týkajících se produkce výroby – sekce, kde mají uživatelské role *vedení společnosti* a *vedení výroby* stejné oprávnění, si můžete prohlédnout na obrázku 3.7. Další diagram případů užití, který se vztahuje pouze pro uživatelskou roli *vedení společnosti*, můžete vidět na obrázku 3.8.

3.6 Doménová analýza

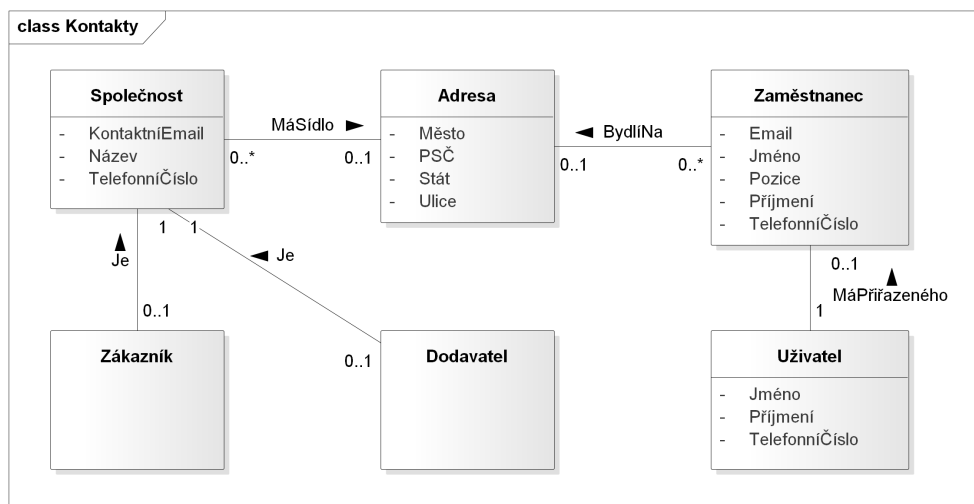
Doménová analýza byla provedena pro hlubší zjištění a pochopení vztahů jednotlivých procesů ve společnosti. Analyzoval jsem vždy jednotlivé logické sekce, které vychází z nefunkčních požadavků.

3.6.1 Kontakty

Sekce kontakty obsahuje entity, které jsou spojeny s informacemi o zaměstnancích, zákaznících a dodavatelích společnosti:

- **Uživatel.** Obsahuje informace o uživateli webové aplikace, který se bude přihlašovat k webové aplikaci.
- **Zaměstnanec.** Eviduje informace o zaměstnanci, který je přiřazen k uživateli webové aplikace.
- **Zákazník.** Nese informace o zákazníkovi společnosti.
- **Dodavatel.** Nese informace o dodavateli společnosti.

Diagram těchto entit je graficky popsán na obrázku 3.9

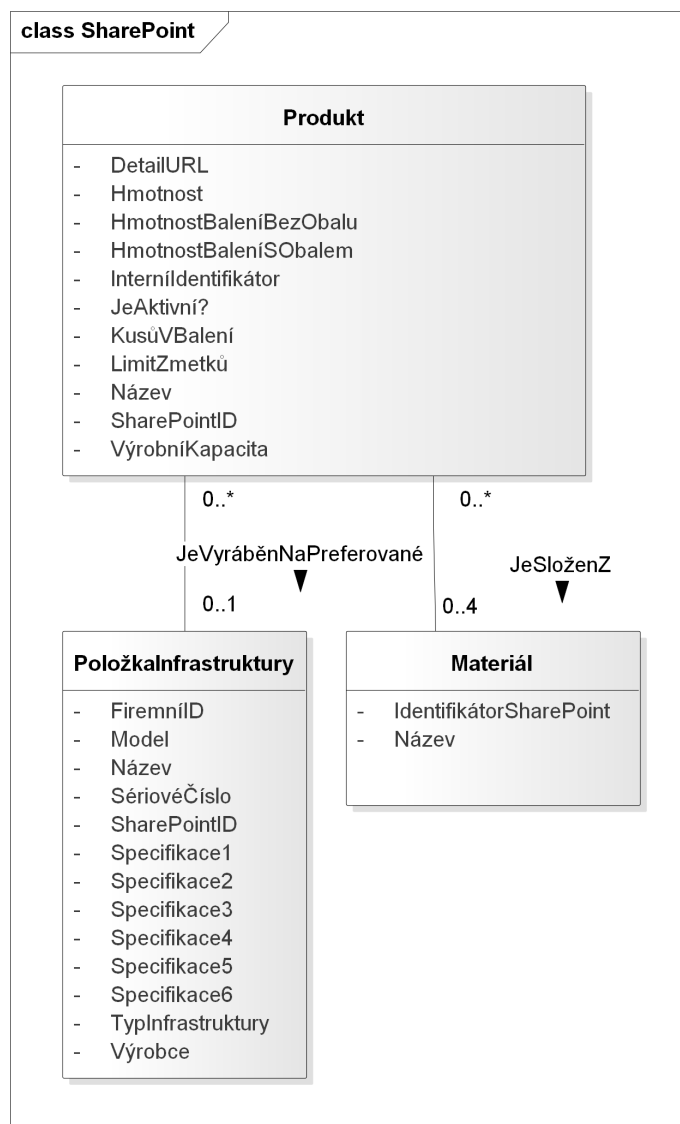


Obrázek 3.9: Diagram entit sekce Kontakty.

3.6.2 Synchronizace SharePoint

Tato sekce obsahuje entity, které obsahují informace z portálu SPF2013. V našem případě se jedná o následující entity:

- **Produkt.** Nese informace o vyráběných produktech.

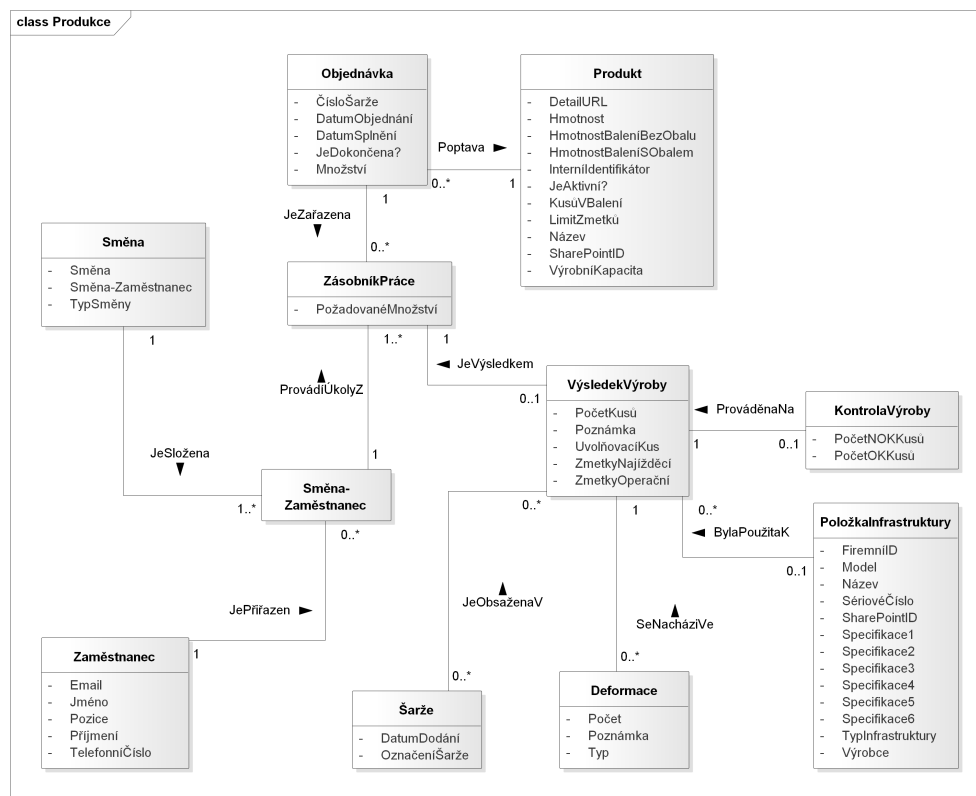


Obrázek 3.10: Diagram entit sekce SharePoint.

- **PoložkaInfrastruktury**. Nese informace o infrastruktuře podniku. V této aplikaci se bude jednat o výrobní stroje.
- **Materiál**. Nese informace o materiálech, které se používají k výrobě daného produktu.

Diagram těchto entit je k nahlédnutí na obrázku 3.10.

3. ANALÝZA



Obrázek 3.11: Diagram entit sekce Evidence produkce.

3.6.3 Evidence produkce

V této sekci se nachází entity, kterých se přímo týká plánování výroby společnosti. Tyto entity kvůli přesnému pochopení popíší detailněji a jejich diagram si můžete prohlédnout na obrázku 3.11.

3.6.3.1 Objednávka

Entita Objednávka eviduje informace o objednávce. Po dokončení produkce požadovaného množství a následného předání objednávky je označena jako dokončená.

3.6.3.2 Směna

Entita Směna obsahuje záznam o směně. Nese informace o datu a typu směny. Obsahuje seznam s odkazy na přiřazení zaměstnance a položek, které mají být v dané směně produkovány.

3.6.3.3 Směna-Zaměstnanec

Entita Směna-Zaměstnanec obsahuje záznam o přiřazení zaměstnance ke směně a jeho výrobní plán pro tuto směnu.

3.6.3.4 ZásobníkPráce

Entita ZásobníkPráce eviduje zásobník práce zaměstnance pro danou směnu, která je dána odkazem na entitu *Směna-Zaměstnanec*.

