



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

<b>Název:</b>	Webový katalog LEGO dílů pro 3D tisk
<b>Student:</b>	David Hübner
<b>Vedoucí:</b>	Ing. Miroslav Hroněk
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Web a multimédia
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	Do konce letního semestru 2017/18

### Pokyny pro vypracování

Cílem práce je vytvořit webovou aplikaci – katalog LEGO součástí a stavebnic umožňující stažení modelů pro 3D tisk. Navrhněte a implementujte webovou aplikaci umožňující prohledávat, procházet a prohlížet součástky z databáze LDraw a stavebnice z databáze Brickset.com. Využijte možnosti WebGL pro interaktivní prohlížení jednotlivých součástí ve 3D náhledu.

1. Nabídněte stažení součástí ve formátu vhodném pro 3D tisk (a zajistěte převedení LDraw formátu na takový formát, využijte vhodnou existující implementaci konverze) jak jednotlivě, tak i v sadách podle stavebnic a barev.
2. Implementujte mechanismus automatické aktualizace databáze LDraw.
3. Analyzujte právní aspekty takové webové aplikace a pokud to bude legálně možné, aplikaci otevřete do sítě Internet.
4. Kód musí být open source, dobře strukturovaný, řádně otestovaný, vhodně komentovaný a dokumentovaný (obojí v anglickém jazyce).

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.  
děkan

V Praze dne 13. února 2017



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

## **Webový katalog LEGO dílů pro 3D tisk**

*David Hübner*

Vedoucí práce: Ing. Miroslav Hrončok

29. června 2017



---

## Poděkování

Na tomto místě bych rád poděkoval všem lidem, kteří mi pomáhali při vzniku této práce. V první řadě Ing. Miroslavu Hrončokovi, vedoucímu mé bakalářské práce, za cenné připomínky k textové části i implementaci. Dále bych také rád poděkoval své rodině za podporu během celé doby tvorby mé práce.



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl 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 29. června 2017

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2017 David Hübner. 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**

HÜBNER, David. *Webový katalog LEGO dílů pro 3D tisk*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017. Dostupný také z WWW: (<https://github.com/hubnedav/bakalarka>).



---

## Abstrakt

Tato bakalářská práce se věnuje návrhu a implementaci webového katalogu LEGO dílů a stavebnic pro 3D tisk. Rešeršní část práce nejprve představuje knihovnu LDraw poskytující 3D modely dílů LEGO. Dále jsou představeny možné zdroje dat o stavebnicích LEGO. Na základě těchto poznatků je následně navrhována a implementována aplikace samotného webového katalogu.

**Klíčová slova** 3D tisk, Webová aplikace, LEGO, PHP, LDraw, WebGL, Symfony



---

## Abstract

This bachelor thesis deals with design and implementation of a web catalogue of LEGO components for 3D printing. The theoretical part introduces the LDraw library, which provides 3D models of LEGO components. Then the possible sources of data concerning LEGO kits are introduced. Based on these findings, an application of the web catalogue is designed and implemented.

**Keywords** 3D printing, Web application, LEGO, PHP, LDraw, WebGL, Symfony



---

# Obsah

<b>Úvod</b>	<b>21</b>
<b>1 Rešerše</b>	<b>23</b>
1.1 Zdroje dat . . . . .	23
1.2 Převod formátu . . . . .	33
1.3 Vykreslení modelů . . . . .	36
<b>2 Požadavky</b>	<b>39</b>
<b>3 Návrh</b>	<b>41</b>
3.1 Technologie . . . . .	41
3.2 Doménový model . . . . .	42
3.3 Uživatelské rozhraní . . . . .	48
<b>4 Implementace</b>	<b>55</b>
4.1 Použité technologie . . . . .	55
4.2 Architektura . . . . .	57
4.3 Načítání dat . . . . .	58
4.4 3D náhled . . . . .	62
4.5 Stažení součástí . . . . .	63
4.6 Obrázky součástí a stavebnic . . . . .	64
4.7 Vyhledávání . . . . .	65
4.8 Náhled pro sociální síť . . . . .	66

