

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra počítačů

**System pro podporu výroby**

**Tomáš Kotlík**

Školitel: Ing. Božena Mannová, PhD.  
Květen 2019



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Kotlík** Jméno: **Tomáš** Osobní číslo: **301088**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávací katedra/ústav: **Katedra počítačů**  
Studijní program: **Softwarové technologie a management**  
Studijní obor: **Softwarové inženýrství**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**System pro podporu výroby**

Název bakalářské práce anglicky:

**Production support system**

Pokyny pro vypracování:

Cílem práce je navrhnout a implementovat systém pro podporu výroby v podniku zabývajícím se leteckou malosériovou výrobou. Systém má umožnit plánování výroby, správu technické dokumentace pro výrobu, správu technologií a norem, sledování kvality výroby, sledování stavu zásob materiálu a dokončené výroby.

Seznamte se problematikou výroby v podniku, pro který je systém určen. Na základě získaných informací specifikujte a analyzujte konkrétní funkční a nefunkční požadavky na systém. Na základě těchto požadavků navrhnete aplikaci. Pro analýzu a návrh použijte vhodné prostředky SI. Navržený program implementujte a výslednou aplikaci otestujte.

Seznam doporučené literatury:

Ian Sommerville: Software Engineering, Global Edition, Pearson Higher Ed, 2016, ISBN1292096144  
Arlow, J., Neustat, I.: UML 2 a unifikovaný proces vývoje aplikací. Computer Press, 2007

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Božena Mannová, Ph.D., Software Engineering and Networking FEL**

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **09.02.2018**

Termín odevzdání bakalářské práce:

Platnost zadání bakalářské práce: **30.09.**

\_\_\_\_\_  
podpis vedoucí(ho) práce

\_\_\_\_\_  
podpis vedoucí(ho) ústavu/katedry

\_\_\_\_\_  
prof. Ing. Pavel Rípků, CSc.  
podpis děkana(ky)

## III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta



## Poděkování

Chtěl bych poděkovat své rodině za soustavnou a trpělivou podporu při studiu. Rovněž bych chtěl vyjádřit hluboký dík své vedoucí práce, která se mi vždy snažila maximálně pomoci a bez jejíž podpory by tato práce vznikala jen obtížně.

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 24. 5. 2019

## Abstrakt

Tato práce se zabývá analýzou, návrhem a implementací systému pro podporu výroby v malém podniku působícím v oblasti leteckého průmyslu. Systém má podpořit plynulou malosériovou výrobu pomocí přehledného plánování, sledování stavu zásob materiálu a dokončené výroby, sledování kvality výroby a správy technologií a technické dokumentace. Klasifikací spadá navrhovaný systém mezi Enterprise resource planning systémy. Součástí práce je prototyp systému v podobě webové podnikové aplikace implementované programovacími jazyky Java a Javascript s použitím frameworku Spring a knihovny React.

**Klíčová slova:** Systém pro podporu výroby, webová podniková aplikace, SPA, Java, Javascript, Spring framework, React.js

**Školitel:** Ing. Božena Mannová, PhD.

## Abstract

This work deals with the analysis, design and implementation of a system to production support in small aviation plant. The system is intended to support small-lot production by help of clear planning, material stock monitoring, and finished manufacturing, and quality monitoring of product and technology documentation. By classification the proposed system belongs among Enterprise Resource Planning Systems. The work also includes a prototype of a web-based enterprise application implemented in Java and Javascript programming languages using the Spring framework and React library.

**Keywords:** Production support system, web based business application, SPA, Java, Javascript, Spring framework, React.js

**Title translation:** Production support system

# Obsah

<b>1 Úvod</b>	<b>1</b>	3.4 Systémové požadavky	28
<b>2 Analýza</b>	<b>3</b>	3.4.1 Funkční požadavky	28
2.1 Specifika letecké výroby	3	3.4.2 Nefunkční požadavky	34
2.2 Současný stav	3	3.5 Architektura systému	34
2.2.1 Plánování	3	3.6 Modely procesů	35
2.2.2 Výroba	4	3.6.1 Model procesu vytvoření prototypu	35
2.2.3 Skladování	4	3.6.2 Model procesu uskladnění dílu	36
2.2.4 Kontrola kvality	4	3.6.3 Model procesu vytvoření plánu	37
2.2.5 Správa technologií	5	<b>4 Implementace</b>	<b>39</b>
2.3 Nedostatky současného stavu	5	4.1 Použité technologie	39
2.3.1 Plánování	6	4.1.1 Programovací jazyky	39
2.3.2 Výroba	6	4.1.2 Knihovny a frameworky	40
2.3.3 Skladování	6	4.1.3 Podpůrné aplikace	40
2.3.4 Správa technologií	6	4.2 Implementace serveru	40
2.3.5 Kontrola kvality	6	<b>5 Testování</b>	<b>43</b>
2.4 Formulace požadavků	6	5.1 Jednotkové testy	43
2.4.1 Business požadavky	7	5.2 Integrované testy	44
2.4.2 Uživatelské požadavky	7	<b>6 Závěr</b>	<b>47</b>
2.5 Možnosti řešení	10	6.1 Hodnocení projektu	47
2.5.1 Existující systémy pro podporu výroby	10	6.2 Možná vylepšení systému	47
<b>3 Návrh</b>	<b>13</b>	<b>Literatura</b>	<b>49</b>
3.1 Případy užití	13	<b>A Obsah doprovodného optického disku</b>	<b>51</b>
3.1.1 Aktéři	13		
3.1.2 Komponenty případů užití	14		
3.1.3 Případy užití - Společné	14		
3.1.4 Případy užití - Administrace	15		
3.1.5 Případy užití - Technologie	16		
3.1.6 Případy užití - Kontrola kvality	17		
3.1.7 Případy užití - Sklad	18		
3.1.8 Případy užití - Plánování výroby	20		
3.1.9 Případy užití - Výroba	20		
3.2 Scénáře případů užití	21		
3.2.1 Scénář Vytvořit prototyp sestavy	22		
3.2.2 Scénář Vytvořit plán výroby	22		
3.3 Doménový model	23		
3.3.1 Administrace	23		
3.3.2 Technologie	24		
3.3.3 Kvalita	24		
3.3.4 Sklad	25		
3.3.5 Plánování	26		
3.3.6 Výroba	27		

## Obrázky

3.1 Hierarchie aktérů . . . . .	14
3.2 Případy užití - Společné . . . . .	15
3.3 Případy užití - Administrace . . . . .	16
3.4 Případy užití - Technologie . . . . .	17
3.5 Případy užití - Kontrola kvality . . . . .	18
3.6 Případy užití - Správa skladových prostor . . . . .	18
3.7 Případy užití - Skladování materiálu a výrobků . . . . .	19
3.8 Případy užití - Plánování výroby . . . . .	20
3.9 Případy užití - Výroba . . . . .	21
3.10 Doménový model - Administrace . . . . .	23
3.11 Doménový model - Technologie . . . . .	24
3.12 Doménový model - Kvalita . . . . .	25
3.13 Doménový model - Skladování . . . . .	26
3.14 Doménový model - Plánování výroby . . . . .	27
3.15 Doménový model - Výroba . . . . .	28
3.16 Architektura systému - pohled na celek . . . . .	34
3.17 Architektura systému - komponenty klienta . . . . .	34
3.18 Architektura systému - komponenty serveru . . . . .	35
3.19 Model procesu - vytvoření prototypu . . . . .	36
3.20 Model procesu - uskladnění dílu . . . . .	37
3.21 Model procesu - vytvoření plánu . . . . .	38

## Tabulky

2.1 Business požadavky . . . . .	7
2.2 Uživatelské požadavky rozvíjející BR1. . . . .	7
2.3 Uživatelské požadavky rozvíjející BR2. . . . .	8
2.4 Uživatelské požadavky rozvíjející BR3. . . . .	8
2.5 Uživatelské požadavky rozvíjející BR4. . . . .	9
2.6 Uživatelské požadavky rozvíjející BR5. . . . .	9
2.7 Uživatelské požadavky rozvíjející BR6. . . . .	9
2.8 Uživatelské požadavky rozvíjející BR7. . . . .	9
2.9 Uživatelské požadavky rozvíjející BR11. . . . .	10
3.1 Funkční požadavky - společné . . . . .	29
3.2 Funkční požadavky - administrace . . . . .	29
3.3 Funkční požadavky - technologie . . . . .	30
3.4 Funkční požadavky - skladování . . . . .	31
3.5 Funkční požadavky - kontrola kvality . . . . .	32
3.6 Funkční požadavky - plánování výroby . . . . .	32
3.7 Funkční požadavky - výroba . . . . .	33
3.8 Nefunkční požadavky . . . . .	34





# Kapitola 1

## Úvod

Firma Eratipa s.r.o. (dále jen Eratipa) se již od roku 2006 zabývá výrobou leteckých konstrukcí. Hlavní náplní jejího výrobního programu je výroba částí draku letadla PC-6 Turboporter (dále jen PC-6). Tuto výrobu realizuje pro svého obchodního partnera, švýcarskou leteckou firmu Pilatus ltd.(dále jen Pilatus), který je vlastníkem licencí a certifikací k letounu PC-6. Z pohledu firmy Pilatus jde o outsourcing části výroby PC-6 do podniku Eratipa.

Po celou dobu existence firmy se ukazuje, že klíčem pro ekonomický úspěch je plynulá výroba, skrze kterou je firma schopna dodržovat termíny dodávek odběrateli. Nedodržením termínů nedochází pouze ke zpoždění plateb za odebrané výrobky, ale i k penalizaci odběratelem formou srážek z domluvené ceny výrobků.

Cílem této práce je analyzovat procesy ovlivňující výrobu. Na základě této analýzy navrhnout a implementovat prototyp systému pro podporu výroby, který umožní udržet výrobu plynulou a plánovanou tak, aby bylo možné dodržovat termíny dodávek výrobků odběrateli.



## Kapitola 2

### Analýza

#### 2.1 Specifika letecké výroby

Pro lepší pochopení dalšího textu je vhodné zdůraznit zvláštnosti letecké výroby, které nemusí být na první pohled zřejmé.

V letecké výrobě se z dílů vyrábí výrobek sestavy, zkráceně sestava (anglicky assembly). Tato sestava může plnit funkci dílu pro jinou sestavu, pro zdůraznění tohoto vztahu o ní lze hovořit jako o podsestavě (anglicky subassembly).

Seznam dílů, včetně podsestav, potřebných k výrobě sestavy je uveden v kusovníku, který je součástí výkresu sestavení pro danou sestavu.

Nejčastějším druhem práce v letecké výrobě je opracovávání, lícování a následné spojování kovových dílů nýtováním. Zhotovitel vyrábí sestavu na univerzálním pracovním stole nebo na přípravku speciálně zkonstruovaném pro konkrétní sestavu.

Letecká výroba se řídí přísnými normami. Platí v ní princip dohledatelnosti, tedy možnosti určit, kdo a kdy se na výrobě dané části letounu podílel.

#### 2.2 Současný stav

Současný stav lze nejlépe popsat procesy, které mají vliv na výrobu a její hospodářské výsledky. Těmito procesy jsou plánování, výroba, skladování, kontrola kvality a správa technologií.

##### 2.2.1 Plánování

Proces plánování výroby vychází z dlouhodobého plánu, který je rámcově dán kontraktem s odběratelem. Na jeho základě je stanoven roční výrobní plán, který rovněž musí být odsouhlasen odběratelem, který se tím zavazuje odebrat naplánované vyrobené množství výrobků.

Roční plán výroby rovněž stanovuje minimální objem expedované dodávky odběrateli, čímž určuje cykly ve výrobě a termíny, které je nutno dodržet.

Plán výroby je vyhotoven v tištěné podobě, zdrojem dat jsou soubory tabulkových procesorů. Dílčí plány a termíny zahájení výroby jednotlivých sestav

jsou zpracovány v podobě písemných poznámek a dokumentů tabulkových procesorů. Zdrojem dat jsou soubory s přehledy normohodin pro nejdůležitější vyráběné sestavy, jejichž tvůrcem je technolog.

### ■ 2.2.2 Výroba

Proces výroby začíná vydáním výrobního příkazu, jehož součástí je výrobní protokol k vyráběné sestavě, zhotoviteli. V tomto protokolu je kromě údajů identifikujících vyrobenou sestavu také uveden seznam výrobních kroků, jejichž správné provedení stvrzuje svým podpisem zhotovitel a kontrolor kvality. Následně je vydána technická dokumentace pro vyráběnou sestavu. Ta obsahuje technické výkresy a kusovník sestavy, na základě kterého je vyskládněn materiál.