3.6.3.5 Šarže

Entita Šarže obsahuje informace o dodaných šaržích materiálu, které se používají a spotřebovávají při výrobě. Entita *Šarže* nese odkaz na entitu *Dodavatel*. A seznam odkazů na položky typu *Materiál*.

3.6.3.6 Deformace

Entita Deformace obsahuje informaci o deformaci produktu.

3.6.3.7 VýsledekVýroby

Entita VýsledekVýroby obsahuje informaci o výsledku výroby – počtu vyrobených kusů a zmetkovitosti během plnění výrobního plánu. Tato entita dále obsahuje odkaz na entitu *ZásobníkPráce* a na seznam použitých šarží (entita *Šarže*) a vzniklých deformací (entita *Deformace*).

3.6.3.8 KontrolaVýroby

Entita KontrolaVýroby obsahuje informace o kontrole vyrobených produktů. Odkazuje se na entitu *VýsledekVýroby*.

3.7 Ostatní entity

Tato sekce obsahuje následující entity:

- **NastaveníSharePoint.** Jedná se o entitu, která obsahuje přihlašovací údaje a adresu URL pro připojení k portálu SPF2013.
- **Notifikace.** Jedná se o entitu, která slouží k ukládání notifikací vytvořených ve WAHORZ.

Návrh

V této kapitole se budeme věnovat nejdříve výběru vhodných technologií a následně návrhu databázového schématu. Nakonec popíšeme, jakým způsobem vznikl návrh uživatelského rozhraní této aplikace.

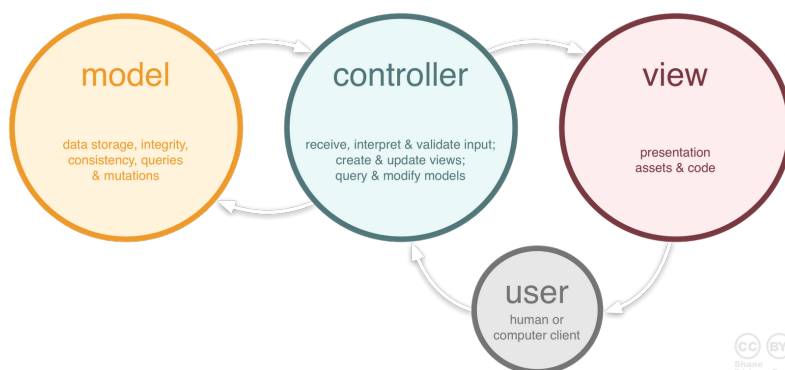
4.1 Použité technologie

Požadavkem zákazníka bylo zprovoznění webové aplikace na serveru s operačním systémem Microsoft Windows Server 2012 Standard, kterým disponuje. Z důvodu kompatibilního přístupu k portálu SPF2013, který umožňuje komunikaci prostřednictvím rozhraní Microsoft SharePoint Services Web Service, ale i z důvodu osobních preferencí bylo rozhodnuto vyvíjet WAHORZ pod platformou .NET programovacím jazykem C# v aplikačním frameworku ASP.NET MVC 6. Pohledová část systému byla vytvořena ve značkovacím jazyku (markup language) ASP.NET Razor. Část aplikace je implementována v JavaScriptu s použitím knihovny React.

4.1.1 Architektura Model-view-controller

Vybraný framework je založen na architektuře MVC – Model-View-Controller. Jedná se o jednu z nejrozšířenějších architektur, která spočívá v oddělení logické části od výstupní[11]. Řeší tedy problémy, kdy se ve stejném úseku kódu nachází logická část aplikace a zároveň její vykreslování uživateli – tento úsek pak zahrnuje databázové dotazy, logiku aplikace a pohledové části, například ve formě HTML kódu, které jsou do sebe zapleteny.

Díky oddělení jednotlivých celků kódu zjednodušuje tato architektura orientaci ve zdrojovém kódu během implementace a případných úpravách aplikace. Nejvíce pak především při týmové práci, kdy pracuje na stejném projektu v jednu chvíli více osob. Architektura je založena na rozdělení do tří nezávislých komponent: modelů, pohledů a kontrolerů – a to takovým způsobem, že



Obrázek 4.1: Popis architektury MVC[12].

modifikace jednotlivých komponent co nejméně ovlivňuje ty ostatní[11]. Následuje popis jednotlivých komponent:

- **Modely** zajišťují reprezentaci dat, tedy především databázové dotazy a bussiness logiku aplikace – například ověření správnosti vstupů. K modelům zásadní nepatří informace závislé na kontrolerech či pohledech[11].
- **Pohledy** obstarávají vykreslení uživatelského rozhraní, například ve formátu HTML[11].
- **Kontrolery** řídí aplikační logiku a tok událostí v aplikaci[11].

Na obrázku 4.1 si můžeme povšimnout, jak spolu jednotlivé komponenty pracují a jakým způsobem reaguje s aplikací postavenou na této architektuře uživatel, který zasílá požadavky na *kontrolery* a jako odpověď dostává *pohled*.

4.1.2 ASP.NET MVC 6

.NET Framework je softwarový framework vyvíjený společností Microsoft[13] určený pro operační systém Microsoft Windows. Tento framework je založen na základě běhového prostředí (run-time system) Common Language Runtime (CLR), který realizuje základní infrastrukturu a umožňuje tak spouštění .NET aplikací. CLR obsahuje základní knihovny pro přístup k datům a pro práci s uživatelským rozhraním[14].

ASP.NET je open source⁶ technologie a součást .NET Framework pro tvorbu webových aplikací. Součástí této technologie je i webový aplikační framework ASP.NET MVC, který implementuje architektonický vzor *MVC*.

Pro vytvoření webové aplikace jsem se rozhodl využít nejnovější verze MVC 6 v ASP.NET 5. V předchozích verzích tohoto frameworku existuje rozdělení do tří skupin použití s jinými cíli:

⁶Software s otevřeným zdrojovým kódem.

- **MVC** slouží pouze pro aplikace s architekturou *MVC*. Návratovou hodnotou *kontrolerů* je zobrazení *pohledu*.
- **Web Pages** slouží pro vytváření jednoduchých dynamických aplikací v jedné vrstvě.
- **Web API** slouží pro implementaci API rozhraní. Návratovou hodnotou *kontrolerů* jsou naformátovaná textová data.

V nové verzi tohoto frameworku došlo ke sjednocení výše uvedených částí. Tato změna spočívá ve sjednocení rodičovské třídy všech *kontrolerů* – dříve se pro *Web API* používala rodičovská třída typu `System.Web.Http.ApiController`, pro *MVC* pak `System.Web.Mvc.Controller`. V nové verzi je rodičovskou třídou pro všechny tyto skupiny třída `Microsoft.AspNet.Mvc.Controller`.

4.1.3 Microsoft DI

Vkládání závislostí, jinak také Dependency Injection (DI), je návrhový vzor pro dosažení volného spojení mezi objekty a na nich závislými objekty[15] a pro dosažení jednoduše spravovatelného kódu[16]. Novinkou v ASP.NET 5 je balíček s názvem Microsoft DI, díky kterému je možné implementovat tento mechanismus[17].

Aby nevznikaly reference na specifické implementace, je třeba nejdříve definovat rozhraní, které jsou následně předávány v konstruktoru třídy, ve které chceme závislost používat[15]. O následné správné vkládání implementací těchto rozhraní a celkovou konfiguraci *DI* se stará tzv. kontejner[15].

V ASP.NET 5 je *kontejner* popsán rozhraním `IServiceProvider`[15]. Při konfiguraci závislosti pomocí tohoto *kontejneru* se nejdříve popíše typ závislosti, dále název rozhraní a třída s implementací rozhraní.

Existují čtyři typy závislostí, rozlišené podle délky jejich životnosti:

- **Scoped** jsou služby závislostí vytvořené při každém požadavku[15].
- **Transient** jsou služby závislostí vytvářené při každém jejich vyžádání[15].
- **Singleton** jsou zavolány pouze jednou při prvním požadavku a dále je volána téže instance dané služby po celou dobu spuštění aplikace[15].
- **Instant** je velmi podobná závislosti typu *Singleton* s rozdílem volání již při spuštění aplikace[15].

Tento balíček poskytovaný společností Microsoft je určený spíše pro menší projekty. Pro ty větší je doporučeno použití produktů jako jsou Autofac či Unity.

4.1.4 Microsoft SQL Server

Při výběru databáze jsem měl na výběr z několika typů databázových serverů díky podpoře Open Database Connectivity⁷ (ODBC).

Po prozkoumání možných variant a kvůli osobním preferencím byl vybrán databázový systém Microsoft SQL Server. Jedná se o relační databázový a analytický systém, který vyvíjí společnost Microsoft. V této práci byla použita verze Microsoft SQL Server 2014 Express. Jedná se o variantu, která je dostupná zdarma všem vývojářům. Obsahuje však některé limitace – jedná se například o využití maximálně jednoho procesoru, použití maximálně 1GB paměti RAM a maximálně velikost jedné databáze je omezena na velikost 10GB[19].

WAHORZ by tímto omezením neměla být ovlivněna. Omezení se týkají především velkých aplikací s náročnými dotazy, například Microsoft SharePoint[20].