<b>5</b>	<b>Testování a nasazení</b>	<b>71</b>
5.1	Testování . . . . .	71
5.2	Nasazení . . . . .	75
<b>6</b>	<b>Právní aspekty</b>	<b>77</b>
6.1	Zdroje dat . . . . .	77
6.2	Obchodní značka LEGO . . . . .	78
6.3	Závěr . . . . .	79
	<b>Závěr</b>	<b>81</b>
	<b>Zdroje</b>	<b>83</b>
<b>A</b>	<b>Seznam použitých zkratk</b>	<b>89</b>
<b>B</b>	<b>Instalační příručka</b>	<b>91</b>
<b>C</b>	<b>Obsah příloženého média</b>	<b>95</b>

---

## Seznam ukázek kódu

1.1	Ukázka součástky ve formátu LDraw [7]	26
1.2	Ukázka souboru typu <i>Alias</i> [16]	30
1.3	Příklad použití programu LDView	35
1.4	Příklad použití programu ADMesh	36
1.5	Příklad použití programu POV-Ray	36
4.1	Ukázka definice mřížky pomocí Semantic UI	57
4.2	Ukázka příkazu pro načtení knihovny LDraw	58
4.3	Ukázka použití komponenty Symfony Process	60
4.4	Ukázka použití knihovny Monolog	60
4.5	Ukázka logu načítání	61
4.6	Ukázka načítání tabulek CSV	61
4.7	Ukázka part_model.yml	62
4.8	Ukázka načtení instrukcí stavebnice	63
4.9	Ukázka konfigurace filtru LiipImagineBundle	66
4.10	Ukázka nastavení mapování entity <i>Set</i>	67
4.11	Ukázka JSON dotazu Elasticsearch	68
4.12	<i>Twig</i> makro pro přidání Open Graph meta tagů	68
5.1	Výstup PHPUnit testů	71
5.2	Příklad jednotkového testu příkazu načtení knihovny LDraw	72
5.3	Příklad integračního testu	73
6.1	Ukázka souboru LICENSE.txt	78

6.2	Prohlášení o popření právního nároku v patičce stránky . . . . .	79
-----	--	----



---

## Seznam tabulek

3.1	Přehled atributů entity <i>Model</i> . . . . .	44
3.2	Přehled atributů entity <i>Color</i> . . . . .	46



---

## Seznam obrázků

1.1	Ukázkový model z knihovny LDraw [3] . . . . .	24
1.2	Vyrenderovaná součástka z ukázky kódu 1.1 [8] . . . . .	27
1.3	Ukázka součástky typu <i>Part</i> a <i>Patterned</i> [14][15] . . . . .	29
1.4	Ukázka součástky typu <i>Shortcut</i> [17][18][19][20] . . . . .	30
1.5	Diagram schématu, Rebrickable [27] . . . . .	32
1.6	Snímek obrazovky programu LDView . . . . .	34
3.1	Doménový model: LDraw a Rebrickable . . . . .	43
3.2	Doménový model: Brickset . . . . .	47
3.3	Návrh hlavní stránky . . . . .	48
3.4	Návrh stránky výpisu stavebnic . . . . .	49
3.5	Návrh stránky výpisu součástek . . . . .	50
3.6	Návrh stránky detailu stavebnice . . . . .	51
3.7	Návrh stránky detailu součástky . . . . .	52
3.8	Návrh stránky seznamu barev . . . . .	53
4.1	Architektura komponent . . . . .	57
4.2	Diagram načítání součástky knihovny LDraw . . . . .	59
4.3	3D náhled součástky . . . . .	63
4.4	Porovnání renderu součástky a Rebrickable [8] . . . . .	65
4.5	Náhled sdílení stavebnice 2000416-1 Duck . . . . .	69
5.1	Pokrytí kódu testy . . . . .	74
6.1	Snímek obrazovky – domovská stránka . . . . .	82

SEZNAM OBRÁZKŮ

---

6.2 Snímek obrazovky – výpis stavebnic . . . . . 82

---

# Úvod

Výroba součástek a různých objektů pomocí 3D tiskáren v domácích podmínkách se v posledních letech stává stále rozšířenější. Je tomu tak především kvůli stále se zvyšující dostupnosti tiskáren, které si může koncový uživatel pořídit a na kterých si může tisknout své objekty.

Mezi jedno z možných využití tiskáren patří výroba dílů oblíbené stavebnice LEGO. Na internetu existují komunity zabývající se tvorbou 3D modelů LEGO dílů. Tyto modely jsou primárně využívány k tvorbě virtuálních stavebnic, při jejichž tvorbě uživatel není omezován vlastnictvím určitého počtu dílů a prostorem, který stavebnice zabírá [1]. Pokud se však někdo rozhodne tyto součástky převést do reálného světa a postavit si skutečnou stavebnici, existuje postup, pomocí kterého se tyto modely dají převést na vhodný formát a následně vytisknout pomocí 3D tiskárny.