Plnění výroby je kontrolováno porovnáním normohodin určených pro výrobu sestavy proti upsanému času práce na sestavě.

Výroba sestavy končí závěrečnou kontrolou kvality výrobku provedenou společně kontrolorem kvality a zhotovitelem, potvrzením protokolu sestavy a předáním výrobku k uskladnění. Přehled o probíhající výrobě si manažer výroby tvoří na základě informací o vydaných výrobních protokolech a na základě pochůzek po pracovištích.

### ■ 2.2.3 Skladování

Proces skladování zahrnuje uskladnění materiálu od dodavatele, uskladnění výrobků z výroby, vyskládnění materiálu a výrobků do výroby a vyskládnění výrobků pro expedici k odběrateli.

Uskladnění materiálu od dodavatele probíhá v rámci domluvených dodávek. Jde o jednorázový proces, kdy se dodaný materiál kontroluje proti objednávce a dodacímu listu. Následně je materiál přemístěn na sklad. Přírůstek materiálu na skladě je evidován manažerem výroby, který má k dispozici dodací list s poznámkami o nedodaném, případně poškozeném materiálu.

Uskladnění výrobků z výroby probíhá převímkou hotového výrobku od zhotovitele skladníkovi, který výrobek umístí do skladu. Přírůstek výrobku na skladě je evidován manažerem výroby na základě protokolu k vyrobené sestavě.

Vyskládnění materiálu a výrobků do výroby probíhá převímkou materiálu dělníkem na základě kusovníku vyráběné sestavy. Úbytek materiálu a výrobků je evidován manažerem výroby na základě vydaného protokolu k sestavě.

Vyskládnění výrobků pro expedici k odběrateli probíhá na základě objednávky odběratele. Úbytek výrobků je evidován manažerem výroby, který je odpovědný za expedici.

### ■ 2.2.4 Kontrola kvality

Proces kontroly kvality výroby ovlivňuje procesy výroby a skladování. Vždy je v něm posuzován materiál nebo výrobek sestavy z hlediska jeho shody s

platnou technickou dokumentací, většinou reprezentovanou platným technickým výkresem, a platnými normami. Neshody s technickou dokumentací musí kontrolor kvality konzultovat s technickým oddělením odběratele. Pouze to je oprávněno vydat instrukce, jak dále pokračovat. Zpravidla jde o rozhodnutí zda neshodu ponechat a informace o ní zanést do speciálního protokolu o neshodě, případně neshodu odstranit opravou, nebo celý výrobek označit jako zmetek a vyrobit nový.

Kontrola kvality při procesu výroby probíhá pravidelně po jednotlivých výrobních krocích uvedených v protokolu sestavy a na závěr při dokončení výroby. Dalším případem jsou mimořádné kontroly, které jsou iniciovány zpravidla podezřením, že není dodržen výrobní postup nebo příslušné normy.

Kontrola kvality při procesu skladování sleduje kvalitu dodaného materiálu a skladování zmetků. Karanténa je speciální druh skladu, kde jsou skladovány zmetky dílů a sestav. Ty jsou vždy při nejbližším termínu expedice vyskladněny a odeslány k odběrateli k dalšímu zpracování, nejčastěji likvidaci.

Záznamy o kontrolách kvality jsou vedeny kontrolorem kvality v podobě protokolů o kvalitě. Ty jsou odesílány odběrateli spolu s dodávkou odpovídajících výrobků. Na straně Eratipy se uchovávají kopie těchto protokolů a na jejich základě je vypracován čtvrtletní přehled o kvalitě výroby.

### ■ 2.2.5 Správa technologií

Proces správy technologií spočívá ve správě tištěné technické dokumentace. Tu tvoří technické výkresy, popisy výrobních postupů a technologií. Protože část této dokumentace pochází od odběratele, firmy Pilatus, musí se správa dokumentace řídit požadavky odběratele, mezi které patří striktní zákaz digitalizace jakékoliv výkresové dokumentace, evidence počtu kopií platné dokumentace a povinnost vrátit všechnu neaktuální technickou dokumentaci firmě Pilatus. Dalším požadavkem je zabezpečení dokumentace proti krádeži, proto je uchována v archivu s omezeným přístupem.

Výrobní postupy jsou většinou ustaveny na základě zkušeností a znalostí zhotovitelů. Vedeny jsou v podobě písemných poznámek. Ty jsou jako výrobní know-how oficiálně majetkem Eratipy, za jejich správu a aktualizaci však odpovídají zhotovitelé, kteří je mají trvale k dispozici na svém pracovišti.

Přehled o technické dokumentaci a přístup k ní řídí technolog. Ten vydává zhotoviteli příslušnou dokumentaci podle protokolu vyráběné sestavy. Po dokončení výroby kontroluje její vrácení zhotovitelem. V případě poškození či zničení dokumentace řeší s odběratelem její náhradu.

## ■ 2.3 Nedostatky současného stavu

Na základě výše uvedeného popisu současného stavu spolu s rozhovory vedenými s pracovníky zodpovídajícími za nejdůležitější firemní procesy byly identifikovány tyto nedostatky.

### ■ 2.3.1 Plánování

Hlavním nedostatkem je netransparentnost výrobního plánu. Současný systém poznámek a souborů neposkytuje jasný a rychlý pohled na plán výroby.

Dalším nedostatkem je nepružné sdílení vytvořeného plánu. Přehled o plánované výrobě je nutný nejen pro manažera výroby, ale i pro technologa. Technolog na jeho základě musí včas vytvořit příslušné výrobní protokoly a vydat technickou dokumentaci zhotovitelům.

### ■ 2.3.2 Výroba

V současných podmínkách nelze jednoduše zjistit stav výroby jinak pohovorem s kontrolorem kvality, případně prohlídkou stavu práce na pracovišti a pohovorem se zhotovitelem.

### ■ 2.3.3 Skladování

Vyskladnění do výroby a uskladnění výrobku provádí zhotovitelé. Změny stavu skladu se tedy zjistí pouze, pokud je o tom skladník informován. Stav materiálu je sledován na úrovni dávek pro jednotlivé sestavy. Obtížně se sleduje mimořádné vydávání materiálu potřebného k opravám. Informace o stavu zásob se podávají ústně mezi skladníkem a manažerem výroby. Kontrola materiálu potřebného k výrobě tak musí proběhnout vícekrát, než by bylo třeba, kdyby se výdej materiálu do výroby sjednotil a byl k němu oprávněn pouze skladník.

### ■ 2.3.4 Správa technologií

Mezi technologie patří i výrobní postupy. Ty jsou spravovány a aktualizovány zhotoviteli, neexistuje však jednotný přehled o již existujících postupech. To brání například doplnění chybějících pracovních postupů technologem nebo snažším sdílením postupů samotnými zhotoviteli.

### ■ 2.3.5 Kontrola kvality

Souhrnná čtvrtletní zpráva o kvalitě výroby neposkytuje dostatečný přehled na úrovni sestav. Takový přehled by mohl například pomoci při organizaci dodatečného školení zhotovitelů nebo revizi pracovních postupů.

## ■ 2.4 Formulace požadavků

Na základě analýzy současného stavu a jeho nedostatků je možno formulovat požadavky, jejichž uspokojením má dojít k zlepšení popsáném v cíli této práce. Nejprve jsou definovány business požadavky, které se svým zaměřením a úrovní detailu opírají o potřeby organizace jako celku. Rozpracováním business požadavků do větší míry detailu zahrnující potřeby konkrétní skupiny uživatelů vzniknou uživatelské požadavky.

Požadavky na této úrovni analýzy musí být formulovány srozumitelně jak pro zadavatele, tak i pro řešitele. Z tohoto důvodu jsou popsány přirozenou mluvou.

### 2.4.1 Business požadavky

Z pohledu organizace byly identifikovány business požadavky uvedené v následující tabulce 2.1 .

ID	Požadavek
BR1	Potřeba přehledného plánování.
BR2	Potřeba přehledu o výrobě.
BR3	Potřeba přehledného skladování
BR4	Potřeba přehledné kontroly kvality výroby.
BR5	Potřeba přehledné správy technické dokumentace.
BR6	Potřeba přehledné správy výrobních technologií.
BR7	Potřeba přehledné správy norem.
BR8	Potřeba využít stávající technické vybavení pro provoz systému.
BR9	Potřeba nízkých finančních nákladů na provoz systému.
BR10	Potřeba spolehlivosti systému.
BR11	Potřeba rozlišení uživatelů podle pracovního zařazení.
BR12	Potřeba komunikovat se systémem v češtině.

**Tabulka 2.1:** Business požadavky

### 2.4.2 Uživatelské požadavky

Uživatelské požadavky rozvíjející potřebu přehledného plánování (BR1) jsou uvedeny v tabulce 2.2.

ID	Požadavek
UR1	Manažer výroby potřebuje vytvořit plán výroby, protože mu to umožní určit včasné termíny zahájení výroby.
UR2	Manažer výroby potřebuje schválit plán výroby, protože tím se stane použitelným pro výrobu.
UR3	Manažer výroby potřebuje prohlížet již vytvořené plány výroby, protože pro něj mohou mít informační hodnotu.
UR4	Manažer výroby potřebuje prohlížet detail plánu výroby, protože to má informační hodnotu.
UR5	Manažer výroby potřebuje informace o chybějících dílech pro výrobu, protože mu je to umožní včas objednat.
UR6	Manažer výroby potřebuje informace o chybějících pod-sestavách pro vkládané úkoly plánu, protože mu je to umožní dodatečně naplánovat k výrobě.

**Tabulka 2.2:** Uživatelské požadavky rozvíjející BR1.

Uživatelské požadavky rozvíjející potřebu přehledu o výrobě (BR2) jsou uvedeny v tabulce 2.3.

ID	Požadavek
UR7	Manažer výroby potřebuje pohled na etapu výroby schváleného plánu, protože tak lze ověřit jeho plnění.
UR8	Manažer výroby potřebuje založit úkol výroby, protože to je krok výroby a plánu.
UR9	Manažer výroby potřebuje vytvořit požadavek na vyskladnění dílů pro úkol výroby.
UR10	Manažer výroby potřebuje aktualizovat údaje pro úkol výroby, protože tím mění stav výroby.
UR11	Manažer výroby potřebuje ukončit úkol výroby, protože tím se posouvá k dalšímu úkolu výroby.
UR12	Manažer výroby potřebuje vytvořit požadavek na uskladnění hotového výrobku nebo jej poslat jako díl zpět do dalšího úkolu výroby, protože tím udržuje přehlednost stavu skladu a výroby.
UR13	Manažer výroby potřebuje informaci o předpokládaném dokončení celé výroby, protože tak okamžitě zjistí, zda se podaří dodržet plánované termíny.
UR14	Manažer výroby potřebuje uzavřít etapu výroby, protože tím indikuje konec výroby podle daného plánu.
UR15	Manažer výroby potřebuje nahradit zmetek novou sestavou v již prováděném výrobním kroku, protože sestavu musí vyrobit.

**Tabulka 2.3:** Uživatelské požadavky rozvíjející BR2.

Uživatelské požadavky rozvíjející potřebu přehledného skladování (BR3) jsou uvedeny v tabulce 2.4.

ID	Požadavek
UR16	Skladník potřebuje spravovat skladové prostory, protože za ně nese zodpovědnost.
UR17	Skladník potřebuje vystavit příjemku na sklad, protože je to potřeba pro uskladnění dílu.
UR18	Skladník potřebuje vystavit výdejku ze skladu, protože je to potřeba pro vyskladnění dílu.
UR19	Skladník potřebuje přehled dílů na skladu, protože to zlepší dohledatelnost.

**Tabulka 2.4:** Uživatelské požadavky rozvíjející BR3.

Uživatelské požadavky rozvíjející potřebu přehledné kontroly kvality výroby (BR4) jsou uvedeny v tabulce 2.5.

Uživatelské požadavky rozvíjející potřebu přehledné správy technické dokumentace (BR5) jsou uvedeny v tabulce 2.6.



ID	Požadavek
UR20	Kontrolor kvality potřebuje přehled položek v karanténě, protože je musí postupně odesílat odběrateli.
UR21	Kontrolor kvality potřebuje posoudit kvalitu vyráběné sestavy, protože to vyžadují předpisy.
UR22	Kontrolor kvality potřebuje vystavit příjemku do karantény, protože to posiluje dohledatelnost.
UR23	Kontrolor kvality potřebuje vystavit výdejku z karantény, protože to posiluje dohledatelnost.
UR24	Kontrolor kvality potřebuje přehled všech ukončených kontrol kvality, protože to zlepšuje dohledatelnost.