4.1.5 Windows SharePoint Services Web Service

Služba Windows SharePoint Services Web Service poskytuje sadu metod, díky kterým je možné pracovat se systémem Microsoft SharePoint – získávat a měnit data či spravovat jednotlivé sekce tohoto systému. Informace jsou mezi klientskou a serverovou stranou vyměňovány pomocí protokolu Simple Object Access Protocol (SOAP), který umožňuje především přes HTTP protokol vyměňovat informace ve formátu XML. V .NET Frameworku je pro tuto funkcionalitu vytvořeno rozhraní, díky němuž můžeme k systému Microsoft SharePoint přistupovat bez nutnosti implementace vlastní metody pro připojení a příjem dat pomocí protokolu *SOAP*[21].

4.1.6 Pohledová část

Frontend systému je viditelná část webu založena na *pohledech* architektury *MVC*. Základ *pohledu* webové aplikace je psán jazykem HTML v šablonovacím systému ASP.NET Razor, který je určen pro vytváření dynamických pohledů. Část *frontendu* WAHORZ pro správu firemní produkce je implementována pomocí javascriptové knihovny React, která získává data ze serveru skrze API rozhraní.

4.1.6.1 ASP.NET Razor

ASP.NET Razor je značkovací jazyk, syntaxí velmi podobný jazyku C#, díky kterému je v technologii ASP.NET možné vkládat dynamický obsah do webových stránek. Při načítání jednotlivých stránek spustí server překlad tohoto kódu a vygeneruje výstupní HTML kód, který předá uživateli[22].

⁷API rozhraním pro sjednocený přístup k databázovým systémům nezávislý na programovacím jazyku[18].

Zdrojový kód 4.1: Příklad kódu psaného v šablonovacím systému ASP.NET Razor

```
@model ISHORZ.ViewModels.Contacts.CustomerViewModel
@{
    ViewData["Title"] = "Create Customer";
}
<h2>Create Customer</h2>
<form asp-action="Create">
    <div class="form-horizontal">
        <hr />
        <div asp-validation-summary=
            "ValidationSummary.ModelOnly"
            class="text-danger">
        </div>
        <div class="form-group">
            <label asp-for="CompanyName"
                class="col-md-2 control-label">
            </label>
            <div class="col-md-10">
                <input asp-for="CompanyName"
                    class="form-control" />
            </div>
        </div>
    </div>
</form>
```

Ukázka zdrojového kódu 4.1 znázorňuje použití ASP.NET Razor při práci s daty získanými z *kontroleru*.

4.1.6.2 React

React je *open source* framework implementovaný v JavaScriptu, který slouží pro tvorbu uživatelského rozhraní. Tento framework vznikl v roce 2013 a je vyvíjen společností Facebook[23].

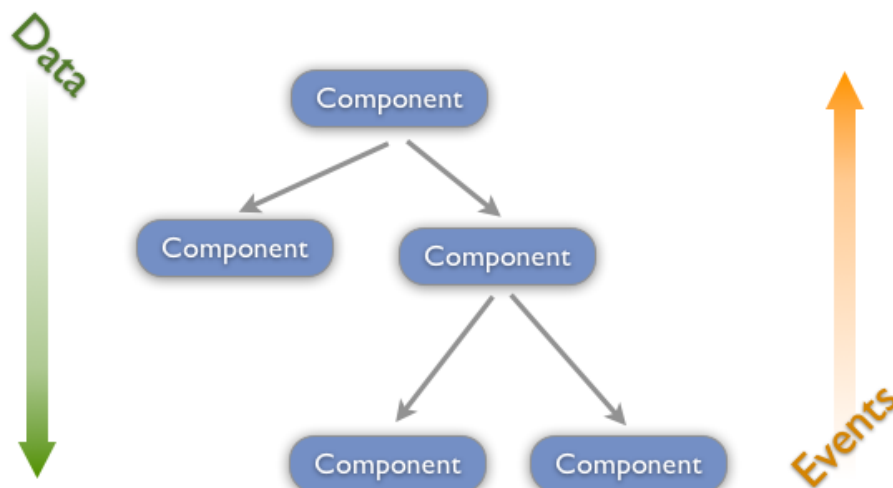
React v podstatě vykonává funkci *pohledové* části architektury *MVC*[24]. Tuto knihovnu jsem používal pouze v jedné části WAHORZ – ve správě produkce. Tato část je určena pro všechny typy uživatelů a můžeme na ní demonstrovat použití nové verze ASP.NET MVC 6 – především pak spojení technologií *MVC* a *WebAPI*.

Pro programování v Reactu je typickou vlastností vytváření opakovaně použitelných komponent. Komponenty jsou třídy, které vykonávají specifické

4. NÁVRH

akce a vrací HTML kód. Pro přístup k datům využívají dva základní mechanismy: props a state. Mechanismem props můžeme získat informace, které byly přiřazeny z nadřazené komponenty. Přistupuje se k nim pomocí proměnné `this.props`. Mechanismus state umožňuje ukládat data do třídní proměnné, která je přístupná pouze uvnitř dané komponenty a přistupuje se k ní pomocí proměnné `this.state`[25].

Každá *komponenta* musí implementovat metodu `render`, ve které se specifikuje podoba vráceného HTML kódu. Zde můžeme používat jak HTML kód, tak i vnořené *komponenty*, které se zapisují jako DOM elementy. Tagy těchto elementů jsou stejné jako názvy vnořovaných komponent. Elementy mohou dále obsahovat atributy, které jsou přístupné pomocí mechanismu *props* uvnitř vnořené *komponenty*[25]. Návrátovou hodnotou metody `render` je HTML kód, který vznikne zpracováním vnořných komponent.



Obrázek 4.2: Vnořování komponent v knihovně React[26].

Takto napsaná aplikace se následně zobrazí v prohlížeči pomocí příkazu `ReactDOM.render`, ve kterém se zadávají dva parametry: specifikace umístění aplikace v HTML kódu, nejčastěji pomocí atributu `id` HTML elementu, a dále se zapíše odkaz na hlavní, kořenovou *komponentu*[25].

Sestavení aplikace implementované v Reactu si lze představit jako stromovou strukturu popsanou na obrázku 4.2. Kořen stromu je hlavní *komponenta* aplikace a jednotlivé listy jsou *komponenty* vnořené do rodičovských komponent. V levé části obrázku je dále znázorněno předávání dat od kořene směrem k jednotlivým listům, tedy *komponentám*, které tyto data zpracovávají.

V pravé části pak volání událostí *komponent* směrem ke kořeni, přičemž se očekává, že některý předek *komponenty* toto volání zpracuje.

Pro představu podoby syntaxe v knihovně React příkládám zdrojový kód 4.2.

Zdrojový kód 4.2: Příklad kódu psaného v knihovně React

```
var Create = React.createClass({
  getInitialState: function () {
    return { messages: null, success: [],
            products: null, customers: null }
  },
  componentWillMount: function () { .. Init Data .. },
  send: function (data, cb) { .. Send Data .. },
  render: function () {
    var load = !(this.state.customers == null
                || this.state.products == null);
    return (
      <Loader loaded={load}>
        <h2>Nov objedn vka </h2>
        <Form customers = {this.state.customers}
              messages = {this.state.messages}
              products = {this.state.products}
              onSend = {this.send}
              onMessage = {this.props.onMessage} />
      </Loader >);
  }
});
```

4.2 Databázový model

Pro vytvoření databáze jsem použil objektové relační mapování (ORM). Jedná se o techniku, která umožňuje převádět data mezi objektovým a relačním databázovým modelem[27].

Výhodami *ORM* je právě možnost pracovat s objektovým modelem a tento přístup nás proto více odstiňuje od vytváření databázových dotazů. Tím také zjednodušuje implementaci a zlepšuje čitelnost zdrojového kódu.

Konverzi dat ale často postihují různé komplikace označené termínem *object-relational impedance mismatch*. Kvůli tomu byly v publikaci od Martina Fowlera[11] popsány 4 návrhové vzory pro *ORM*, které by měly těmto problémům předcházet:

- **Table Data Gateway** je návrhový vzor, který slouží jako brána(gateway)

pro CRUD operace nad databázovou tabulkou. Každá tabulka databáze je reprezentována třídou. V té by měly být implementovány CRUD⁸ operace.

- **Row Data Gateway** je návrhový vzor, který slouží jako brána (gateway) pro CRUD operace nad jedním řádkem databázové tabulky. Atributy daných tříd pak odpovídají jednotlivým sloupcům jim odpovídajícím tabulek.
- **Active Record** je návrhový vzor, ve kterém pro každou tabulku a její řádek existuje odpovídající třída a její atributy. Každý objekt *Active Record* je zodpovědný za načítání a ukládání dat z, resp. do, databáze.
- **Data Mapper** je návrhový vzor, u kterého neexistují CRUD operace v jednotlivých třídách. Ty obstarává mapovací prostředek a třída je tak zbavena všech závislostí na databázi.

Další nevýhodou techniky *ORM* je poměrně velká režie a tím relativně nižší výkon.

V technologii .NET má vývojář možnost výběru z mnoha dostupných *ORM* produktů. Výběr *ORM* jsem v mém případě zúžil na tři *open source* produkty, se kterými jsem se již setkal:

- **Entity Framework Core**, dále jen EF, je nástroj od společnosti Microsoft, který je nástupcem Entity Framework 6. Jedná se o technologii pro ADO.NET⁹, kterou od roku 2008 vyvíjí společnost Microsoft. EF umožňuje práci na vyšší úrovni abstrakce a díky tomu je programátor téměř úplně odstíněn od použití jazyka SQL[29].
- **Dapper.NET** je velmi jednoduchý *ORM* nástroj, který je poskytován pod hlavičkou webového portálu StackExchange. V tomto nástroji vytváří vývojář dotazy podobné jazyku SQL a díky tomu je podle dostupných testů výkonnosti rychlejší než EF[30].
- **NHibernate** je posledním *ORM* nástrojem, který zde uvedu. Jedná se o nástroj vyvíjený komunitou NHibernate Community.