Výstup práce pomůže uživatelům, kteří si chtějí vytisknout svou vlastní stavebnici. Dosavadní postupy pro získání podkladů pro tisk jsou zdlouhavé a je nutné je provést pro každou součástku stavebnice zvlášť.

Z výše uvedených důvodů jsem se rozhodl pro volbu tématu, které výběr i tisk vlastních stavebnic usnadní a ušetří tím uživateli velké množství času jinak stráveného vyhledáváním podkladů k tisku.

## Cíl práce

Cílem práce je navrhnout a implementovat webovou aplikaci umožňující stažení modelů LEGO součástek a stavebnic pro 3D tisk. Návrh je potřeba provést na základě analýzy knihovny LDraw, poskytující modely jednotlivých LEGO součástek, a databáze stavebnic Brickset, které bude aplikace využívat.

Aplikace musí umožňovat vyhledávání, procházení a prohlížení samotných dílů i celých stavebnic. Součástky musí být možné prohlížet interaktivně ve 3D náhledu. Dále aplikace musí zajišťovat automatický převod dílů z formátu knihovny LDraw do formátu vhodného pro 3D tisk a nabídnout jejich stažení jak jednotlivě, tak v sadách podle stavebnic a barev.

V neposlední řadě je nutné analyzovat právní aspekty výsledné aplikace a v případě, že to zjištěné skutečnosti dovolí, aplikaci otevřít do sítě Internet.

## Rešerše

V rámci rešerše jsem se seznamoval se službami, které jsou specifikovány v zadání mé bakalářské práce, a s programy umožňujícími převod formátu LDraw na formát vhodný pro 3D tisk.

### 1.1 Zdroje dat

V této sekci budou představeny následující webové služby:

**LDraw** systém pro LEGO Computer Aided Design (CAD).

**Brickset** webová aplikace zaměřená na informace o LEGO stavebnicích a součástkách.

**Rebrickable** webová aplikace určená k organizaci sbírek LEGO stavebnic a součástek.

#### 1.1.1 LDraw

LDraw je otevřený standard pro LEGO CAD programy, které uživatelům umožňují vytvářet virtuální modely a scény. Je možné ho použít k dokumentaci stavebnic, které jste fyzicky postavili, k vytvoření stavebních pokynů jako od společnosti LEGO, k vykreslení 3D fotorealistických obrázků virtuálních modelů a dokonce i k vytvoření animace. [1]

Podle [2, s. 30] se systém LDraw skládá ze tří základních částí, které se navzájem doplňují. Těmito částmi jsou:

- souborový formát LDraw,
- knihovna součástek,
- programy pro tvorbu modelů.

Pro tvorbu mé bakalářské práce jsou důležité pouze první dvě zmíněné součásti systému LDraw. Obě části nyní podrobně analyzuji a zjištěné skutečnosti později uplatním při návrhu doménového modelu aplikace a mechanismu načítání dat do mé aplikace v kapitole 3.



Obrázek 1.1: Ukázkový model z knihovny LDraw [3]

### Formát LDraw

Základním stavebním kamenem, kolem kterého je postavený celý systém LDraw, je stejnojmenný souborový formát LDraw. Soubory v tomto formátu slouží jak k ukládání jednotlivých součástek, tak i k ukládání celých stavebnic. Soubory jednotlivých tvarů, ze kterých se součástky skládají a zároveň součástky samotné mají typicky příponu *DAT*, pro celé stavebnice (skládající se z více součástek) se používá přípona *LDR*.

Všechny soubory LDraw se řídí pravidly podle oficiální dokumentace, která je dostupná na [4]. Tato pravidla v následujících odstavcích ve zjednodušené podobě představím.



## Parsování

Soubory ve formátu LDraw jsou dle specifikace [5] textové v kódování UCS/Unicode Transformation Format (UTF-8). Skládají se z předem neurčeného množství příkazů, přičemž každý řádek odpovídá právě jednomu příkazu.