**Tabulka 2.5:** Uživatelské požadavky rozvíjející BR4.

ID	Požadavek
UR25	Technolog potřebuje vytvořit katalog technické dokumentace, protože to zlepší přehlednost archivu dokumentace.

**Tabulka 2.6:** Uživatelské požadavky rozvíjející BR5.

Uživatelské požadavky rozvíjející potřebu přehledné správy výrobních technologií (BR6) jsou uvedeny v tabulce 2.7.

ID	Požadavek
UR26	Technolog potřebuje vytvořit katalog dílů a sestav, protože to usnadní proces plánování a skladování.
UR27	Technolog potřebuje vytvořit kusovníky sestav, protože to usnadní proces plánování, výroby a skladování.
UR28	Technolog potřebuje katalog pracovních postupů, protože to umožní snáze zaškolovat pracovníky.

**Tabulka 2.7:** Uživatelské požadavky rozvíjející BR6.

Uživatelské požadavky rozvíjející potřebu přehledné správy norem a směrnic (BR7) jsou uvedeny v tabulce 2.8.

ID	Požadavek
UR29	Technolog potřebuje seznam platných norem a směrnic, protože se jimi řídí procesy výroby, skladování, kvality a technologie.

**Tabulka 2.8:** Uživatelské požadavky rozvíjející BR7.

Uživatelské požadavky rozvíjející potřebu rozlišení uživatelů podle pracovního zařazení (BR11) jsou uvedeny v tabulce 2.9.

ID	Požadavek
UR30	Správce systému potřebuje spravovat uživatele včetně přidělení rolí, protože uživatelé mají být rozlišeni.

**Tabulka 2.9:** Uživatelské požadavky rozvíjející BR11.

## 2.5 Možnosti řešení

Naplnění business požadavků lze dosáhnout třemi způsoby, které se liší komplexností řešení, finančními náklady na zavedení a nároky na obsluhu.

První možnost je upravit stávající systém závaznými pravidly pro vedení a sdílení dokumentů. Pravidla vymezí formát, způsob sdílení a oprávněné uživatele dokumentů. Toto řešení má minimální finanční náklady, vyžaduje však od zainteresovaných pracovníků vysokou pracovní morálku a nasazení. Neřeší rychlost a přehlednost, která závisí plně na pracovnících. Proto tuto možnost lze považovat pouze za provizorní řešení.

Druhou možností je použít již nějaký existující ERP systém. Zde jsou rizika vysokých finančních nákladů a malé možnosti plně postihnout specifika firemních procesů. Možné alternativy jsou uvedeny v následující podkapitole.

Třetí možností je vytvořit nový systém pro podporu výroby rozpracováním business požadavků do funkcionalit a omezení systému. Toto řešení má největší šanci postihnout specifika firemních procesů. Finanční náročnost je příznivější ve srovnání s druhou výše uvedenou možností a to zejména z důvodu rozsahu očekávané implementované funkcionality.

Všechna výše uvedená řešení rovněž předpokládají úpravu některých firemních procesů, zejména procesu výroby a skladování ve styčném bodě výdeje materiálu do výroby, kde je nutné omezit možnost zhotovitelů vyskladnit si bez záznamu materiál.

### 2.5.1 Existující systémy pro podporu výroby

Systémy pro podporu výroby spadají do kategorie ERP systémů. Proto byl k nalezení již existujících systémů použit portál organizace ERP Focus, která se věnuje celé této oblasti i z pohledu softwarového inženýrství.

Pro potřeby této práce byly hledány systémy pomocí portálu výše zmíněné organizace splňující následující kritéria vyhledávání.

- Business sector: manufacturing.
- Industry: aerospace and defense.
- System hosting: installed on premise.
- ERP software features: inventory management, planning and scheduling, quality control, warehouse management.

Výsledek hledání byl následně zúžen na 3 produkty, které jsou cíleny na malé výrobní podniky. Tím byly z přehledu vyloučeny aplikace jako SAP

ERP nebo Microsoft Dynamics AX, které se specializují na střední a větší podniky. Veškeré údaje jsou platné ke květnu 2018.

#### **DBA Manufacturing.**

- Tvůrce: DBA Software.
- Stránky produktu: [www.dbamanufacturing.com](http://www.dbamanufacturing.com)
- Komponenty navíc: ano.
- Konfigurace u zákazníka: ano.
- Počet nabízených licencí: 5, 10+.
- Cena za licenci: 599USD, 499USD.
- Zpoplatněná podpora: ano.
- Cena za podporu: 295USD za 6 měsíců.

#### **E2 Manufacturing system.**

- Tvůrce: Shoptech Software.
- Stránky produktu: [www.shoptech.com/e2-manufacturing-system/](http://www.shoptech.com/e2-manufacturing-system/)
- Komponenty navíc: ano.
- Konfigurace u zákazníka: ano.
- Počet nabízených licencí: nevedeno.
- Cena za licenci: nevedeno.
- Zpoplatněná podpora: ano.
- Cena za podporu: nevedeno.

#### **Q2S ERP.**

- Tvůrce: FeneTech.
- Stránky produktu: [q2serp.com](http://q2serp.com)
- Komponenty navíc: ano.
- Konfigurace u zákazníka: ano.
- Počet nabízených licencí: nevedeno.
- Cena za licenci: nevedeno.
- Zpoplatněná podpora: ano.
- Cena za podporu: nevedeno.

Některé ERP systémy neuvádí v přehledech ceny. V tom případě pro potřeby srovnání uvažujeme nejhorší možnou variantu, tedy že pro danou položku je neuvedená cena vždy vyšší než cena uvedená.

Z přehledu je patrné, že existující systémy jsou nákladné, navíc vyžadují pokračující platby za poskytovanou podporu. Při zavedení těchto systémů je nutné provést konfiguraci, která je rovněž nákladná. Systémy navíc obsahují komponenty, které by nenašly ve firmě Eratipa uplatnění. Proto se jeví jako nejschůdnější řešení vývoj systému podpory výroby na míru.

# Kapitola 3

## Návrh

Tato část se zabývá návrhem systému a jeho jednotlivých částí. Případy užití vymezují základní funkcionalitu očekávanou od systému. Detailnější pohled na zejména komplexnější případy užití nabízí scénáře případů užití. Doménový model poskytuje pohled na systém z hlediska datových entit a jejich vzájemných vztahů. Systémové požadavky, rozdělené dále na funkcionální a nefunkcionální požadavky představují již detailní pohled na funkcionalitu systému a jeho parametry a omezení.

### 3.1 Případy užití

Případy užití definují skupiny uživatelů systému, tzv. aktéry, a jejich interakce se systémem, případně s podmnožinou systému, zvané případy užití. Vzhledem k rozsahu navrhovaného systému se jeví rozumné rozdělit případy užití do komponent podle podnikových procesů, kterých se týkají.

#### 3.1.1 Aktéři

V systému vystupují tito aktéři:

- Uživatel
- Administrátor
- Skladník
- Technolog
- Kontrolor kvality
- Manažer výroby

**Uživatel** je společným předkem všech zbývajících aktérů. Jeho případy užití jsou děděny všemi potomky.

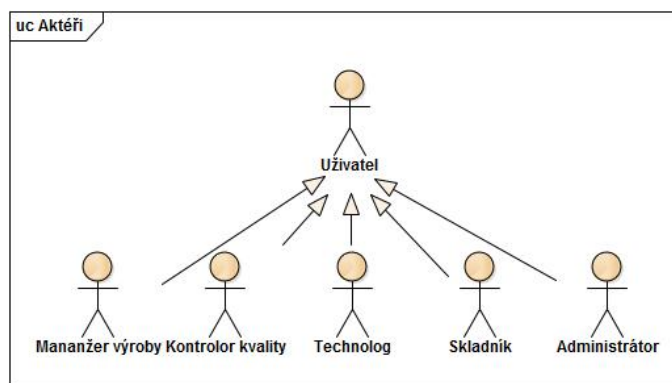
**Administrátor** spravuje uživatelské účty v systému.

**Skladník** spravuje skladové prostory a je zodpovědný za uskladnění a vyskladnění materiálu, polotovarů a hotových výrobků.

**Technolog** spravuje technickou dokumentaci a výrobní technologie firmy.

**Kontrolor kvality** provádí kontrolu kvality výrobků a materiálu.  
**Manažer výroby** vytváří a schvaluje plány výroby. Rovněž dohlíží na plnění výroby.

Aktéři v systému tvoří hierarchii, kterou zobrazuje diagram na obrázku 3.1. Z něj je patrné, že všichni aktéři sdílí množinu případů užití, která přísluší



**Obrázek 3.1:** Hierarchie aktérů

jejich společnému předkovi.

### ■ 3.1.2 Komponenty případů užití

Z pohledu podnikových procesů popsanych v předchozím textu byly případy užití rozděleny do následujících komponent:

- Společná
- Administrace
- Technologie
- Sklad
- Kvalita
- Plánování výroby
- Výroba

Podrobnější pohled na jednotlivé komponenty nabízejí následující kapitoly.

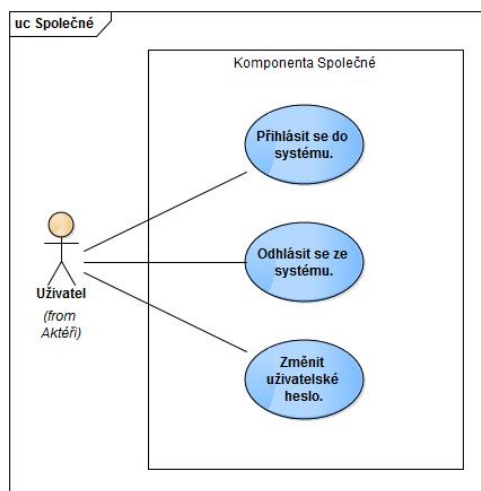
### ■ 3.1.3 Případy užití - Společné

Komponenta sdružuje interakce aktéra **Uživatel** a obsahuje případy užití, které se týkají přístupu do systému. Mezi ně patří:

- Přihlásit se do systému. V systému nelze pracovat bez přihlášení. Přihlášení se děje pomocí uživatelského jména a hesla.

- Odhlásit se ze systému.
- Změnit heslo. Uživatel má možnost změnit si své přihlašovací heslo.

Výše uvedené zachycuje diagram na obrázku 3.2.



Obrázek 3.2: Případy užití - Společné

### 3.1.4 Případy užití - Administrace

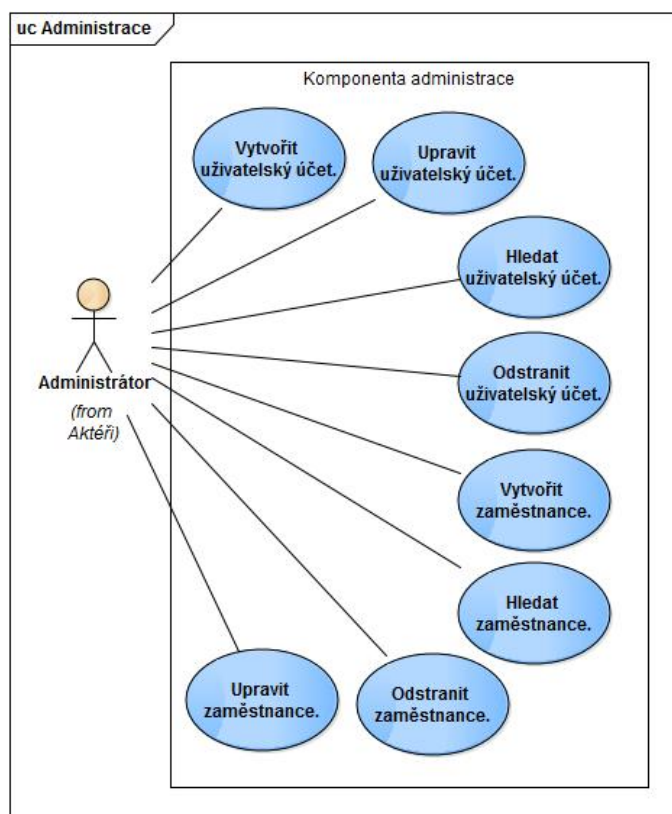
Komponenta sdružuje případy užití, jejichž cíli jsou správa uživatelských účtů a správa zaměstnanců. Z hlediska podpory výroby jde o případy užití plnicí v systému podpůrnou roli. Jejich prostřednictvím jsou do systému zaváděni například noví uživatelé a zaměstnanci.