Především z osobních důvodů, a protože jsem našel o tomto *ORM* nejvíce dostupných informací na internetu, jsem se rozhodl pro použití EF. Tento framework je možné aplikovat třemi různými přístupy:

- **Database First Approach.** Tento přístup slouží k vytvoření datového modelu z existující databáze. Entity Framework z této databáze následně vygeneruje kódy entit datového modelu[31]. Ukázku z procesu vytváření zdrojového kódu můžete vidět na obrázku 4.3.

⁸Create,read,update,delete - vytvoření,čtení,editace,smazání.

⁹Množina tříd umožňující přístup k datům a databázovým systémům[28].

- **Model First Approach.** Tento přístup je založen na vytvoření doménového modelu a databáze pomocí ORM návrhářského nástroje. Vývojář v tomto nástroji navrhuje entity a jejich vzájemné vazby a nakonec nechá z tohoto návrhu vygenerovat databázi[31]. Práci v návrhářském nástroji můžete zpozorovat na obrázku 4.4.
- **Code First Approach.** Tento způsob spočívá ve vytvoření databáze z doménového modelu. Díky tomuto stačí navrhnout pouze entity a vztahy mezi nimi jako klasické třídy. Následně je z těchto tříd vygenerována databáze[31]. Příklad takto napsané entity jako klasické třídy si můžete prohlédnout v ukázce zdrojového kódu 4.3.

Zdrojový kód 4.3: Příklad třídy při použití metody Code First Approach

```
public class Category
{
    public int Id { get; set; }
    public string Name { get; set; }
}
public class Product
{
    public int Id { get; set; }
    public string Name { get; set; }
    public int CategoryId { get; set; }
}
```

Pro vytvoření databáze jsem vybral metodu *Code First Approach* – nejdříve byly navrženy entity a jejich vzájemné vazby, implementovány jako třídy a poté z nich byla vygenerována databáze. Tabulky ze vzniklé databáze dělím do kategorií, které charakterizují jejich použití. Výsledný diagram databáze je dostupný v příloze C.1.

4.2.1 Tabulky se synchronizovanými informacemi z portálu SharePoint

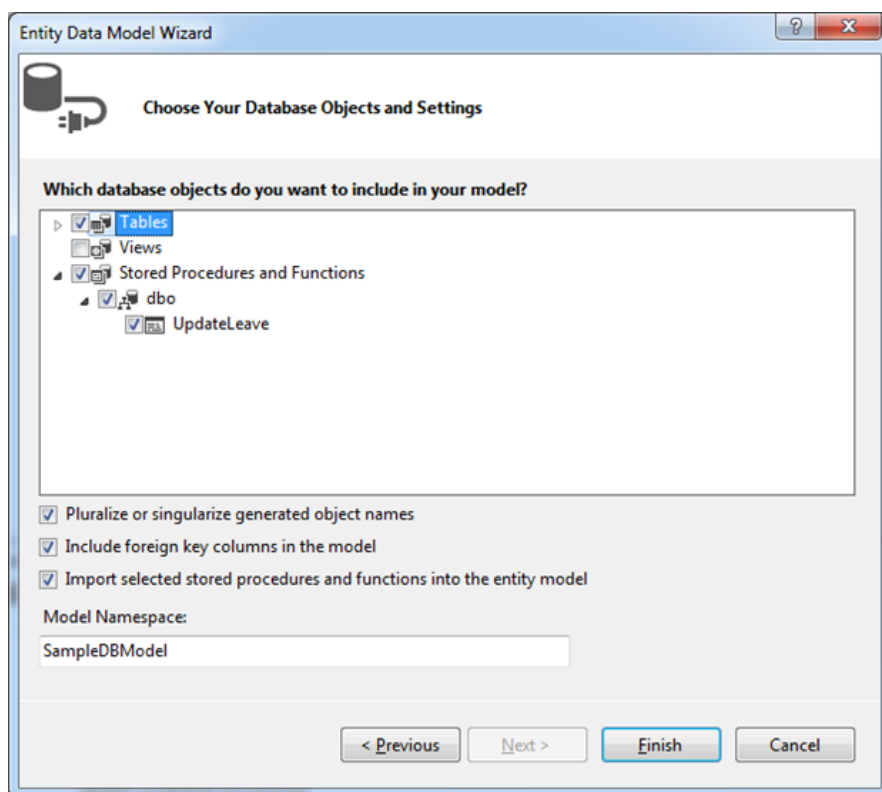
Tyto tabulky obsahují data, která se synchronizují s portálem SharePoint.

V této části evidujeme následující tabulky: `InfrastructureItem`, `Material`, `Product`, `ProductInfrastructureItem`, `ProductMaterial`.

4.2.2 Tabulky s informacemi o uživateli

Tyto tabulky obsahují informace o uživateli ve webové aplikaci.

4. NÁVRH



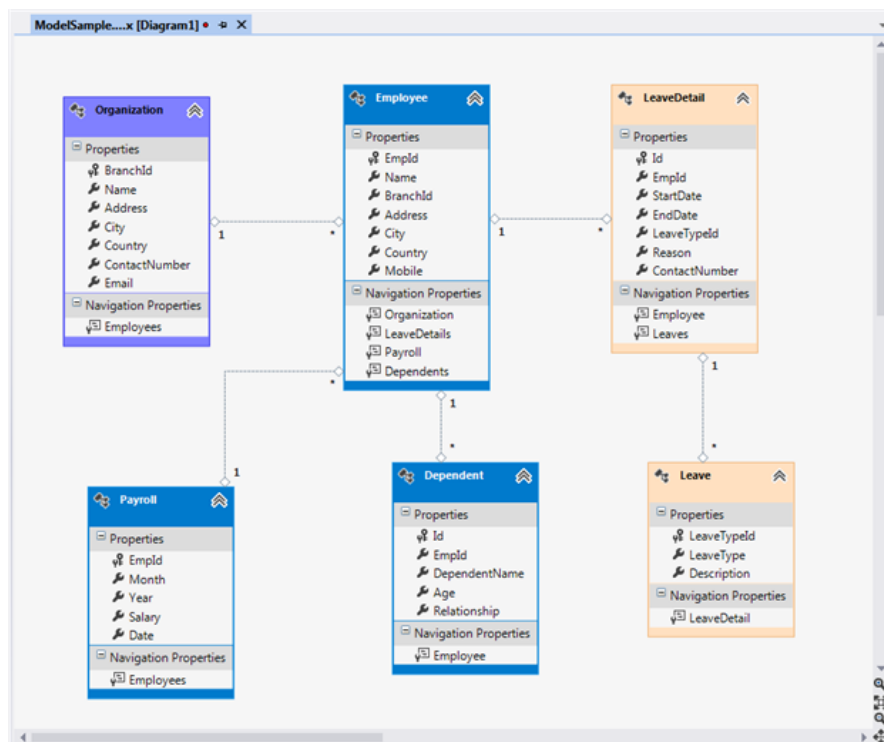
Obrázek 4.3: Průvodce generování zdrojového kódu entit a jejich vztahů z databáze v nástroji Microsoft Visual Studio.

V této části evidujeme následující tabulky: `AspNetRoleClaims`, `AspNetRoles`, `AspNetUserClaims`, `AspNetUserLogins`, `AspNetUserRoles`, `AspNetUsers`.

4.2.3 Tabulky webové aplikace

Tabulky webové aplikace, které obsahují informace o objednávkách, plánech výroby, jeho výsledcích a kontrolách. Dále tabulky s informacemi o zaměstnancích, dodavatelích, zákaznících, nastavení připojení k portálu SPF2013 a notifikacích.

V této části evidujeme následující tabulky: `Address`, `Batch`, `ChangeLog`, `Company`, `Customer`, `CustomerAddress`, `Deformation`, `Employee`, `Notification`, `Order`, `ProductionResult`, `ProductionResultBatch`, `ProductionResultInspection`, `SharePointSetting`, `Shift`, `ShiftEmployees`, `Supplier`, `SupplierAddress` a `WorkStack`.



Obrázek 4.4: Příklad modelu entit a jejich vztahů vytvořených pomocí návrhářského nástroje Microsoft Visual Studio.

4.3 Grafické rozhraní

WAHORZ je primárně určena pro webový prohlížeč běžící na osobním počítači či notebooku. Tomu také bylo přizpůsobeno grafické rozhraní, které není kvůli velkému množství robustních tabulek responzivní¹⁰ a může mít tudíž problémy při zobrazování na zařízeních s menší zobrazovací plochou – například na tabletech či mobilních telefonech.

Před samotným zpracováním grafického návrhu webové aplikace jsem vytvořil sadu wireframů¹¹ popisující rozmístění jednotlivých ovládacích prvků na webu. Ukázkou tohoto návrhu můžete vidět na obrázku 4.5.

Ve vrchní části webu se nachází hlavní menu celé aplikace. V pravé části nabídky jsou umístěny položky s uživatelskými akcemi – nepřihlášený uživatel zde nalezne odkazy pro přihlášení a registraci. Přihlášený uživatel zde nalezne nabídky pro odhlášení nebo správu účtu.

V levé části menu se nachází položky pro práci s webovou aplikací v závislosti na roli jednotlivých uživatelů. Například operátor výroby zde uvidí pouze odkaz pro správu produkce.