Typ příkazu na řádce je určen prvním znakem, který se na ní vyskytuje (s výjimkou bílých znaků<sup>1</sup>). Možné typy řádků jsou následující:

0: **komentář nebo META příkaz**

1: **reference na jiný soubor**

Příkaz: 1 <colour> x y z a b c d e f g h i <file>, kde:

- <colour> je id barvy specifikované v [6],
- x y z jsou souřadnice pro umístění součástky,
- a b c d e f g h i je levá horní  $3 \times 3$  matice standardní  $4 \times 4$  homogenní transformační matice,
- <file> je jméno vkládaného souboru.

2: **úsečka**

3: **trojúhelník**

4: **čtyřúhelník**

5: **volitelná úsečka**

V ukázce kódu 1.1 můžeme vidět příklad jednoduché součástky ve formátu LDraw. Tuto vyrenderovanou součástku je možné vidět na obrázku 1.2.

## Knihovna LDraw

Knihovna LDraw poskytuje velké množství<sup>2</sup> 3D modelů LEGO součástek ve formátu LDraw.

<sup>1</sup>Bílý znak je dle specifikace [5] definován jako jedna nebo více mezer, znaků tabulátoru nebo jejich kombinace.

<sup>2</sup>V době psaní práce obsahuje knihovna 9 781 součástek.

## 1. REŠERŠE

---

```
1 0 Brick 2 x 2
2 0 Name: 3003.dat
3 0 Author: James Jessiman
4 0 !LDRAW_ORG Part UPDATE 2002-03
5 0 !LICENSE Redistributable under CCAL version 2.0 : see CReadme.txt
6
7 0 BFC CERTIFY CCW
8
9 0 !HISTORY 2001-10-26 [PTadmin] Official Update 2001-01
10 0 !HISTORY 2002-05-07 [unknown] BFC Certification
11 0 !HISTORY 2002-06-11 [PTadmin] Official Update 2002-03
12 0 !HISTORY 2007-05-07 [PTadmin] Header formatted for Contributor
13 0 !HISTORY 2008-07-01 [PTadmin] Official Update 2008-01
14
15 1 16 0 4 0 1 0 0 0 -5 0 0 0 1 stud4.dat
16
17 0 BFC INVERTNEXT
18 1 16 0 24 0 16 0 0 0 -20 0 0 0 16 box5.dat
19
20 4 16 20 24 20 16 24 16 -16 24 16 -20 24 20
21 4 16 -20 24 20 -16 24 16 -16 24 -16 -20 24 -20
22 4 16 -20 24 -20 -16 24 -16 16 24 -16 20 24 -20
23 4 16 20 24 -20 16 24 -16 16 24 16 20 24 20
24
25 1 16 0 24 0 20 0 0 0 -24 0 0 0 20 box5.dat
26
27 1 16 10 0 10 1 0 0 0 1 0 0 0 1 stud.dat
28 1 16 -10 0 10 1 0 0 0 1 0 0 0 1 stud.dat
29 1 16 10 0 -10 1 0 0 0 1 0 0 0 1 stud.dat
30 1 16 -10 0 -10 1 0 0 0 1 0 0 0 1 stud.dat
31 0
```

**Ukázka kódu 1.1:** Ukázka součástky ve formátu LDraw [7]



**Obrázek 1.2:** Vyrenderovaná součástka z ukázky kódu 1.1 [8]

Součástky z knihovny je možné obdržet jednotlivě z adresy <http://www.ldraw.org/library/official/filepath>, kde filepath je cesta k souboru v knihovně LDraw [9]. Další možností je obdržení kompletní knihovny v podobě ZIP archivu. Struktura obsahu tohoto archivu je zobrazena na diagramu adresářové struktury 1.1.

	models.....	složka, do které jsou ukládány vytvořené modely
	p .....	primitiva používaná při tvorbě modelů
	parts .....	kompletní součástky
	* .dat.....	soubory součástek
	s .....	často používané společné části součástek
	CAlicense.txt.....	licenční podmínky
	CAreadme.txt.....	vysvětlení licenčních podmínek
	LDConfig.ldr.....	definice barev
	Readme.txt.....	popis obsahu archivu
	mklist.exe .....	program MKList k vytvoření seznamu součástek
	mklist1_6.zip.....	archiv se zdrojovým kódem programu MKList

**Adresářová struktura 1.1:** Obsah archivu complete.zip

Aby byla vytvořená součástka přidána do oficiální knihovny LDraw, musí projít schvalovacím procesem. V tom je kontrolováno dodržení všech pravidel specifikace formátu [5] i další pravidla platná pro knihovnu [10].