**Administrátorovi** nabízí:

- Vytvořit uživatelský účet. Každý uživatel systému musí mít právě jeden aktivní uživatelský účet pod kterým v systému vystupuje.
- Upravit uživatelský účet. Umožňuje upravit libovolný údaj uživatelského účtu. Včetně aktivace a deaktivace účtu a nastavení nového hesla účtu.
- Hledat uživatelský účet. Vyhledání podle uživatelského jména, jména a nebo příjmení vlastníka účtu. Výsledek hledání zobrazen jako seznam účtů.
- Odstranit uživatelský účet. Umožňuje odstranit účet, který byl založen omylem. Není ekvivalentní s deaktivací účtu bývalých pracovníků, které jsou dále v systému přítomny.
- Vytvořit zaměstnance. Vytvoří záznam o zaměstnanci, který je použit dále v systému, zejména při výrobě a kontrole kvality.
- Hledat zaměstnance. Hledání zaměstnance podle zaměstnaneckého čísla (firma zaměstnává i cizince, kteří nemají rodné číslo), příjmení a pracovního zařazení.

- Upravit zaměstnance. Upravení libovolného údaje záznamu zaměstnance.
- Odstranit zaměstnance. Odstranění záznamu zaměstnance ze systému, pouze pro záznamy, které vznikly omylem.

Výše uvedené zachycuje diagram na obrázku 3.3.



Obrázek 3.3: Případy užití - Administrace

### 3.1.5 Případy užití - Technologie

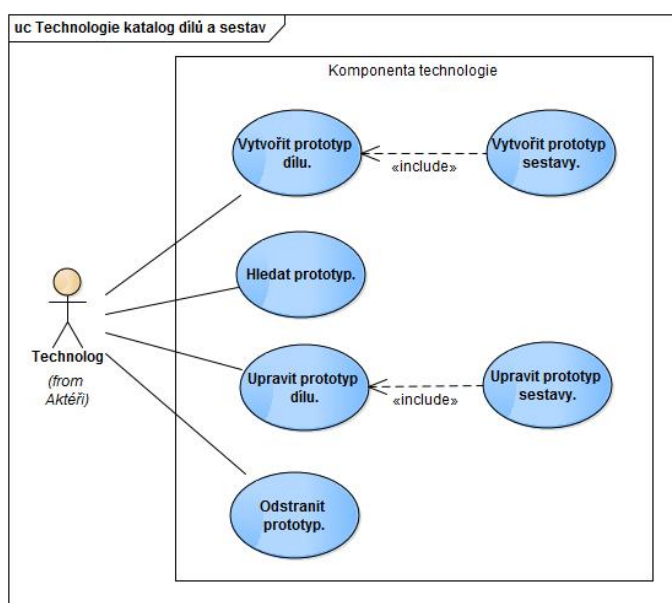
Komponenta technologie sdružuje případy užití, které mají za úkol vnést do systému technologické údaje nezbytné pro použití v systémových procesech skladování, plánování a výroby. Jedná se zejména o tvorbu prototypů dílů a sestav, které slouží jako vzory pro výrobky sestav. Aktérem v této komponentě je **Technolog**, jehož možnosti jsou:

- Vytvořit prototyp dílu. Vytvoří prototyp dílu, který slouží jako vzor pro fyzické díly uskladněné jako výrobní materiál.
- Vytvořit prototyp sestavy. Vytvoření prototypu sestavy, která je z hlediska technologie rozšířením prototypu dílu, spojená s kusovníkem.
- Hledat prototyp. Vyhledání prototypu dílu podle výkresového čísla.
- Upravit prototyp dílu. Upravení libovolného údaje prototypu dílu.



- Upravit prototyp sestavy. Upravení libovolného údaje prototypu sestavy, včetně úpravy kusovníku.
- Odstranit prototyp. Odstraní prototyp dílu nebo sestavy včetně kusovníku. Pouze pro prototypy vzniklé omylem, které nefigurují jako díly na jiných kusovnících sestav.

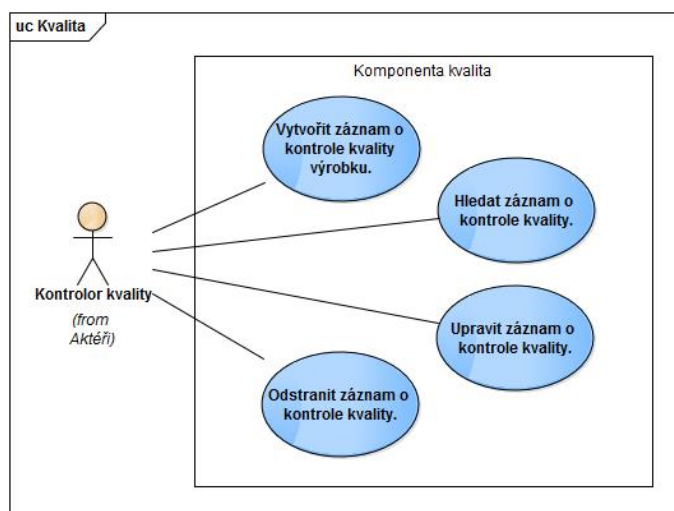
Pro lepší ilustraci vztahu případů užití prototypu dílu a prototypu sestavy může sloužit následující diagram 3.4.



Obrázek 3.4: Případy užití - Technologie

### 3.1.6 Případy užití - Kontrola kvality

Komponenta kvalita sdružuje případy užití, které slouží pro vytvoření a správu záznamů o kontrolách kvality pro výrobky sestav. Dotýká se tak podnikového procesu výroby a skladování. Možnosti aktéra v komponentě plně vystihuje diagram 3.5.

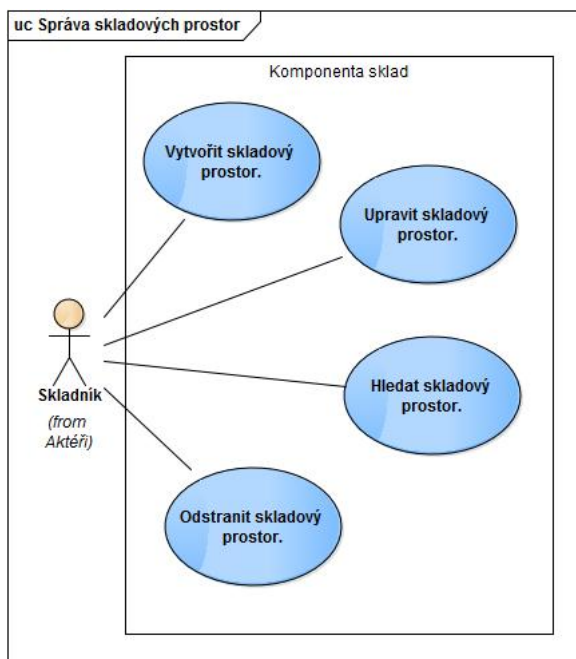


Obrázek 3.5: Případy užití - Kontrola kvality

### 3.1.7 Případy užití - Sklad

Komponenta sklad obsahuje případy užití, které se vážou ke správě skladových prostor a dále k samotnému skladovacímu procesu. Z hlediska podpory výroby se jedná o důležitou část systému, která by případy užití měla snížit některé nedostatky identifikované v kapitole 2.3.3.

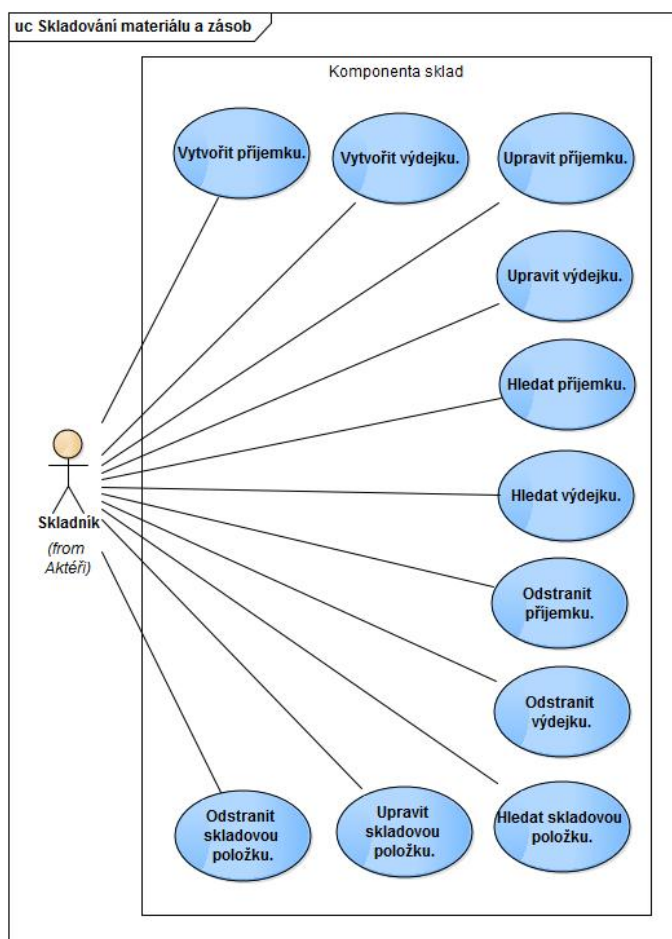
Správu skladových prostor znázorňuje diagram 3.6, ke kterému není třeba doprovodný komentář.



Obrázek 3.6: Případy užití - Správa skladových prostor

Skladování materiálu a výrobků zachycuje diagram 3.7. Z něj je patrné,

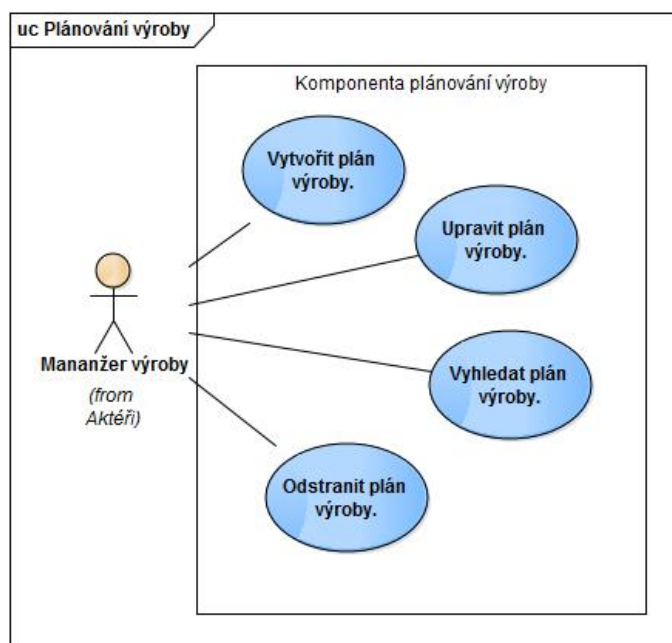
že pohyb dílů ze skladu do výroby a pohyb sestav z výroby do skladu je zachycen pomocí skladových dokumentů příjemky a výdejky. Pro samotný přehled stavu zásob na skladu slouží skladová položka, která je vytvořena při tvorbě příjemky na sklad a upravena při tvorbě výdejky. Platí, že každý díl může být na sklad přijat právě jednou a rovněž z něj vydán do výroby právě jednou. Pohyb zásob mezi sklady v tomto systému lze zachytit úpravou skladové položky. Není však vnímán jako podstatný pro podporu výroby.