¹⁰Způsob zobrazení optimalizovaný pro všechny zobrazovací zařízení.

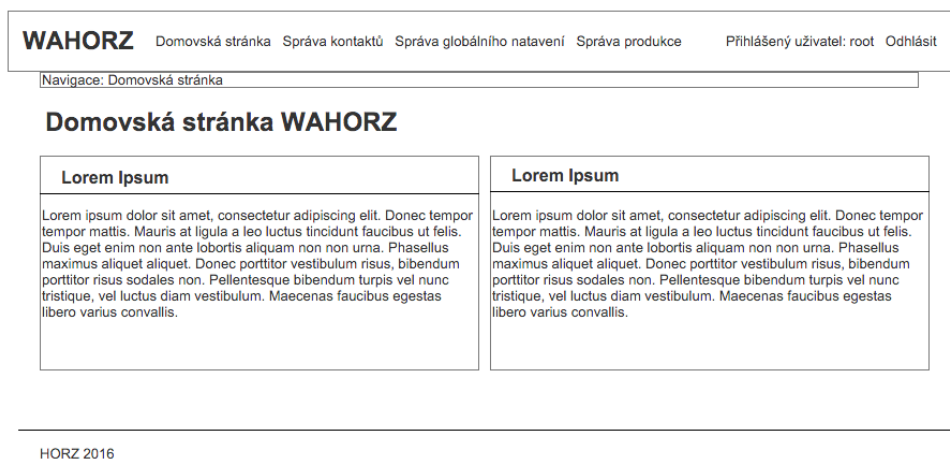
¹¹Drátový model, či skica webu.

4. NÁVRH

Pod nabídkou je umístěna drobečková navigace, která usnadní orientaci ve webové aplikaci.

Pod drobečkovou navigací se nachází obsah jednotlivých částí webu. Obsáhlejší sekce webové aplikace poskytují v této obsahové části další úroveň nabídky na levé straně. Tím však také snižují plochu poskytnutou části s obsahem o 25 %.

Uživatelské rozhraní je určeno pro prohlížení ve webovém prohlížeči a je psáno v jazyce HTML5. Pro vytvoření uživatelského rozhraní jsem využil volně dostupnou sadu nástrojů Bootstrap, která obsahuje návrhářské šablony využívající HTML, CSS a JavaScriptu. Dále jsem také využíval javascriptový framework jQuery.



Obrázek 4.5: Návrh drátového modelu pro domovskou stranu WAHORZ.

Realizace

V této kapitole popíší model nasazení a prostředí, ve kterých byla webová aplikace spouštěna. Následně popíší strukturu projektu, který byl implementován ve vývojovém prostředí Microsoft Visual Studio 2015 Community (MSVS2015). Na závěr popíší způsob přístupu uživatelů do aplikace.

5.1 Model nasazení

Webový systém bude spuštěn na serveru s operačním systémem Windows Microsoft Server 2012 R2 Standard pod webovou službou Microsoft Information Internet Services. Databáze Microsoft SQL Server 2014 Express bude spuštěna na stejném serveru.

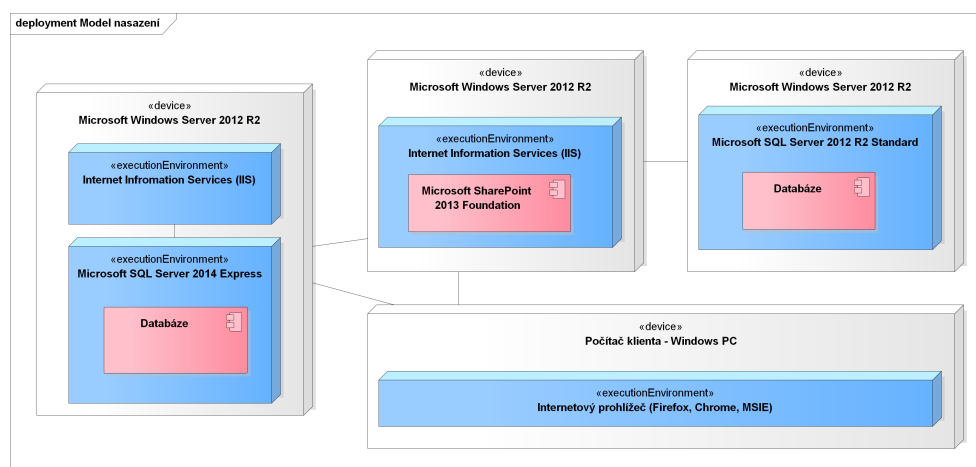
Systém SPF2013 je spuštěn na serveru, který je přístupný z internetové sítě. Databáze tohoto serveru je umístěna na jiném stroji. Synchronizace dat se serverem SPF2013 bude probíhat pomocí API rozhraní Windows SharePoint Services Web Service.

5.2 Struktura projektu

Tento projekt jsem rozdělil do několika modulů:

- ISHORZ je výchozí část projektu, která zajišťuje správné spuštění aplikace, nacházejí se zde *kontrolery*, třídy pro *DI*, *pohledy* a výchozí složka webového serveru.
- ISHORZ.Dbcontext je modul, který obsahuje informace o připojení k databázi a změny v databázi, které byly provedeny.
- ISHORZ.Models je modul, který obsahuje modely entit. Jedná se tedy o část, ze které byla následně vygenerována databáze.

5. REALIZACE



Obrázek 5.1: Návrh modelu nasazení.

- ISHORZ.SPConnect je modul, díky kterém je umožněno získat data ze serveru SPF2013.

Rád bych zde podrobněji uvedl, jak je každý z těchto modulů dále strukturovaný a popsal, jaké soubory se v daných adresářích nacházejí.

5.2.1 Modul ISHORZ

Hlavním modulem této webové aplikace je modul ISHORZ, který byl založen při vytvoření projektu ve MSVS2015. V kořeni adresářové struktury tohoto modulu se nachází několik velmi důležitých souborů, které slouží ke spuštění aplikace. Rád bych zde zmínil alespoň ty nejdůležitější:

- Soubor `appsettings.json` nese především zabezpečené informace, například o připojení k databázi. Jedná se o následníka konfiguračního souboru `web.config`, který je používán v nižších verzích ASP.NET MVC[32].
- Soubor `project.json` obsahuje informace o serverových závislostech na jiných projektech a balíčcích třetí stran. Dále obsahuje další důležité informace o nastavení serveru – například příkazy, které se mají provést před spuštěním serveru nebo seznam adresářů a souborů, kterých se netýká sestavení projektu[33].
- Soubor `Startup.cs` je soubor, který se provádí po spuštění serveru a načítá serverovou konfiguraci, konfiguraci tříd pro *DI*, *EF*, databázový kontext, mapování adres a podobně[33].

Tento modul dále obsahuje adresáře, do kterých jsou ukládány specifické zdrojové kódy.

5.2.1.1 Adresář Controllers

V adresáři `Controllers` se nachází *kontrolery* použité ve webové aplikaci. V kořeni této složky jsou umístěny *kontrolery*, které jsou volány při přístupu přes klasický pohled. Ve složce `api` jsou umístěny *kontrolery*, které se volají při přístupu k API rozhraní aplikace, tedy při používání části implementované pomocí knihovny React.

5.2.1.2 Adresář Helpers

V adresáři `Helpers` se nacházejí soubory se třídami, které usnadňují implementaci opakovaných úloh.

5.2.1.3 Adresář Services

V adresáři `Services` se nachází třídy pro *DI*.

5.2.1.4 Adresář ViewModels

V adresáři `ViewModels` se nachází soubory s pohledovými modely. Ty programátorovi usnadňují přechod mezi entitami ve tvaru, v jakém jsou například vyplněny formuláře nebo zobrazovány skutečná data na webové stránce, a skutečnými databázovými entitami. Pohledové modely by také měly obsahovat implementaci mapování mezi těmito třídami a skutečnými databázovými entitami.

5.2.1.5 Adresář Views

V adresáři `Views` se nachází soubory pohledové části aplikace. Tyto soubory mají příponu `cshtml` a jsou napsány ve značkovacím jazyku ASP.NET Razor.

5.2.1.6 Adresář Tests

Adresář `Tests` obsahuje soubory se třídami pro jednotkové testování webové aplikace. O jednotkovém testování se zmiňuji v kapitole 6.

5.2.1.7 Adresář wwwroot

Adresář `wwwroot` je kořenový adresář webového serveru. Ten obsahuje dokumenty, které jsou používány pro vykreslování webové stránky – HTML, CSS, JavaScript a použité obrázky.

5.2.2 Modul ISHORZ.Models

Pro definici modelů jsem vytvořil modul `ISHORZ.Models`, který obsahuje v adresáři `Models` soubory s třídami modelů webové aplikace.

Zdrojový kód 5.1: Úsek zdrojového kódu, který znázorňuje připojení k SPF2013

```
private SPService.Lists SPContext;
public SharePoint(string username,
                 string password,
                 string url){
    this.SPContext=new ISHORZ.SPConnect.SPService.Lists(){
        Url = url;
        Credentials=new NetworkCredential(username, password);
    };
    try {
        this.Lists=GetSharePointLists(sp.GetListCollection());
    } catch (Exception e) {
        throw e;
    }
    return;
}
```

5.2.3 Modul ISHORZ.Dbcontext

Pro definování podoby databáze a jejích vztahů s entitami aplikace jsem vytvořil modul `ISHORZ.Dbcontext`. V kořeni adresáře tohoto modulu se nachází soubor `ApplicationDbContext.cs`. Tento soubor obsahuje informaci, jaké entity z balíčku `ISHORZ.Models` budou uloženy v databázi, případně jejich vzájemné vztahy. Také určuje, jakým způsobem bude vytvořena databáze a jak bude k datům přistupovat EF.