### Hlavička souboru

Každý LDraw soubor, který se nachází v knihovně, musí obsahovat hlavičku s obsahem řídícím se pravidly [11]. V hlavičce souboru jsou uložena metadata

poskytující doplňující informace o součástce. Mezi tyto informace mimo jiné patří:

- název,
- autor,
- licence,
- typ součástky,
- klíčová slova,
- kategorie.

### Typy prvků

V knihovně se nachází prvky několika různých typů. Specifikace je v tomto ohledu velmi rozsáhlá [11][12]. Typ prvku je možné rozpoznat podle:

- názvu (první řádek souboru LDraw),
- META příkazů hlavičky,
- čísla (jméno souboru).

Po detailním prozkoumání většího množství prvků musím podotknout, že ne všechny soubory knihovny přesně odpovídají specifikaci.

Možné typy prvků:

- **Part:** Geometricky kompletní součástka.
- **Subpart:** Sám o sobě nekompletní útvar.
- **Primitive:** Základní geometrický útvar vyskytující se v mnoha součástkách.

Pro více informací o primitivech a jejich kompletní seznam doporučuji [13].

- **Shortcut:** Součástka složená z více kompletních součástek.

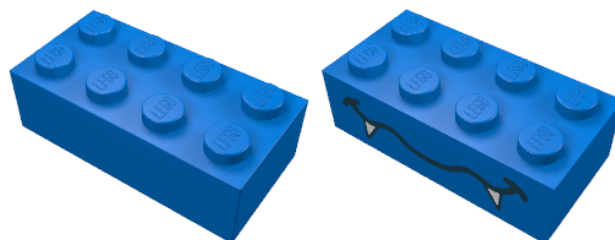
Na obrázku 1.4 je možné vidět render součástky 3015c01 (nahore) skládající se ze součástek 3815, 3816, 3817 (dole).

- **Alias:** Soubor odkazující na jiné místo v knihovně (například z důvodu používání dvou různých čísel pro stejný prvek společností LEGO).

Soubor typu *Alias* typicky obsahuje pouze jeden příkaz typu 1. Náhled souboru typu *Alias* je k vidění v ukázce kódu 1.2.

- **Physical\_Colour:** Soubor odkazující na jiné místo v knihovně a navíc určující barvu.
- **Sticker:** Potisk.
- **Patterned:** Součástka s potiskem.

Porovnání součástky typu *Part* a odpovídající součástky typu *Patterned* je vidět na obrázku 1.3.



**Obrázek 1.3:** Ukázka součástky typu *Part* a *Patterned* [14][15]

## Licence

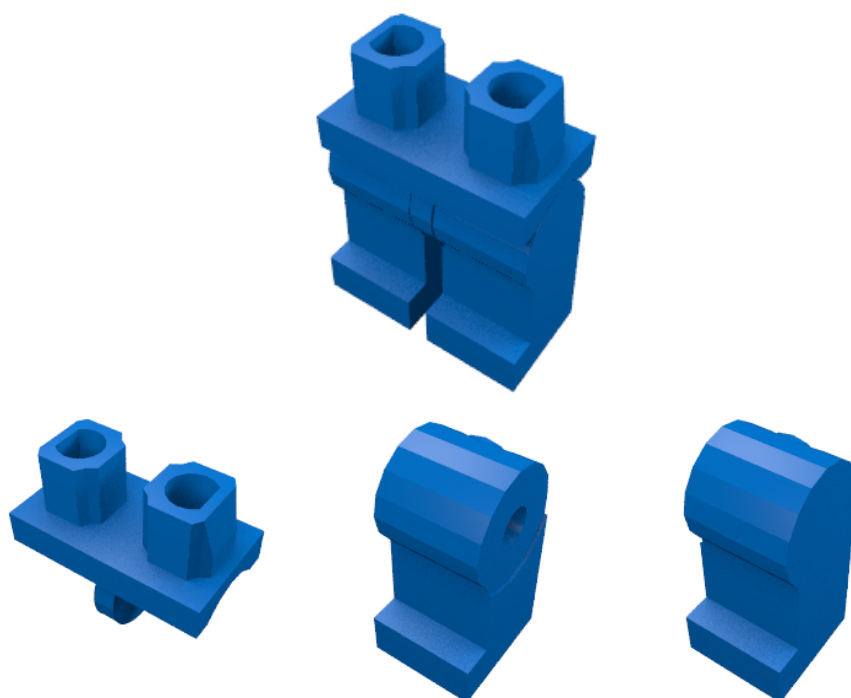
Všechny součástky zahrnuté do oficiální knihovny LDraw jsou vydány pod licencí Creative Commons Attribution 2.0 Generic (CC-BY) [21]. Ta umožňuje úpravu i sdílení díla za předpokladu dodržení podmínky o uvedení autora, licence a označení případných provedených změn.