**Obrázek 3.7:** Případy užití - Skladování materiálu a výrobků

### 3.1.8 Případy užití - Plánování výroby

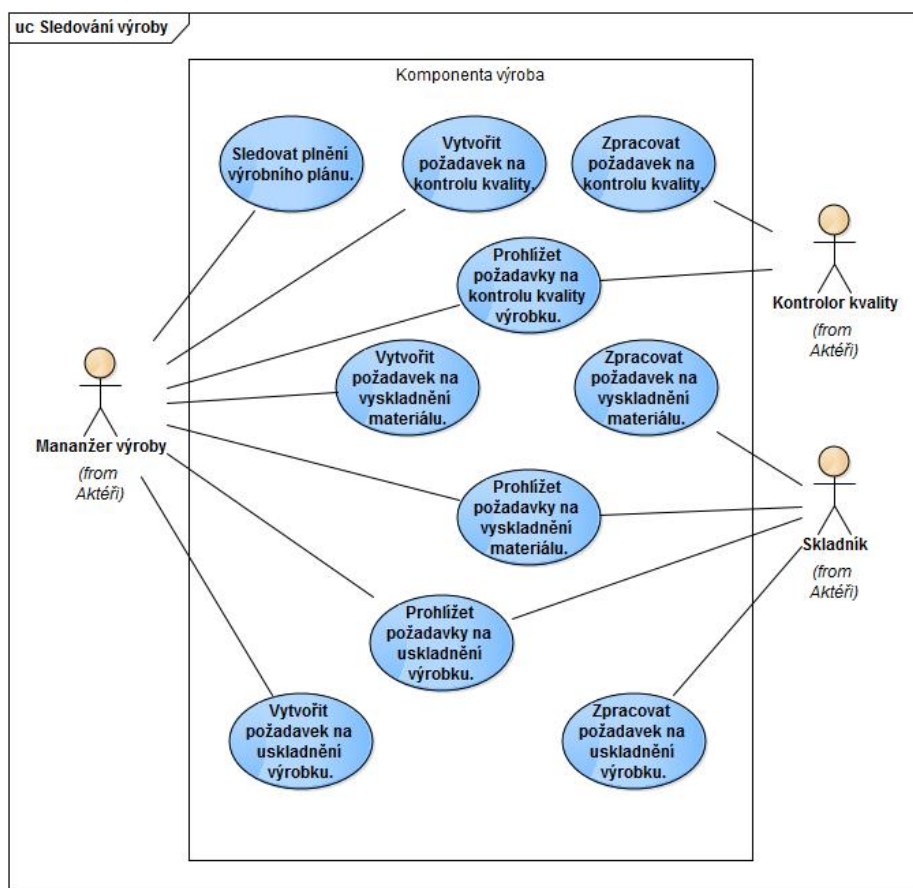
Diagram případů užití plánování výroby na obrázku 3.8 by svou stručností mohl navodit dojem, že nejde o důležitou část z pohledu podpory výroby. Opak je pravdou. V návrhu systému pro podporu výroby představuje tvorba výrobního plánu předpoklad pro plynulou výrobu, nejen z hlediska správného naplánování jednotlivých úkolů plánu, ale i informace o stavu materiálu pro vytvořený plán.



Obrázek 3.8: Případy užití - Plánování výroby

### 3.1.9 Případy užití - Výroba

Sledování výroby představuje diagram na obrázku 3.9. V této části systému dochází ke kooperaci tří aktérů, kteří zasahují do jednotlivých fází výrobního procesu. Manažer výroby sleduje a aktualizuje stav výrobního úkolu. Vytváří požadavky na vyskladnění materiálu ze skladu, kontrolu kvality dokončeného výrobku a jeho následné uskladnění. Kontrolor kvality a skladník tyto požadavky zpracovávají a vytváří tak propojení mezi plánem, výrobou a skladem. Dohromady tak vytváří obraz o stavu výroby, který je pro její podporu nutný.



Obrázek 3.9: Případy užití - Výroba

## 3.2 Scénáře případů užití

Scénář případu užití je způsob jak detailněji ilustrovat interakci aktéra či aktérů a systému v případě užití. Scénář podává informaci o počátečních podmínkách a aktérech vystupujících ve scénáři. Obsahuje hlavní scénář, který v sérii kroků popisuje interakce mezi aktéry a systémem, která vede k úspěšnému dosažení cíle případu užití. Dále obsahuje rozšíření scénáře o kroky, které mohou nastat v případech vedoucích k neúspěchu. Rovněž stanovuje, co v daném případě užití představuje úspěšné a neúspěšné dosažení cíle.

V následující části jsou uvedeny ukázky scénářů vytvořených pro navrhovaný systém podpory výroby. Pohledem na diagramy případů užití uvedených v předchozí části lze vyvodit, že některé případy užití jsou podobné, z tohoto důvodu byly pro ukázkou vybrány scénáře, které nejlépe vystihují podstatu navrhovaného systému. (Další scénáře případů užití jsou k dispozici v elektronické podobě na doprovodném médiu v souboru *ucscenare.pdf*)

### ■ 3.2.1 Scénář Vytvořit prototyp sestavy

**Aktér:** Technolog

**Počáteční podmínky:** aktér přihlášený do systému, na domovské stránce své role.

**Hlavní úspěšný scénář:**

1. Technolog zvolí možnost vytvořit nový prototyp sestavy.
2. Systém zobrazí formulář pro vytvoření prototypu sestavy.
3. Technolog vyplní id prototypu.
4. Systém zkontroluje validitu id.
5. Systém zkontroluje, zda je id volné k použití v seznamu prototypů dílů.
6. Technolog vyplní zbytek souhrných údajů ve formuláři.
7. Systém zkontroluje validitu vyplněných dat.
8. Technolog vybere možnost přidat kusovník.
9. Systém nabídne seznam existujících prototypů dílů.
10. Technolog vybere díl z nabídky a vloží jej do kusovníku.
11. Systém vloží díl do seznamu kusovníku.
12. Technolog uzavře kusovník.
13. Systém vytvoří prototyp sestavy s kusovníkem

**Rozšíření scénáře:**

- 4a Systém zjistí nevalidní id a zobrazí varování.
- 8b. Technolog nevybere možnost vytvořit kusovník.
- 9b. Systém upozorní na nutnost vytvořit kusovník.
- 10c. Technolog nechá kusovník prázdný.
- 11c. Systém nedovolí uzavřít kusovník.
- 12c. Systém nevytvoří prototyp sestavy s kusovníkem.

**Konečný úspěšný stav:** vytvořen nový prototyp sestavy s kusovníkem.

**Konečný neúspěšný stav:** nevytvořen nový prototyp sestavy.

### ■ 3.2.2 Scénář Vytvořit plán výroby

**Aktér:** Manažer výroby

**Počáteční podmínky:** aktér přihlášen do systému, na domovské stránce přihlášené role.

**Hlavní úspěšný scénář:**

1. Manažer výroby zvolí možnost vytvořit nový plán výroby.
2. Systém zobrazí formulář pro souhrnné informace plánu výroby.
3. Manažer výroby vyplní souhrnné informace plánu výroby.
4. Systém zkontroluje validitu vyplňovaných údajů.
5. Manažer výroby odešle formulář souhrnných informací.
6. Systém zkontroluje správnost odeslaných dat (neduplicitní data, správný formát).

7. Systém vytvoří nový otevřený plán výroby.
8. Systém zobrazí informaci o úspěšném vytvoření prázdného nového plánu výroby.
9. Systém zobrazí formulář pro přidání nového úkolu plánu.
10. Manažer výroby výroby vyplní formulář nového úkolu plánu.
11. Systém provede validaci zadaných údajů.
12. Manažer výroby odešle formulář nového úkolu plánu.
13. Systém zkontroluje správnost odeslaných dat.
14. Systém vytvoří nový úkol plánu.
15. Systém zobrazí seznam všech úkolů plánu pro tento plán výroby.
16. Manažer výroby ukončí vytváření plánu.

#### Rozšíření scénáře:

- 4a. Systém zjistí nevalidní informace.
- 5a. Systém zablokuje odeslání formuláře.
- 6a. Návrat ke kroku 3.
- 10b. Manažer výroby nevytvoří krok výroby. Scénář končí úspěchem s prázdným otevřeným plánem.
- 16c. Manažer výroby pokračuje vložением dalšího úkolu návratem k bodu 10 hlavního scénáře.

**Konečný úspěšný stav:** vytvořen nový otevřený plán výroby.

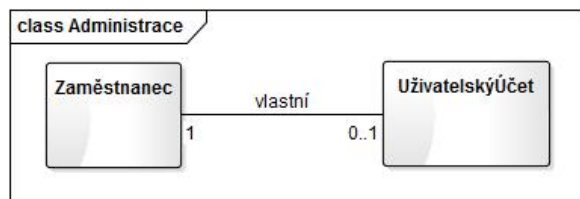
**Konečný neúspěšný stav:** nevytvořen nový plán výroby.

## 3.3 Doménový model

Tak jako případy užití i doménový model navrhovaného systému je rozdělen do komponent, které se týkají podnikových a podpůrných procesů.

### 3.3.1 Administrace

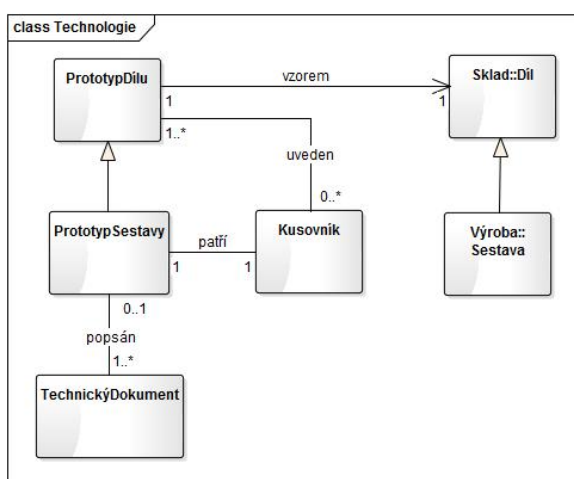
Velice jednoduchý doménový model administrace na obrázku 3.10 zachycuje vztah mezi zaměstnancem a uživatelským účtem. Zaměstnanci vedení v systému nemusí vlastnit uživatelský účet, který naopak musí být vždy vlastněn právě jedním zaměstnancem.



Obrázek 3.10: Doménový model - Administrace

### 3.3.2 Technologie

Doménový model technologie na diagramu 3.11 zachycuje vztahy mezi entitami, které mají z hlediska systému pro podporu výroby podpůrný význam. Tvoří strukturu nutnou pro zachycení složení jednotlivých sestav pomocí kusovníků a přidružuje k nim technické dokumenty. Z diagramu je patrné, že sestava může být na kusovníku další sestavy jako díl. Tím je vytvořena stromová struktura, která má pomoci při plánování sestav a podsestav. Dále stojí za povšimnutí vztah mezi prototypem dílu a dílem.

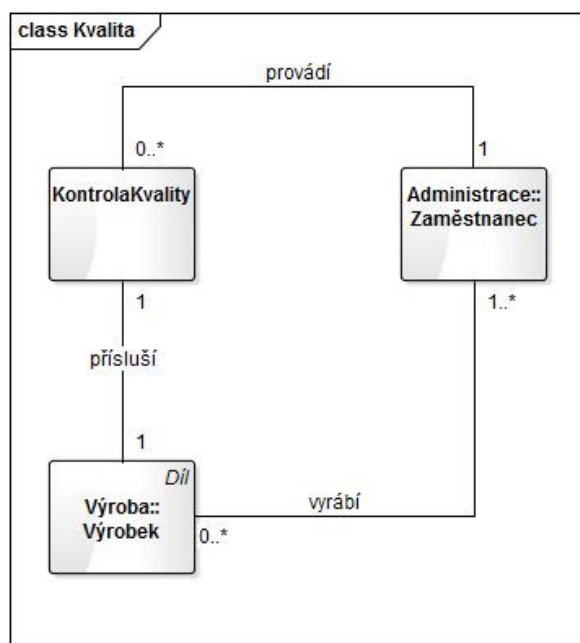


**Obrázek 3.11:** Doménový model - Technologie

### 3.3.3 Kvalita

Kontrola kvality, zachycená na diagramu 3.12, přiřazuje výrobku zhotovitele a kontrolora.