Dále se zde nachází adresář `Migrations`. V tomto adresáři jsou soubory, které popisují změny ve třídách modelů, které se nazývají migrace. Z *migrací* se pak následně provádí aktualizace databáze.

5.2.4 Modul ISHORZ.SPConnect

Pro připojení k SPF2013 byl vytvořen modul `ISHORZ.SPConnect`. Zde se nachází soubory se třídami pro připojení k serveru SPF2013 a následné zpracování dat z formátu XML do listů. Toto připojení probíhá pomocí API rozhraní Windows SharePoint Services Web Service, které jsem popsal výše. K úspěšnému připojení je třeba zadat validní webovou adresu a přihlašovací údaje. Ukázkou zdrojového kódu, který slouží pro připojení k SPF2013, můžete vidět ve zdrojovém kódu 5.1.

5.2.5 Řízení přístupu

V této webové aplikaci bylo třeba zajistit, aby se k vybraným informacím dostali pouze vybraní uživatelé. Toho jsem dosáhl pomocí autentizace a následné autorizace.

Autentizace je proces, který ověřuje pravost uživatelem poskytnutých přihlašovacích údajů. Jakmile je uživatel autentizován, je mu poskytnuta identita, která je reprezentována nejčastěji souborem cookie[16].

Autorizace je proces povolení k provedení nějaké operace na základě předem definovaných pravidel. V kontextu této webové aplikace se jedná o přístup ke kontrolerům a jeho metodám dle vyžadované uživatelské role nebo identity[16].

Od verze ASP.NET MVC 5 je poskytováno rozhraní pro správu uživatelů s názvem ASP.NET Identity. Předchozí verze ASP.NET MVC sice poskytovaly podobné rozhraní pod názvem ASP.NET Membership, ale z důvodu špatného návrhu bylo nejdříve několikrát upraveno a nakonec úplně nahrazeno. Největšími přínosy nové verze je pak možnost několika způsobů autentizace, standardně s použitím databáze SQL, ve které jsou uchována uživatelská data[16].

Proces autorizace je rozdělen do tří základních vrstev:

- **Autorizace celého kontroleru** je způsob ověření oprávnění uživatele, kdy pro všechny metody kontroleru platí stejné pravidlo přístupu. Autorizace se v této metodě vztahuje na poskytnutí přístupu jedné či více uživatelských rolí.
- **Autorizace metody kontroleru** je způsob ověření oprávnění uživatele, kdy se pravidlem pro přístup řídí pouze daná metoda kontroleru. Autorizace se v této metodě vztahuje na poskytnutí přístupu jedné či více uživatelských rolí.
- **Autorizace v třídách pro *DI*** je způsob ověření oprávnění jednotlivých uživatelů, kteří se snaží dostat ke svým datům uloženým v databázi. Protože aplikace přistupuje k databázi pouze skrze třídy pro *DI*, bylo třeba tento typ autorizace umístit právě do těchto tříd.

V ASP.NET je autorizace podle uživatelské role zajištěna pomocí atributů, které se v kontrolerech zapisují před jednotlivé třídy resp. metody společně se seznamem povolených rolí. Po přístupu k danému kontroleru resp. metodě je vyhodnoceno uživatelské oprávnění a podle toho vrácen výstup webové aplikace.

Ověření jednotlivých uživatelů probíhá pomocí porovnávacích operátorů – na jedné straně je identifikátor aktuálně přihlášeného uživatele, na druhé straně je identifikátor uživatele s povoleným přístupem u dané položky. Pokud nastane rovnost, je uživatel připuštěn k provádění dané operace. V opačném případě je mu přístup zamítnut.

Testování

Po implementaci aplikace a jejím spuštění na testovacím serveru jsem vytvořil několik jednotkových testů a testovacích scénářů, abych zjistil, zda-li aplikace správně funguje a reaguje na akce uživatelů.

6.1 Jednotkové testování

Jednotkové testování je označení pro způsob automatizovaného testování, které slouží pro otestování kódu aplikace a jeho smyslem je zjistit, zda-li je tento kód v pořádku implementován a aplikace vrací očekávané hodnoty.

Ve verzi ASP.NETMVC 6 je pro účely *jednotkového testování* dostupných několik balíčků. Z důvodu nejlépe dohledatelné dokumentace jsem si vybral balíček s názvem xUnit. Ten jsem použil pro otestování *kontrolerů* a tříd pro *DI*.

6.2 Uživatelské testování

V poslední části vývoje aplikace jsem navštívil společnost HORZ s.r.o., abych společně se zaměstnanci otestoval uživatelské rozhraní a funkčnost WAHORZ. Předem musím oznámit, že se nejednalo o plnohodnotné testování webové aplikace, protože jako vývojář mám k této aplikaci vztah.

I přesto jsem se rozhodl vytvořit několik scénářů, které poté vybraní testeři prováděli. V průběhu testování aplikace jsem pak zjišťoval, jestli je aplikace schopna dostatečně obsloužit zákazníky.

Tohoto testování se zúčastnili tři uživatelé webové aplikace s rozdílnými rolami. Každý uživatel vyplnil před začátkem testu vstupní dotazník a po skončení testování výstupní dotazník.

6.2.1 Scénáře

- **Scénář č.1 – pro vedení společnosti.** Prvním úkolem vedení společnosti bylo přihlásit se do webové aplikace a následně provést synchronizaci dat se systémem SPF2013. Poté měl uživatel za úkol schválit registrovaného uživatele a přiřadit mu roli vedoucího výroby.
- **Scénář č.2 – pro vedení společnosti.** Druhým úkolem bylo přihlášení uživatele do webové aplikace a následné vytvoření objednávky dle vlastního uvážení.
- **Scénář č.3 – pro vedení výrobní sekce.** Prvním úkolem pro zaměstnance vedení výroby bylo přihlášení uživatele do systému a následné zjištění objednávky s nejbližším datem expedice. Této objednávce měl dále uživatel za úkol přiřadit plán výroby na datum před požadovaným termínem expedice objednávky.
- **Scénář č.4 – pro vedení výrobní sekce.** Ve druhém scénáři pro vedení výrobní sekce byla po uživateli vyžadována editace výrobního plánu, vyplnění výsledku a kontroly výroby tohoto plánu. Nakonec byl uživatel vyzván k ukončení objednávky.
- **Scénář č.5 – pro operátory výroby.** Prvním úkolem pro operátora výroby bylo přihlášení do webové aplikace. Následovalo zobrazení výrobního plánu na aktuální den.
- **Scénář č.6 – pro operátory výroby.** Druhým úkolem operátora výroby bylo vyplnění výsledku výroby v aktuálním dni a odhlášení z webové aplikace.

6.2.2 Shrnutí testování

Podle získaných informací ze vstupních dotazníků jsem se dozvěděl, že nejmenší zkušenost s ovládáním počítačů má *operátor výroby*. Uživatelé typu *vedení výroby* a *vedení společnosti* uvedli, že jsou mírně pokročilými uživateli PC.

Podobně pak také dopadlo samotné testování – uživatelé s většími zkušenostmi vykazovali lepší orientaci ve WAHORZ, kdežto uživatel s menší zkušeností s ovládáním počítačů si občas nevěděl rady.

To mě vedlo k tomu, aby byla aplikace později doladěna a aby byly některé ovládací prvky viditelnější – to se týká především správy vlastní výroby pro *operátory výroby*.

Po otestování WAHORZ měli uživatelé možnost projít si aplikaci sami a já byl přítom připraven jim odpovídat na případné dotazy.

Závěr

Mezi hlavní cíle této bakalářské práce bylo vytvořit webovou aplikaci, která splňuje požadavky zákazníka HORZ s.r.o., a kterou bude tato společnost, na rozdíl od současných systémů, používat pravidelně s větším nadšením. Jak se tento záměr povedl zatím nemohu soudit, protože nedošlo k úplnému dokončení nasazení WAHORZ v této společnosti, to je plánováno v létě roku 2016.

Na základě analýzy procesů ve společnosti jsem zjistil informace, které mi s následným návrhem WAHORZ velice pomohly. Co se týká správy produkce, jedná se především o úkony, které společnost provádí v době mezi založením a expedicí objednávky.

Při implementaci samotného systému jsem dbal především na snadnou orientaci, kterou jsem předem konzultoval se zaměstnanci firmy.

Také jsem se snažil naimplementovat jádro systému tak, aby bylo do budoucna snadno rozšiřitelné. Toho jsem docílil především díky použití architektury *MVC*, která oddělila jednotlivé části aplikace, a použití *DI*, které mi poskytlo volnost při implementaci předem navržených rozhraní.

Případné rozšíření WAHORZ se týká především podpory komunikace s účetním programem POHODA a plánované mobilní aplikace pro tablety a telefony. Každý zaměstnanec by pak byl na danou směnu vybaven tímto zařízením. Díky tomu by měl informace o plánu výroby a možnost vyplnit výsledky produkce v momentě provádění dané práce.

Doufám, že společnost HORZ s.r.o. bude s fungováním WAHORZ spokojena a budu s ní moci v budoucnu navázat další spolupráci, například při implementaci výše zmíněných plánovaných rozšíření či při jakémkoliv jiné možnosti spolupráce týkající se informačních systémů.