## 1. REŠERŠE

---

```
1 0 ~Moved to 6141
2 0 Name: 4073.dat
3 0 Author: [PTadmin]
4 0 !LDRW_ORG Part Alias UPDATE 2015-01
5 0 !LICENSE Redistributable under CCAL version 2.0 : see Creadme.txt
6
7 0 BFC CERTIFY CCW
8
9 0 !HISTORY 2015-10-11 [PTadmin] Official Update 2015-01
10
11 0 // Plate 1 x 1 Round
12
13 1 16 0 0 0 1 0 0 0 1 0 0 0 1 6141.dat
```

**Ukázka kódu 1.2:** Ukázka souboru typu *Alias* [16]



**Obrázek 1.4:** Ukázka součástky typu *Shortcut* [17][18][19][20]

### 1.1.2 Brickset

Brickset je populární webová aplikace poskytující detailní informace a novinky o stavebnicích LEGO. Má rozsáhlou základnu aktivních uživatelů<sup>3</sup> a sama o sobě prohlašuje, že je primárním zdrojem informací o LEGO stavebnicích na internetu [23].

#### Poskytovaná data

Data z databáze Brickset jsou dostupná přes veřejné Application Programming Interface (API). Přes API je možné obdržet informace týkající se stavebnic, instrukce k jejich postavení, recenze a obrázky stavebnic. Dále je přes Brickset API možné spravovat uživatelské sbírky stavebnic.

Přestože webové stránky Brickset umožňují i prohlížení inventářů jednotlivých stavebnic, Brickset nemá oprávnění k jejich sdílení přes své API a ve své dokumentaci [24] odkazuje k využití API webové stránky Rebrickable.com.

#### API

Podrobná specifikace API s výčtem dostupných metod je na [24]. Brickset API je implementováno jako služba postavená nad protokolem Simple Object Access Protocol (SOAP).

#### Závěr

Pro účely vytvářené aplikace není využití samotných dat poskytovaných přes API Brickset dostačující. Bude tedy nutné využít další zdroje poskytující informace o inventářích stavebnic. API služby Brickset bude využito pouze k obdržení popisu stavebnic, obrázků, instrukcí a recenzí.

### 1.1.3 Rebrickable

Rebrickable [25] je webová aplikace sloužící ke správě sbírek LEGO součástek a stavebnic. Umožňuje uživateli prohlížení rozsáhlé databáze obsahující více než 11 500 oficiálních stavebnic a více než 25 000 unikátních součástek prodávaných společností LEGO od roku 1950 po současnost [26].

---

<sup>3</sup>Brickset má více než 150 000 registrovaných uživatelů a v průměru 20 000 aktivních uživatelů každý měsíc. [22]

## Poskytovaná data

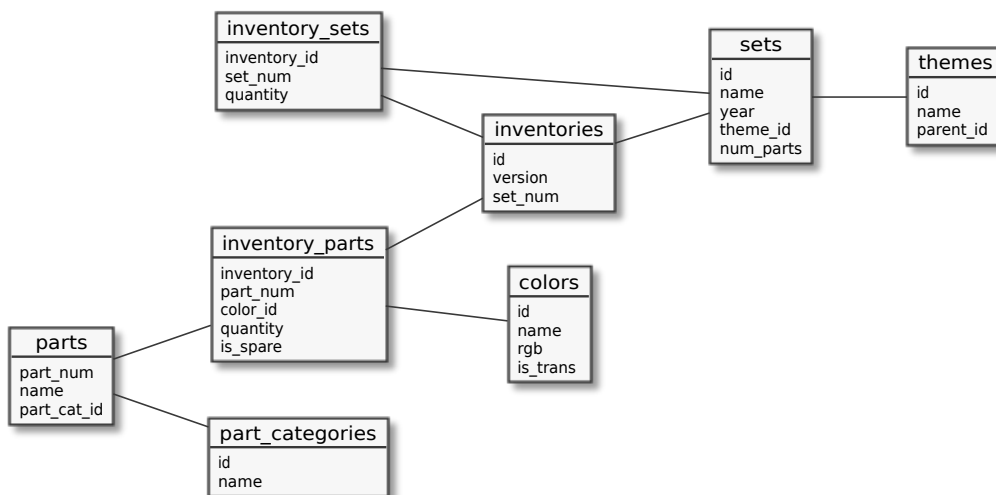
Rebrickable svou databázi zpřístupňuje dvěma způsoby. Jedním způsobem je pravidelný (měsíční) výpis z databáze formou tabulek ve formátu *Comma-separated values (CSV)*. Druhou možností je získávání dat přes veřejné API. Struktura tabulek *CSV*, které jsou ke stažení ze stránek [27] je znázorněna na obrázku 1.5.