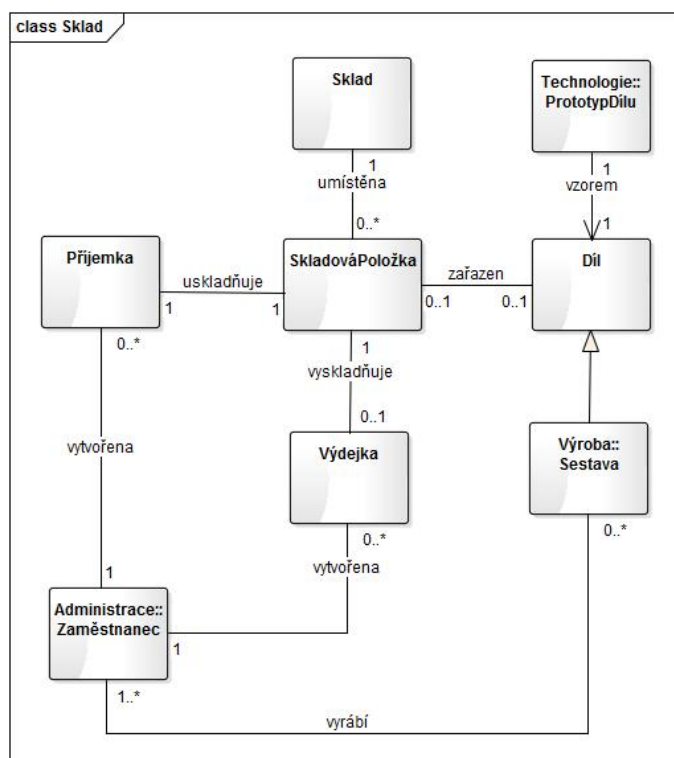




Obrázek 3.12: Doménový model - Kvalita

### 3.3.4 Sklad

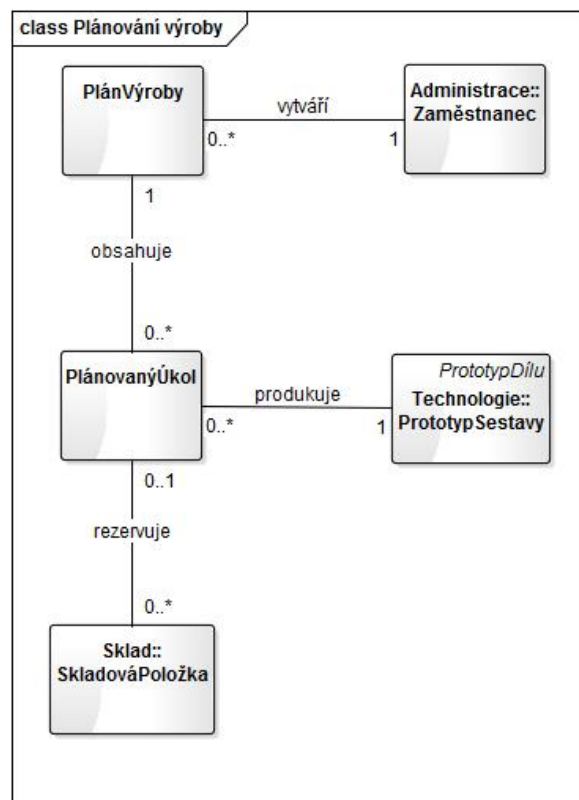
Doménový model skladování na diagramu 3.13 zachycuje entity a jejich vztahy vznikající při procesu skladování. Při pohledu na diagram by se mohlo zdát, že skladová položka je nadbytečná a její roli by mohl zastoupit díl. Tento způsob řešení bere v potaz fakt, že ne každý díl je nutné uskladnit. Vyrobená sestava, která je dílem pro následující výrobní úkol tak nemusí být mezi těmito výrobními kroky fiktivně uskladněna. Dále lze z modelu usoudit, že uskladnění a vyskladnění se uskutečňuje prostřednictvím příjemky a výdejky. To znamená, že lze zpětně dohledat tok materiálu skladem, což podporuje princip dohledatelnosti zmíněný na počátku analýzy.



Obrázek 3.13: Doménový model - Skladování

### 3.3.5 Plánování

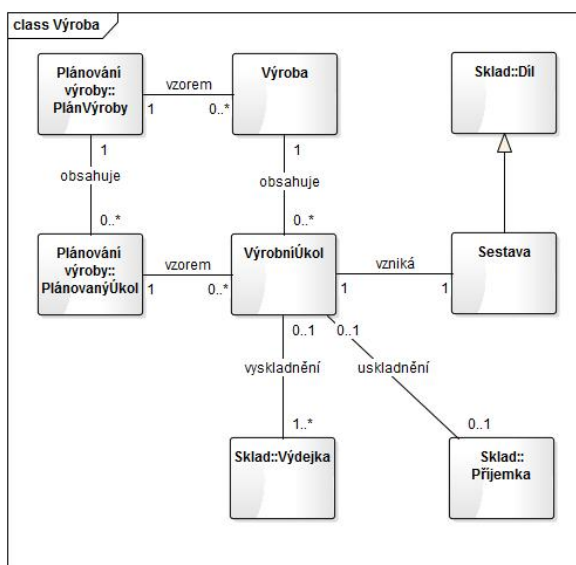
Na doménovém modelu plánování na diagramu 3.14 stojí za povšimnutí vztah mezi plánovaným úkolem a prototypem sestavy. Z hlediska plánování není v době tvorby plánu důležité, aby předmět plánovaného úkolu nesl informace o konkrétním výrobku, proto je vhodnější prototyp sestavy, který nese skrze kusovník informace potřebné k rezervaci dílů na skladě představovanými skladovou položkou. Díky tomuto řešení lze jeden plán použít jako vzor pro více výrobních cyklů.



Obrázek 3.14: Doménový model - Plánování výroby

### 3.3.6 Výroba

Doménový model výroby na diagramu 3.15 ukazuje vztah mezi plánem výroby a probíhající výrobou. Za povšimnutí stojí provázání výdejky materiálu s výrobním úkolem, na který je určen a příjmkou na sklad, která dokumentuje uskladnění dokončené sestavy.



Obrázek 3.15: Doménový model - Výroba

## 3.4 Systémové požadavky

Systémové požadavky jsou detailnějším popisem uživatelských požadavků. Mohou sloužit k návrhu systému a proto musí kompletní a konsistentní specifikací celého systému.[1] Ty se dále dělí na požadavky, které definují funkce systému a nazývají se funkční nebo též funkční požadavky. Ostatní požadavky spadají do kategorie lapidárně nazývané nefunkčních požadavků, které definují parametry a omezení navrhovaného systému.

### 3.4.1 Funkční požadavky

Funkcionalita navrhovaného systému je definována 60 funkčními požadavky. Pro přehlednost jsou požadavky rozděleny do skupin odpovídajících rozdělení případů užití z předchozí části 3.1.2.

## ■ Společné

ID	Funkční požadavek
FRQ1	Systém musí vyžadovat autentikaci uživatele pomocí uživatelského jména a hesla pro platný uživatelský účet.
FRQ2	Systém musí upozornit uživatele na neplatnou kombinaci uživatelského jména a hesla.
FRQ3	Systém musí uživateli umožnit odhlásit se.
FRQ4	Systém musí uživateli umožnit změnu hesla pro uživatelský účet, na kterém je přihlášen.

**Tabulka 3.1:** Funkční požadavky - společné

## ■ Administrace

ID	Funkční požadavek
FRQ5	Systém musí umožnit administrátorovi vytvořit nový uživatelský účet.
FRQ6	Systém musí umožnit administrátorovi změnit veškeré údaje spojené s uživatelským účtem, který není aktivní.
FRQ7	Systém musí umožnit administrátorovi změnit status účtu z aktivního (lze se na něj přihlásit) na neaktivní (nelze se přihlásit).
FRQ8	Systém musí umožnit administrátorovi vyhledat uživatelský účet podle zvoleného klíče z množiny {uživatelské jméno, příjmení}.
FRQ9	Systém musí umožnit administrátorovi odstranit uživatelský účet, který není aktivní a nepoužíval se (jako opravu pro chybné vytvoření účtu).

**Tabulka 3.2:** Funkční požadavky - administrace

## ■ Technologie

ID	Funkční požadavek
FRQ10	System musí umožnit technologovi vytvořit nový záznam o technické dokumentaci a uložit ho do katalogu.
FRQ11	System musí umožnit technologovi hledat záznamy o technické dokumentaci podle klíče z množiny {identifikátor dokumentu, název dokumentu, kategorie dokumentu, umístění dokumentu}.
FRQ12	System musí umožnit technologovi upravit veškeré údaje pro vybraný záznam.
FRQ13	System musí umožnit technologovi odstranit vybraný záznam z katalogu.
FRQ14	System musí umožnit technologovi vytvořit nový prototyp dílu.
FRQ15	System musí umožnit technologovi vytvořit nový prototyp sestavy včetně kusovníku.
FRQ16	System musí umožnit technologovi potvrdit kompletnost sestavy a jejího kusovníku.
FRQ17	System musí umožnit uživateli vyhledat prototyp dílu podle klíče z množiny {identifikační číslo, sériové číslo, český název, anglický název}
FRQ18	System musí umožnit technologovi odstranit prototyp z katalogu, který nebyl nikdy použit.
FRQ19	System musí umožnit technologovi zneplatnit prototyp, který tak nebude nadále používán pro výrobu, ale zůstane po něm záznam.
FRQ20	System musí umožnit technologovi upravit všechny položky zvoleného prototypu.
FRQ21	System musí umožnit technologovi upravit všechny položky zvoleného prototypu sestavy včetně úpravy kusovníku.

**Tabulka 3.3:** Funkční požadavky - technologie

## ■ Skladování

ID	Funkční požadavek
FRQ22	Systém musí umožnit skladníkovi vytvořit nový skladový prostor.
FRQ23	Systém musí umožnit skladníkovi hledat skladový prostor podle klíče z množiny {název, lokace}.
FRQ24	Systém musí umožnit skladníkovi upravit veškeré údaje skladového prostoru.
FRQ25	Systém musí umožnit skladníkovi odstranit prázdný skladový prostor.
FRQ26	Systém musí umožnit skladníkovi zobrazit seznam skladových prostor.
FRQ27	Systém musí umožnit skladníkovi vybrat prototyp pro dodaný materiál.
FRQ28	Systém musí umožnit skladníkovi určit skladový prostor pro uskladnění materiálu.
FRQ29	Systém musí umožnit skladníkovi vytvořit skladovou položku.
FRQ30	Systém musí umožnit skladníkovi hledat skladovou položku podle klíče z množiny {identifikační číslo, sériové číslo, název}.
FRQ31	Systém musí umožnit skladníkovi upravit skladovou položku.
FRQ32	Systém musí umožnit skladníkovi odstranit skladovou položku jen v případě, pokud nebyla položka již vyskladněna do výroby nebo k expedici.
FRQ33	Systém musí umožnit skladníkovi vybrat požadavek na vyskladnění materiálu.
FRQ34	Systém musí umožnit skladníkovi vytvořit na základě vybraného požadavku výdejku materiálu ze skladu.
FRQ35	Systém musí uložit datum vyskladnění materiálu uvedeného na výdejce.
FRQ36	Systém musí umožnit skladníkovi vybrat požadavek na uskladnění vyrobené sestavy.
FRQ37	Systém musí umožnit skladníkovi vytvořit na základě požadavku příjemku do skladu.
FRQ38	Systém musí umožnit skladníkovi vytvořit novou skladovou položku na základě příjemky do skladu.
FRQ39	Systém musí umožnit skladníkovi vytvořit výdejku pro expedici výrobku.
FRQ40	Systém musí umožnit skladníkovi prohlížet výdejky.
FRQ41	Systém musí umožnit skladníkovi prohlížet příjemky.
FRQ42	Systém musí umožnit skladníkovi upravit výdejku.
FRQ43	Systém musí umožnit skladníkovi upravit příjemku.
FRQ44	Systém musí umožnit skladníkovi odebrat výdejku.
FRQ45	Systém musí umožnit skladníkovi odebrat příjemku.

**Tabulka 3.4:** Funkční požadavky - skladování

### Kontrola kvality

ID	Funkční požadavek
FRQ46	System musí umožnit kontrolorovi vybrat požadavek na kontrolu kvality výrobku.
FRQ47	System musí umožnit kontrolorovi vytvořit záznam o kontrole kvality výrobku uvedeném v požadavku.
FRQ48	System musí umožnit kontrolorovi hledat záznam o kontrole kvality podle klíče z množiny {id sestavy, data kontroly, výsledku kontroly}.
FRQ49	System musí umožnit kontrolorovi upravit záznam o kontrole kvality.
FRQ50	System musí umožnit kontrolorovi odstranit záznam o kontrole kvality a to pouze u výrobku, který nebyl uskladněn, nebo odeslán dále do výroby.

**Tabulka 3.5:** Funkční požadavky - kontrola kvality

### Plánování výroby

ID	Funkční požadavek
FRQ51	System musí umožnit manažerovi výroby zadat souhrnné informace o plánu výroby.
FRQ52	System musí umožnit manažerovi výroby přidat úkol do výrobního plánu, úkolem se rozumí požadavek na výrobu sestavy.
FRQ53	System musí umožnit manažerovi výroby odebrat úkol výrobního plánu a to vždy spolu se všemi úkoly, které jsou na odebíraném plánu závislé.
FRQ54	System musí umožnit manažerovi výroby kontrolu dostupného materiálu na skladu.
FRQ55	System musí umožnit manažerovi výroby naplánovat výrobu podsestav, které nejsou dostupné jako materiál pro výrobu na skladu.
FRQ56	System musí umožnit manažerovi výroby hledat plán výroby podle klíče z množiny {id plánu, datum zahájení plánu}.
FRQ57	System musí umožnit manažerovi výroby upravit plán výroby, včetně úprav úkolů plánu.
FRQ58	System musí umožnit manažerovi výroby odstranit neuzavřený výrobní plán.
FRQ59	System musí umožnit manažerovi výroby schválit uzavřený plán pro výrobu. Tímto krokem se schválený plán stává výrobním plánem zařazeným za právě probíhající výrobní plán.