Literatura

- [1] HORZ s.r.o.: *HORZ TJ STADION TEPLICE spol. s r.o., zpracování plastů, výroba dílů z EPDM a polyethylenu, výroba a úprava forem*. [online]. [cit. 2016-05-07]. Dostupné z: <http://www.horz.cz>
- [2] komise. EU, E.: *Nová definice malých a středních podniků*. Lucemburk: EUR-OP, 2006, ISBN 9289479175.
- [3] Ministerstvo financí ČR: *Výpis z Registru ekonomických subjektů ČSÚ v ARESV[online]*. [cit. 2016-02-05]. Dostupné z: http://www.info.mfcr.cz/cgi-bin/ares/darv_res.cgi?ico=61534196&jazyk=cz&xml=1
- [4] Microsoft: *Plan browser support in SharePoint 2013*. [cit. 2016-05-07]. Dostupné z: <https://technet.microsoft.com/en-us/library/cc263526.aspx>
- [5] Office Team: *Office 2010 System Requirements[online]*. Jan 2010, [cit. 2016-02-05]. Dostupné z: <https://blogs.technet.microsoft.com/office2010/2010/01/22/office-2010-system-requirements/>
- [6] AIIM: *What is SharePoint?[online]*. [cit. 2016-02-05]. Dostupné z: <http://www.aiim.org/what-is-microsoft-sharepoint>
- [7] Microsoft: *SharePoint Online[online]*. [cit. 2016-02-05]. Dostupné z: <https://products.office.com/cs-cz/sharepoint/sharepoint-online-collaboration-software>
- [8] Microsoft: *Licencování SharePointu 2013[online]*. [cit. 2016-02-05]. Dostupné z: <https://products.office.com/cs-cz/sharepoint/sharepoint-licensing-overview>
- [9] Microsoft: *Hardware and software requirements for SharePoint 2013[online]*. Mar 2016, [cit. 2016-02-05]. Dostupné z: <https://technet.microsoft.com/en-us/library/cc262485.aspx>

- [10] Fpweb.net: *COMPARE SHAREPOINT 2003 WITH SHAREPOINT 2010 AND 2013*[online]. [cit. 2016-02-05]. Dostupné z: <http://www.fpweb.net/sharepoint-2013/compare-sharepoint-2003-2010-2013/>
- [11] Fowler, M.: *Patterns of Enterprise Application Architecture*. Addison-Wesley Professional, 2002, ISBN 0321127420.
- [12] Zeeshan, A. A.: *Programming in Java using the MVC architecture*[online]. Feb 2015, [cit. 2016-02-05]. Dostupné z: <http://www.codeproject.com/articles/879896/programming-in-java-using-the-mvc-architecture>
- [13] Microsoft: *Microsoft .NET Home*[online]. 2016, [cit. 2016-02-05]. Dostupné z: <https://www.microsoft.com/net/>
- [14] Microsoft: *Common Language Runtime (CLR)*[online]. [cit. 2016-02-05]. Dostupné z: <https://msdn.microsoft.com/en-us/library/8bs2ecf4>
- [15] Smith, S.: *Dependency Injection*[online]. Microsoft, [cit. 2016-02-05]. Dostupné z: <https://docs.asp.net/en/latest/fundamentals/dependency-injection.html>
- [16] Freeman, A.: *Pro ASP.NET MVC 5 Platform*. Apress, 2014, ISBN 1430265418.
- [17] Walther, S.: *Top 10 Changes in ASP.NET 5 and MVC 6*[online]. Apr 2015, [cit. 2016-02-05]. Dostupné z: <http://stephenwalther.com/archive/2015/02/24/top-10-changes-in-asp-net-5-and-mvc-6>
- [18] Microsoft: *Microsoft Open Database Connectivity (ODBC)*[online]. [cit. 2016-02-05]. Dostupné z: [https://msdn.microsoft.com/en-us/library/ms710252\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/ms710252(v=vs.85).aspx)
- [19] Microsoft: *Features Supported by the Editions of SQL Server 2014*[online]. [cit. 2016-02-05]. Dostupné z: [https://msdn.microsoft.com/en-us/library/cc645993\(v=sql.120\).aspx](https://msdn.microsoft.com/en-us/library/cc645993(v=sql.120).aspx)
- [20] Microsoft: *Hardware and software requirements for SharePoint 2013*[online]. [cit. 2016-02-05]. Dostupné z: <https://technet.microsoft.com/en-us/library/cc262485.aspx>
- [21] Microsoft: *Introduction to the Windows SharePoint Services Web Service*[online]. [cit. 2016-02-05]. Dostupné z: [https://msdn.microsoft.com/en-us/library/dd587241\(v=office.11\).aspx](https://msdn.microsoft.com/en-us/library/dd587241(v=office.11).aspx)
- [22] W3Schools: *ASP.NET Razor - Markup*[online]. [cit. 2016-02-05]. Dostupné z: http://www.w3schools.com/aspnet/razor_intro.asp

-
- [23] Fedosejev, A.: *React.js Essentials*. Packt Publishing - ebooks Account, 2016, ISBN 1783551623.
- [24] React: *Why React?* Dostupné z: <https://facebook.github.io/react/docs/why-react.html>
- [25] React: *Tutorial*. Dostupné z: <https://facebook.github.io/react/docs/tutorial.html>
- [26] Vjeux: *Community Round-Up 11*[online]. Nov 2013, [cit. 2016-02-05]. Dostupné z: <https://facebook.github.io/react/blog/2013/11/18/community-roundup-11.html>
- [27] Keith, M.; Schnicariol, M.: Object-Relational Mapping. In *Pro JPA 2*, Springer, 2009, s. 69–106.
- [28] Microsoft: *ADO.NET Overview*[online]. [cit. 2016-02-05]. Dostupné z: [https://msdn.microsoft.com/en-us/library/h43ks021\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/h43ks021(v=vs.110).aspx)
- [29] Adya, A.; Blakeley, J. A.; Melnik, S.; aj.: Anatomy of the ado. net entity framework. In *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, ACM, 2007, s. 877–888.
- [30] ppanyukov: *Performance: Entity Framework 7 vs. Dapper.net vs. raw ADO.NET*[online]. May 2015, [cit. 2016-02-05]. Dostupné z: (<https://ppanyukov.github.io/2015/05/20/entity-framework-7-performance.html>)
- [31] Kamalasanan, A. K.: *Different Approaches of Entity Framework*[online]. Červenec 2013, [cit. 2016-02-05]. Dostupné z: <https://www.simple-talk.com/dotnet/.net-framework/different-approaches-of-entity-framework/>
- [32] Steve Smith, S. A.: *Migrating Configuration From ASP.NET MVC 5 to MVC 6*[online]. Microsoft, [cit. 2016-02-05]. Dostupné z: <https://docs.asp.net/en/latest/migration/configuration.html>
- [33] Steve Smith, E. R.: *Understanding ASP.NET 5 Web Apps*[online]. Microsoft, [cit. 2016-02-05]. Dostupné z: <https://docs.asp.net/en/latest/conceptual-overview/understanding-aspnet5-apps.html#the-project-json-file>

Seznam použitých zkratek

- API** Application Programming Interface
- BI** Business Intelligence
- CLR** Common Language Runtime
- DI** Dependency Injection
- DOM** Document Object Model
- EF** Entity Framework
- HTML** HyperText Markup Language
- MSVS2015** Microsoft Visual Studio 2015 Community
- MVC** Model View Controller
- ODBC** Open Database Connectivity
- ORM** Objektově relační mapování
- RAM** Random-access memory
- REST** Representational State Transfer
- SOAP** Simple Object Access Protocol
- SPF2013** Microsoft SharePoint Foundation 2013
- SPO2010** Microsoft Office 2010
- SQL** Structured Query Language
- WAHORZ** Webová aplikace pro objednávky a plánování výroby
- XML** Extensible Markup Language

A. SEZNAM POUŽITÝCH ZKRATEK

XSS Cross Site Scripting

Instalační příručka

Testovací instance WAHORZ bude v omezené době přístupná na URL adrese <http://s2.kalibrace.cz:54321>.

B.1 Instalační požadavky

Serverovou stranu WAHORZ lze sputit pouze na platformě Microsoft Windows na webovém serveru Internet Information Service (IIS) ve verzi 7.5 a výše. Do IIS je třeba nainstalovat modul `httpPlatformHandler` (k dispozici ke stažení na webové adrese <https://docs.asp.net/en/latest/publishing/iis.html>).

Dalším požadavkem je instalace relačního databázového systému Microsoft SQL Server. Doporučuji použít verzi Microsoft SQL Server 2014 Express, kterou jsem využíval při implementaci.

Ze složky `bin` na přiloženém CD zkopírujte obě složky `approot` a `wwwroot` do předem zvoleného adresáře ve Vašem počítači, který budu v následujícím textu označovat jako `app`.

B.1.1 Nastavení databáze

Vytvořte na Vašem databázovém serveru novou databázi, kterou pojmenujete `horz` a vytvořte k ní uživatelský přístup přes autentizaci SQL serveru. Postup, jak vytvořit uživatelský přístup v databázi naleznete na webové adrese <https://msdn.microsoft.com/en-us/library/aa337562.aspx>.

Nyní se přesuňte do adresáře `app/approot/src/ISHORZ/` a otevřete soubor `appsettings.json` v textovém editoru. V sedmém řádku tohoto souboru změňte přístupové údaje k vytvořené SQL databázi:

- Text `WIN-SMVG50DKIF\SQLSERVER` nahraďte názvem Vašeho databázového serveru.
- Text `horz_agent` nahraďte jménem vytvořeného uživatele.