## Číslování součástek

Velkou výhodou použití dat služby Rebrickable je systém číslování součástek, který vychází z číslování u knihovny LDraw [28]. To ušetří velké množství práce při mapování součástek z databáze Rebrickable na součástky knihovny LDraw.

## Licence

Data aplikace jsou veřejně dostupná pro jakékoliv účely, včetně komerčních. Autoři aplikace pouze žádají o zveřejnění prohlášení o původu dat. [29]



Obrázek 1.5: Diagram schématu, Rebrickable [27]

## Závěr

Pro získání dat o inventářích jednotlivých stavebnic bude využita databáze služby Rebrickable ve formě *CSV* tabulek. Využití kompletních dat je pro účely



aplikace optimálním řešením. Na rozdíl od varianty dynamického získávání dat přes API umožňuje předem připravené namapování součástek na 3D modely a celkově rychlejší přístup k datům.

### 1.1.4 Další zdroje dat

V rámci rešerše jsem se seznamoval i s dalšími službami, které poskytují data o inventářích stavebnic LEGO – Bricklink [30] a BrickOwl [31]. Obě služby poskytují také svá data pomocí API, nevyhází však u číslování součástek ze systému LDraw a tak by párování 3D modelů a součástek bylo o poznání složitější.

## 1.2 Převod formátu

S ohledem na požadavek na zajištění převodu formátu LDraw je v této sekci nejprve představen formát *Stereolithography format (STL)* nejčastěji využívaný k 3D tisku [32]. Následně jsou představena možná řešení samotného převodu. Nakonec jsou představeny nástroje pro vykreslení souborů *STL*.

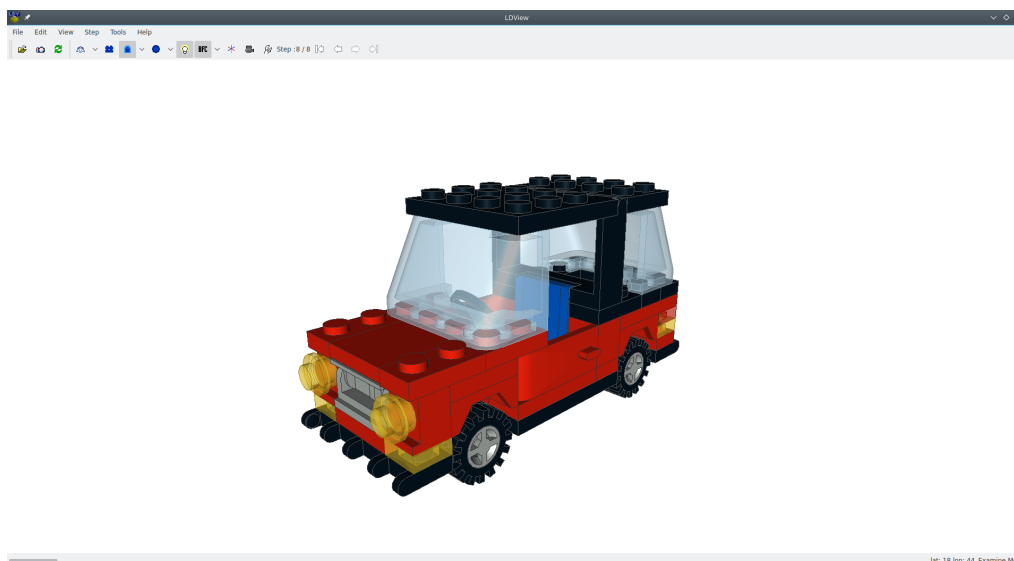
### 1.2.1 STL

Souborový formát *STL* byl vyvinut firmou 3D Systems. Soubor popisuje trojrozměrnou povrchovou geometrii modelu a je nejčastěji používaným souborovým formátem pro potřeby 3D tisku. [32]