**Tabulka 3.6:** Funkční požadavky - plánování výroby



 Výroba

ID	Funkční požadavek
FRQ60	System musí umožnit manažerovi výroby aktualizovat údaje u nedokončených úkolů.
FRQ61	System musí umožnit manažerovi výroby ukončit úkol výrobního plánu.
FRQ62	System musí umožnit manažerovi výroby vytvořit požadavek na kontrolu kvality pro ukončený výrobní úkol.
FRQ63	System musí umožnit manažerovi výroby vytvořit požadavek na vyskladnění materiálu pro úkol výroby.
FRQ64	System musí umožnit manažerovi výroby vytvořit požadavek na uskladnění dokončeného a zkontrolovaného výrobku.
FRQ65	System musí umožnit kontrolorovi a manažerovi výroby prohlížet požadavky na kontrolu kvality.
FRQ66	System musí umožnit skladníkovi a manažerovi výroby prohlížet požadavky na vyskladnění materiálu.
FRQ67	System musí umožnit skladníkovi a manažerovi výroby prohlížet požadavky na uskladnění výrobku.
FRQ68	System musí umožnit kontrolorovi kvality zpracovat požadavek na kontrolu kvality a po dokončení změnit jeho stav na uzavřený.
FRQ69	System musí umožnit skladníkovi zpracovat požadavek na vyskladnění materiálu a po dokončení změnit jeho stav na uzavřený.
FRQ70	System musí umožnit skladníkovi zpracovat požadavek na uskladnění výrobku a po jeho dokončení změnit jeho stav na uzavřený.

**Tabulka 3.7:** Funkční požadavky - výroba

### 3.4.2 Nefunkční požadavky

Navrhovaný systém musí splňovat nefunkční požadavky uvedené v tabulce 3.8.

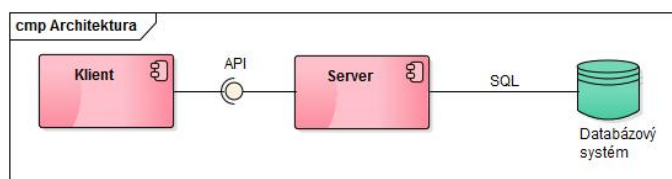
ID	Požadavek
NFR1	Všechny části systému musí být schopny provozu na OS Windows 7.
NFR3	Systém musí být schopen provozu na vnitropodnikové síti.
NFR4	Systém nesmí vytvářet finanční náklady v podobě plateb za softwarové licence třetím stranám.
NFR5	Systém musí být schopen provozu bez odstávek minimálně 12 hodin denně v obvyklé pracovní době od 6:00 do 18:00
NFR7	Systém musí být schopen používat českou znakovou sadu a uživatelské rozhraní musí být popsáno v češtině.
NFR8	Systém musí být chráněn proti uživatelům bez platného účtu.

Tabulka 3.8: Nefunkcionální požadavky.

## 3.5 Architektura systému

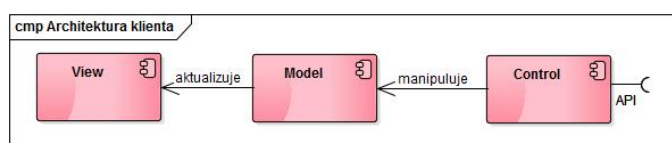
Systémové požadavky formulované v předchozí části lze využít při návrhu architektury systému. Z požadavků je patrné, že systém musí podporovat práci několika uživatelů najednou, v některých částech systému má tato práce rys kooperace více uživatelů. Znamená to, že systém mezi uživateli sdílí data. Nejblíže výše popsanému odpovídá architektura klient-server webové podnikové aplikace.

Celkový pohled na navrhovaný systém nabízí diagram 3.16. Z něj je patrná základní struktura systému. Následující diagram 3.17 ukazuje jednoduchou



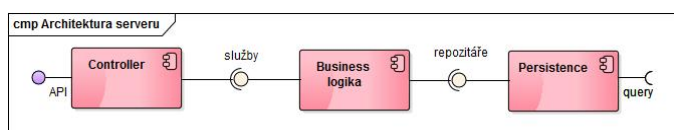
Obrázek 3.16: Architektura systému - pohled na celek

strukturu klienta, navrženou v souladu se známým archetypem MVC. Klient slouží primárně jako uživatelské rozhraní s minimem logiky umístěné v části Control, která je rovněž zodpovědná za komunikaci se serverem.



Obrázek 3.17: Architektura systému - komponenty klienta

Serverová část se skládá z komponent Controlleru, Business logiky a Perzistence. Jak ukazuje diagram na obrázku 3.18 Controller vystavuje aplikační rozhraní ro komunikaci s klientem a využívá rozhraní služeb, které nabízí komponenta business logiky. V ní je implementována business logika pro procesy administrace, skladování, kontrolu kvality, plánování výroby a sledování výroby. Komponenta business logiky využívá rozhraní komponenty Perzistence pro ukládání dat v databázovém systému, jak je patrné z celkového pohledu na obrázku 3.16.



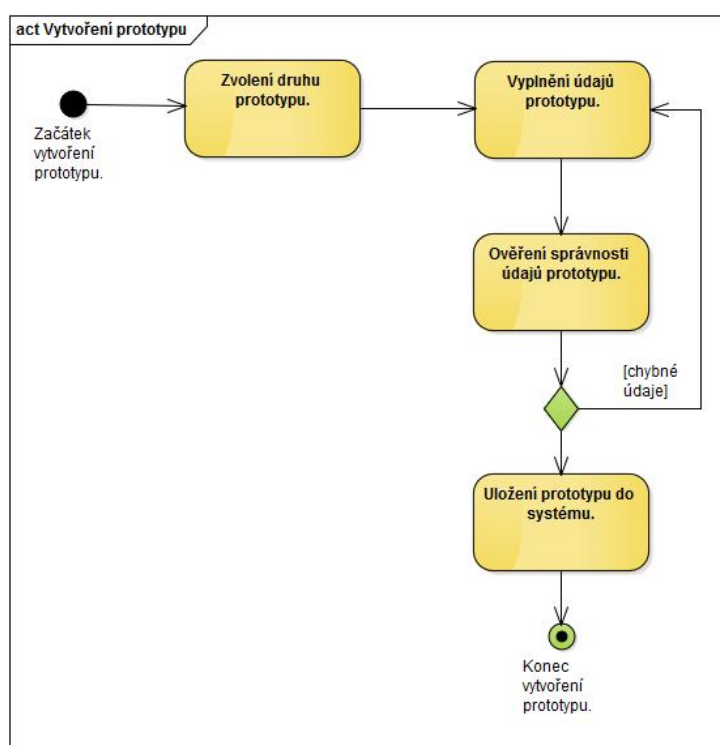
Obrázek 3.18: Architektura systému - komponenty serveru

## 3.6 Modely procesů

Při návrhu systému je důležité nezabývat se pouze strukturou systému, ale i jeho dynamikou. Tu nejlépe ilustrují modely systémových procesů. S ohledem na rozsah navrhovaného systému není možné na tomto místě ukázat všechny procesy. Proto byly zvoleny ukázky tří procesů, které představují charakteristické operace v systému podpory výroby. Více diagramů systémových procesů lze nalézt na přiloženém médiu v souboru *bprace.ea*.

### 3.6.1 Model procesu vytvoření prototypu

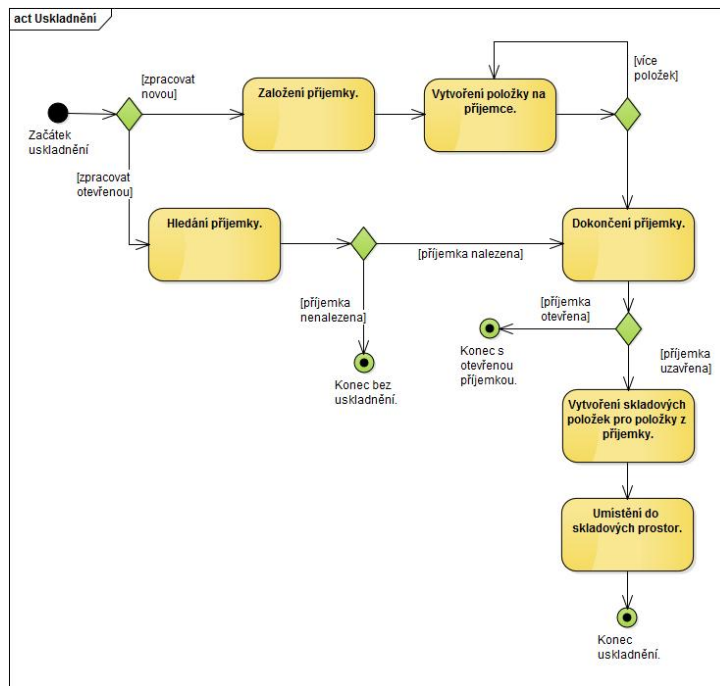
Proces vytvoření prototypu přísluší komponentě Technologie a je zachycen na diagramu 3.19. Pracuje s entitou PrototypDílu.



**Obrázek 3.19:** Model procesu - vytvoření prototypu

### ■ 3.6.2 Model procesu uskladnění dílu

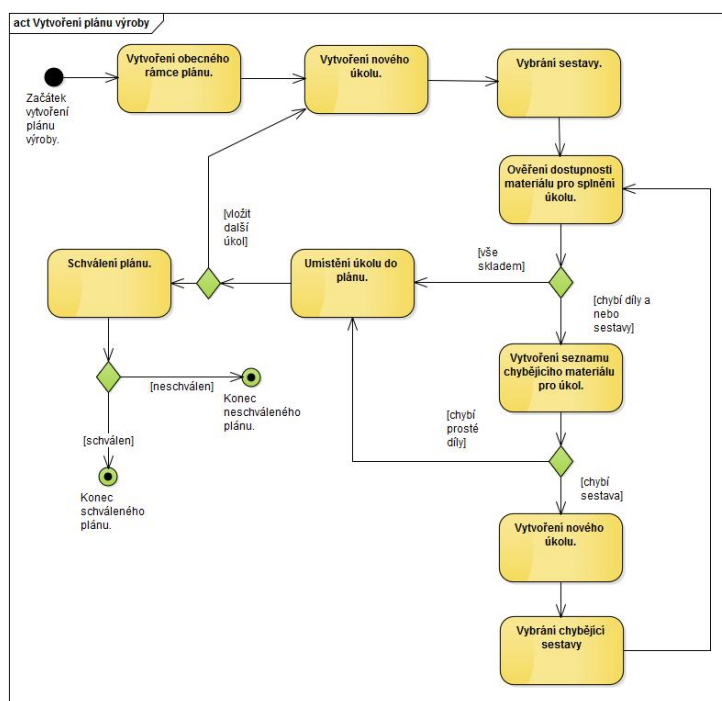
Proces uskladnění dílu přísluší komponentě Sklad a spočívá ve tvorbě příjemky a v následném vytvoření skladových položek podle počtu dílů na příjemce. Příjemku není nutné dokončit, v takovém případě však nedojde k uskladnění dílu. Nedokončené příjemky lze dokončit později. To umožňuje například vytvářet otevřené příjemky po ukončení výrobního úkolu pro vyrobenou sestavu v komponentě Výroba. Následným zpracováním takovéto příjemky dojde k uskladnění vyrobené sestavy. Výše zmíněné ilustruje diagram 3.20.



Obrázek 3.20: Model procesu - uskladnění dílu

### 3.6.3 Model procesu vytvoření plánu

Proces vytvoření plánu výroby patří do komponenty Plánování. Za zmínku stojí část plánování, ve které uživatel přidává úkoly plánu se sestavou jako předmětem výroby. Po kontrole dostupnosti dílů je možné naplánovat chybějící podsestavy jako další výrobní úkol, na kterém je původní úkol s chybějící podsestavou závislý. Celkový pohled na proces nabízí diagram 3.21.