- Text `Heslo123?!` nahradte heslem k vytvořenému uživateli.

Soubor uložte a zavřete.

B.1.2 Vytvoření webové stránky v IIS

Tento postup je prováděn v Internet Information Service Manager, který byl nastaven v anglickém jazyce.

- Otevřete IIS manažer.
- V levé části pod vybraným serverem klikněte pravým tlačítkem na složku **Sites** a vyberte **Add Website**.
- V následujícím formuláři vyberte cestu ke složce `adresar/wwwroot` a zvolte si konfiguraci tohoto webu dle vlastního rozhodnutí, minimálně zadejte port aplikace, a stiskněte OK.
- Vyberte nově vzniklou položku aplikace v levé části **IIS** a v prostřední části otevřete **Configuration Editor**.
- Pod položkou **section** vyberte **system.webServer/handlers** a ujistěte se, že v pravé části aplikace není tato sekce uzamčena.
- V levé části se přesuňte do sekce **Application Pools**, najděte aplikační fond Vámi vytvořené aplikace a dvojklikem na tento řádek otevřete nastavení tohoto fondu aplikací.
- Pod položkou **.NET CLR version** vyberte možnost **No Managed Code** a stiskněte OK.

V této chvíli by měla být WAHORZ připravena ke spuštění. První spuštění aplikace trvá o trochu déle kvůli inicializaci databáze.

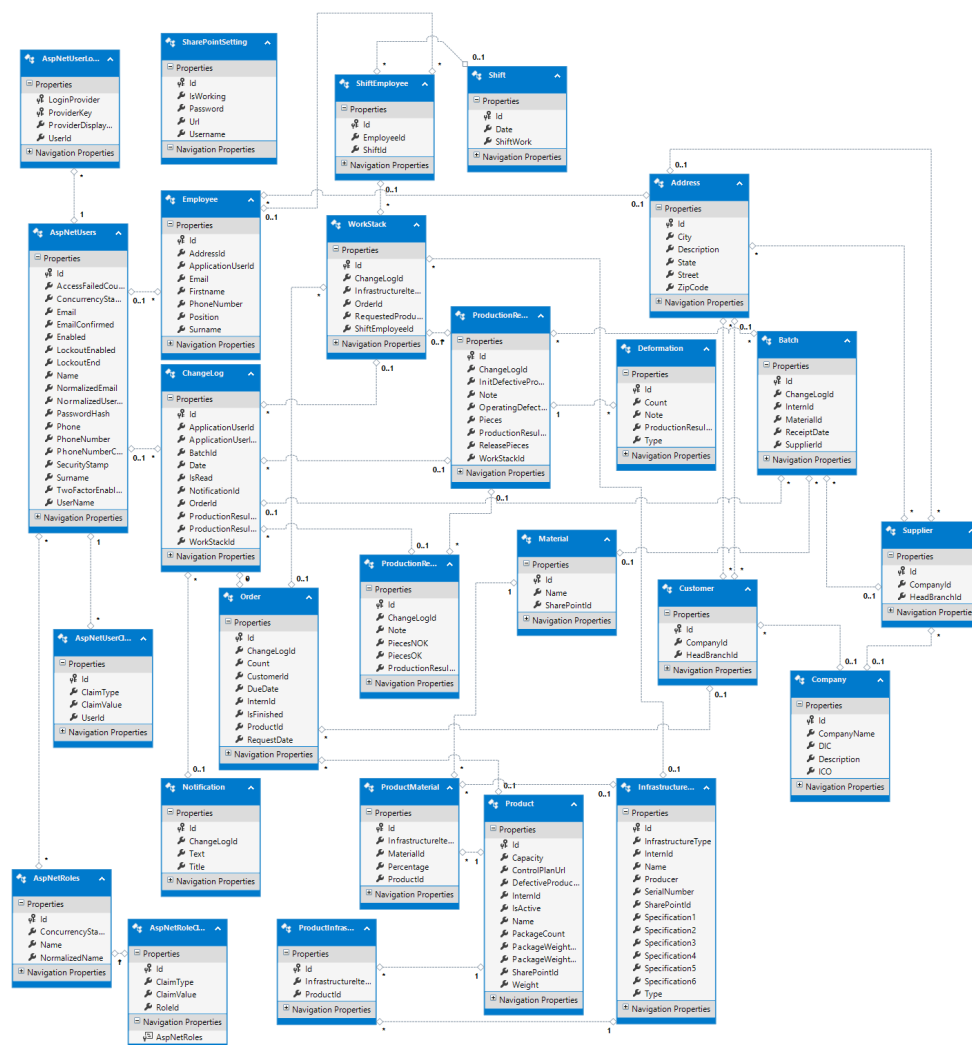
B.2 Spuštění aplikace ve webovém prohlížeči

Aplikace se spouští ve webových prohlížečích, které byly zahrnuty v nefunkčních požadavcích. Prohlížeč musí mít povolený JavaScript.

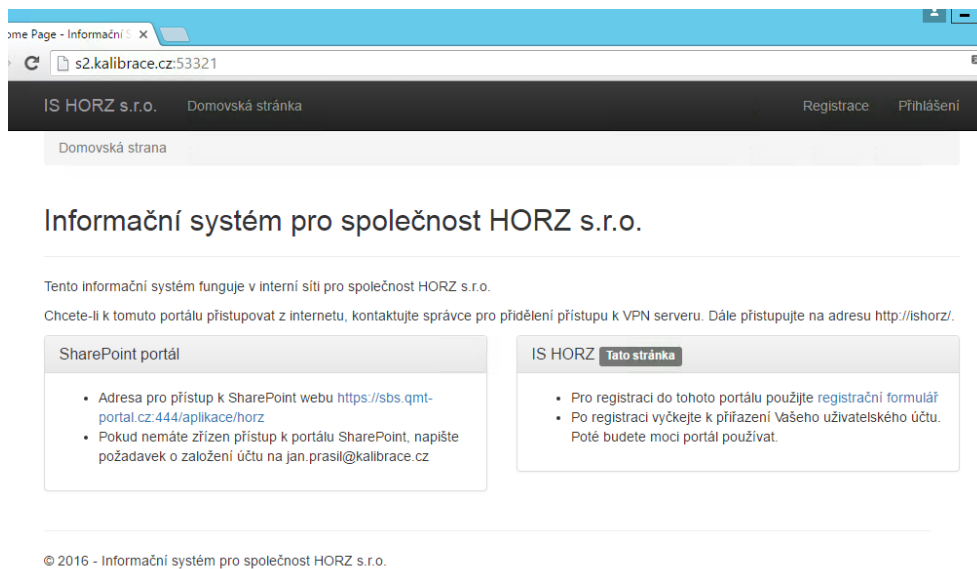
PŘÍLOHA **C**

Přílohy

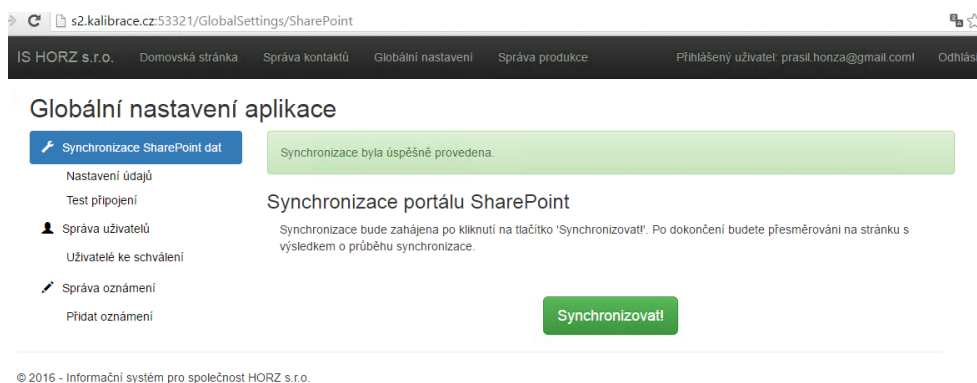
C. PŘÍLOHY



Obrázek C.1: Databázový diagram WAHORZ. Pro lepší čitelnost doporučuji verzi na CD.



Obrázek C.2: Úvodní obrazovka WAHORZ pro nepřihlášeného uživatele.



Obrázek C.3: Obrazovka s oznámením o úspěšné synchronizaci.

IS HORZ s.r.o. Domovská stránka Správa kontaktů Globální nastavení Správa produkce Přihlášený uživatel: prasil.honz@gmail.com Odhlásit

Správa produkce společnosti

- Seznam objednávek
- Šarže materiálu
 - Seznam šarží
 - Nová šarže materiálu
- Plán výroby
 - Seznam plánů výroby
 - Nový plán výroby
 - Zobrazit plán pro uživatele
- Kontrola výroby
 - Seznam položek ke kontrole
 - Seznam provedených kontrol

Nová objednávka

Identifikace šarže

Datum zadání

Datum dodávky

Vyberte zákazníka

Vyberte požadovaný produkt

Zadejte požadované množství

Obrázek C.4: Obrazovka s formulářem pro založení nové objednávky.

Obsah přiloženého CD

Níže je popsána adresářová struktura na přiloženém CD.

```
readme.txt.....stručný popis obsahu CD
bin.....adresář s instalačním balíčkem serveru
src
├── impl.....zdrojové kódy implementace
├── thesis.....zdrojová forma práce ve formátu LATEX
├── img.....obrazové přílohy
└── text.....text práce
    └── thesis.pdf.....text práce ve formátu PDF
```