Obrázek 3.21: Model procesu - vytvoření plánu

## Kapitola 4

### Implementace

Tato část se zabývá implementací prototypu systému navrženého na základě poznatků shromážděných v předchozích kapitolách. Jsou zde diskutovány použité technologie doprovázené ukázkami jejich užití při implementaci částí systému.

Po důkladné analýze a vypracování návrhu je patrné, že systém je velmi rozsáhlý a komplexní, zejména z pohledu provázanosti jednotlivých datových entit, a proto by jeho kompletní implementace ve stanoveném termínu nebyla možná. Po dohodě s vedoucí práce bylo rozhodnuto implementovat pouze část systému pro podporu výroby, jmenovitě komponenty administrace, technologie a skladování. Tak je stále možné na prototypu demonstrovat alespoň částečnou použitelnost při podpoře výroby.

#### 4.1 Použité technologie

Přehled použitých technologií při implementaci lze rozčlenit na použité programovací jazyky, knihovny a frameworky a podpůrné aplikace.

##### 4.1.1 Programovací jazyky

Při realizaci návrhu bylo třeba vzít v úvahu, který programovací jazyk nabízí nejvíce možností, jak při implementaci dosáhnout cíle efektivně a zároveň dodržet systémové požadavky. To znamená jak existenci množství knihoven založených na tomto jazyku, tak i širokou bázi znalostí v podobě dostupné dokumentace či znalostí uživatelů, která by mohla pomoci řešit případné problémy vzniklé při implementaci. Z tohoto důvodu byla zvolena dvojice jazyků Java a Javascript, které se dlouhodobě drží v popředí zájmu mezi vývojáři zejména webových podnikových aplikací. Jejich všeobecnou oblíbenost je možné doložit například na statistice společnosti TIOBE<sup>1</sup>

<sup>1</sup>TIOBE Index for May 2019. Tiobe [online]. [cit. 2019-05-23]. Dostupné z: <https://www.tiobe.com/tiobe-index/>

### 4.1.2 Knihovny a frameworky

Na straně serveru byl použit framework Spring, zejména jeho části Spring Boot, která umožňuje spouštět aplikace bez nutnosti je nahrávat na aplikační server, obsahuje totiž embedded Tomcat servlet kontejner. Dále Spring Data JPA, který umožňuje snadné vytváření repozitářů entitních tříd skrze rozhraní, která tato knihovna pomocí proxy tříd instancuje a tím vývojáři šetří čas strávený při psaní tzv. boilerplate kódu.

Na straně klienta jsou použity knihovny postavené na jazyku Javascript a prostředí Node.js. Jde o knihovnu pro návrh uživatelského rozhraní React.js, dále knihovku Redux, která udržuje stav aplikace, knihovnu react router, která umožňuje aplikaci vystupovat jako Single page application a knihovnu axios, která slouží pro vyřizování asynchronních požadavků klienta na server. Při implementaci byla využívána kniha[2] věnovaná těmto knihovnám.

### 4.1.3 Podpůrné aplikace

Důležitou součástí implementace je databázový systém PostgreSQL, který spoluzajišťuje perzistenci ukládáním dat systému do relační databáze. Byl vybrán na základě dostupné licence a široké znalostní bázi, opět v podobě množství uživatelů a dostupné dokumentaci.

## 4.2 Implementace serveru

Implementace serverové části aplikace spočívá v implementaci tříd představujících controller, service a repository. Vše ostatní zajišťuje framework Spring.

Controller zpracovává požadavek, pro který na základě mapování cest na jednotlivé metody určí další zpracování, například transformací příchozích dat z JSON formátu do Java objektu, který dále předá třídě implementující service. Následující kód je ukázkou controlleru. Za povšimnutí stojí konfigurace četnými anotacemi. Metoda findAll zpracovává požadavek přicházející na adresu zmíněnou nad hlavičkou třídy. Ostatní metody mají cestu vždy v anotaci nad svou definicí.

```
@RestController
@RequestMapping("/api/technology/prototypeparts")
public class PrototypePartController {

    private Logger logger = LoggerFactory.getLogger(
        PrototypePartController.class);
    @Autowired
    private PrototypePartService prototypePartService;

    @GetMapping(produces = "application/json;charset=UTF-8")
    public HttpEntity<Iterable<PrototypePart>> findAll(){
```



```

return new HttpEntity<Iterable<PrototypePart>>(
prototypePartService.findAll());
}
.
.
.//

```

Service představuje místo implementace business logiky aplikace. V navrhovaném systému je zodpovědá za komunikaci s perzistentní vrstvou, implementovanou jako repository. Následující fragment kódu výše zmíněné ilustruje.

```

@Service
public class PrototypePartService {

private PrototypePartRepository prototypePartRepo;

public PrototypePartService(
PrototypePartRepository prototypePartRepo) {
super();
this.prototypePartRepo = prototypePartRepo;
}

public Iterable<PrototypePart> findAll() {
return prototypePartRepo.findAll();
}

public PrototypePart createPrototypePart(
PrototypePart prototypePart) {
if(prototypePart != null)
return prototypePartRepo.save(prototypePart);
return null;
}
...

```

Repository je díky Spring Data JPA definované jako rozhraní, které je následně frameworkem pomocí proxy tříd instancováno a to pokud není explicitně uvedena vlastní implementace tohoto rozhraní. Toto řešení má mnoho výhod od úspory času, kdy není nutné psát opakující se kód (dry), po skutečnost, že framework je schopen parsovat názvy metod rozhraní a z nich pak vytvářet dotazy při komunikaci s databází. To dokládá níže uvedený kód.

```

public interface PrototypePartRepository
extends CrudRepository <PrototypePart, Long>{
public PrototypePart findByPartIDEquals(String partId);
public List<PrototypePart> findByNameCZLike(String nameCZ);
public List<PrototypePart> findByNameENLike(String nameEN);
}

```



# Kapitola 5

## Testování

Charakteristik testování je mnoho, stručnou a výstižnou nabízí například [1]. Podle něj je testování softwaru je součástí procesu validace a verifikace. Zahrnuje spouštění implementovaného systému s testovacími daty a vyhodnocováním výstupu. V této práci byly při vývoji použity jednotkové testy, taktéž zvané unit testy, a integrační testy.

### 5.1 Jednotkové testy

Při jednotkových testech dochází k testování nejmenších funkčních celků programu. U objektového programování jde zpravidla o testování metod jednotlivých tříd. Testovaný objekt je izolován od okolí a potřebné závislosti pro vykonání testu se emulují pomocí tzv. mocků, u kterých je předem definované chování.

Příklad jednotkového testu je na následujícím výpisu kódu. Z důvodů formátování jsou vynechány importy.

```
@RunWith(SpringRunner.class)
@DataJpaTest
public class EmployeeRepositoryTest {

    @Autowired
    EmployeeRepository employeeRepo;

    @Autowired
    TestEntityManager testEntityManager;

    @Test
    public void whenFindByLastName_ThenReturnEmployees() {
        //given
        Employee employee1 = new Employee();
        employee1.setFirstName("John");
        employee1.setLastName("Doe");
        testEntityManager.persist(employee1);
        testEntityManager.flush();
    }
}
```

```

//when
List<Employee> foundEmployees = employeeRepo.findByLastName("Doe");
Employee oneEmployee = foundEmployees.get(0);

//then
assertThat(oneEmployee.getLastName()).
isEqualTo(employee1.getLastName());
assertThat(oneEmployee.getLastName()).
isEqualTo("Doe");
}

```

Tento jednotkový test se týká repozitáře. Testuje uživatelem definovanou funkci vyhledání uživatele podle příjmení (lastname). Na výpisu stojí za povšimnutí mock objekt `TestEntityManager`, který slouží k vytvoření výchozí situace, kdy je do in-memory databáze uložen objekt, který se testovaný repozitář pokusí správně získat. K tomuto slouží poněkud zřetěžená konstrukce `assertThat` na konci výpisu.

## 5.2 Integrované testy

Integrované testy, jak název napovídá, integrují více testovaných komponent. Integrace komponent totiž může ústít v problémy, které při interakci částí systému mohou nastat. [1]

Následující ukázka dokumentuje, že integrovaný test již využívá minimum mock objektů. Většinou pouze na vstupu testu.

```

@RunWith(SpringRunner.class)
@WebMvcTest(EmployeeController.class)
public class EmployeeIntegrationTest {

    @Autowired
    MockMvc mockMvc;

    @Test
    public void performGet..._ThenGetJSONResult() {
        try {
            mockMvc.perform(
                MockMvcRequestBuilders.get("/api/admin/employees").
                    contentType(
                        MediaType.APPLICATION_JSON). andExpect(MockMvcResultMatchers.
                            status().
                                isOk()).
                    andExpect(MockMvcResultMatchers.jsonPath("$.employee").
                        exists());
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}

```

```
}
```

```
}
```

```
}
```

V tomto integračním testu je controlleru podstrčen požadavek GET `/api/admin/employees`, na který má systém odpovědět výpisem všech zaměstnanců v podobě pole JSON objektů. Tuto skutečnost se snaží zachytit příkaz, který bylo nutné z hlediska nedostatku mít rozkouskovat na řádky.

Při přípravě testů se řešitel opíral mimo jiné o dokumentaci frameworku Spring<sup>1</sup> a o další inspiraci se postaral web Baeldung<sup>2</sup>, který se této problematice také věnuje.

---

<sup>1</sup>[www.spring.io](http://www.spring.io)

<sup>2</sup><https://www.baeldung.com/spring-boot-testing>



# Kapitola 6

## Závěr

V závěrečné části této práce je na místě zhodnotit úspěch a dopady projektu a to jak z pohledu zadavatele, tak i z pohledu řešitele. Dále je nabídnut i výhled do budoucna v podobě možných vylepšení systému.

### 6.1 Hodnocení projektu

V analytické části se podařilo odhalit kritická místa procesů výroby firmy Eratipa. Byla navržena řešení, která odpovídala zejména ekonomickým možnostem zadavatele. Zvolena byla varianta návrhu vlastního nového systému, který by zadavatel mohl s minimálními náklady provozovat.

Potřeby zadavatele vyjádřené jeho business požadavky reflektuje návrh systému, který je dále podrobně rozpracoval až na úroveň systémových požadavků.

Méně úspěšnou část projektu představuje implementace, kdy se podařilo vytvořit pouze částečně funkční prototyp systému. To bylo způsobeno zejména rozsáhlostí zadání a množstvím času, který na řešení zbýval.

Z pohledu řešitele byla přínosná možnost projít si celým vývojovým cyklem při tvorbě softwaru a ověřit si tak znalosti nabyté při studiu softwarového inženýrství. Následná implementace umožnila blíže se seznámit s technologiemi používanými při vývoji tohoto druhu aplikací jako jsou například části frameworku Spring a knihovny React.js.

### 6.2 Možná vylepšení systému

Vylepšení systému pro podporu výroby se mohou ubírat jak ve směru dalších úprav návrhu systému tak i jeho další implementace.

Z pohledu návrhu systému by bylo možné doplnit například komponentu plánování výroby o možnost generovat dokument objednávky chybějícího materiálu pro plánovaný úkol výroby. Dále by bylo možné automatizovat plánování podsestav v jednotlivých úkolech, tak aby se snížilo riziko, že plánovač udělá při sestavování plánu výroby chybu.

Z pohledu implementace se možné vylepšení nevyhnutelně týká dokončení zbývajících komponent plánování výroby, sledování výroby a sledování kvality.







## Literatura

- [1] I. Sommerville, *Software Engineering, Global Edition*. Pearson Education Limited, 2016.
- [2] A. Banks and E. Porcello, *Learning React*. Sebastopol, CA: O'Reilly Media, 1st edition ed., 2017.



# Příloha A

## Obsah doprovodného optického disku

Vše co bylo vytvořeno spolu s projektem a zároveň by se nevešlo do tištěné verze je umístěno na přiloženém disku. Adresářová struktura přílohy:

- thesis - adresář bakalářské práce, napsané pomocí latexu.
- eaproj - adresář s projektem v enterprise architectu - diagramy.
- sypovyr - serverová část aplikace, maven projekt.
- uimocks - návrhy obrazovek realizované na úrovni komponent v twitter bootstrap4
- uimocksbs3 - starší verze téhož
- ucscenare.odt - soubor dalších uživatelských scénářů.
- uc2frq.ods - soubor s funkčními požadavky mapovanými na případy užití.
- bp001.pdf - zadání bakalářské práce.
- brqurqsrq.ods - mapa požadavků rozpracovaných od business až po systémové.