Modely ve formátu *STL* jsou reprezentovány trojúhelníkovou sítí. Každý trojúhelník je popsán třemi krajními body a normálovým vektorem. [33]

*STL* modely je možné ukládat dvěma způsoby:

- **ASCII:** Verze formátu čitelná pro člověka. Je však datově velmi neefektivní.
- **Binární:** Verze formátu nabízí efektivnější uložení dat modelu za cenu ztráty čitelnosti.



Obrázek 1.6: Snímek obrazovky programu LDView

### 1.2.2 LDView

LDView je 3D prohlížeč modelů a součástek ve formátu LDraw, představeném v sekci *Formát LDraw* na straně 24. Program je veřejně dostupný pod licencí GNU General Public License, verze 2 (GPLv2) [34].

Mezi jedny z funkcí programu patří:

- schopnost exportovat LDraw modely do formátů *POV-Ray*, *3DS* nebo *STL*,
- schopnost pořizovat snímek aktuálního pohledu.

Program je dostupný jak ve verzi s grafickým prostředím, tak v konzolové verzi využitelné například na systémech bez grafického prostředí.

Pro více informací o programu a jeho dostupných funkcích doporučuji [35].

### Převod formátu

Převod formátu LDraw do jiných formátů pomocí programu LDView je velmi jednoduchý. V ukázce 1.3 je příklad použití konzolové verze programu.

Při konzultaci s vedoucím bylo odhaleno, že převedené součástky mají 10krát menší měřítko a jsou o 90° otočené kolem osy X. Formát *STL* dle specifikace [33] nemá jednotku a není definováno, která osa ukazuje směrem vzhůru. Při Fused Deposition Modeling (FDM) 3D tisku je však praxí, že se předpokládá jednotka mm a osa Z směřující vzhůru. Z tohoto důvodu je vhodné provést korekci souborů *STL*.

Oprava těchto chyb ve zdrojovém kódu programu by stála velké množství práce a náročné zorientování se v cizím kódu. Proto jsem se rozhodl opravu měřítka a natočení modelu provést pomocí nástroje ADMesh.

```
1 $ ldview ./ldraw/parts/3003.dat -LDrawDir=./ldraw -ExportFile=3003.stl
```

**Ukázka kódu 1.3:** Příklad použití programu LDView

### 1.2.3 ADMesh

ADMesh [36] je open-source program umožňující provádět operace nad soubory ve formátu *STL*. Jedná se o program bez grafického rozhraní spouštěný z příkazové řádky. V ukázce 1.4 je příklad použití programu ADMesh pro zvětšení měřítka, rotaci kolem osy X a následného uložení v podobě binárního *STL* souboru.

Funkce:

- kontrolování závad (chybné normálové vektory, neuzavřené stěny) i jejich automatická oprava,
- rotace modelu,
- translace,
- změna velikosti,
- čtení a zápis binárních i textových *STL* souborů.

```
1 $ admesh 3003.stl --x-rotate=-90 --scale=10 --no-check  
  ↪ --write-binary-stl=3003.stl
```

**Ukázka kódu 1.4:** Příklad použití programu ADMesh

### 1.3 Vykreslení modelů

Pro možnost poskytnout uživatelsky přívětivé a přehledné prostředí aplikace je nepochybně nutné mít možnost zobrazit součástky v podobě statického obrázku. Obrázek většiny součástek z knihovny LDraw je volně k dispozici ke stažení z webových stránek již představené služby Rebrickable [27]. Pro zbytek součástek je však nutné obrázek vyrenderovat.

Nejjednodušší možností pro vyrenderování obrázků součástek je využití již představeného programu LDView, který bude v aplikaci využíván. Program umožňuje jako jednu ze svých funkcí i vykreslení aktuálního pohledu. Tato možnost se však ukázala jako velmi nespolehlivá (v závislosti na operačním systému, na kterém je program volán).

Proto jsem se rozhodl využít jiné nástroje k vyrenderování obrázků součástek a to sice programy stl2pov [37] a POV-Ray [38].

#### 1.3.1 POV-Ray

Dle [39] je POV-Ray nástroj pro vykreslování trojrozměrných scén s důrazem na co nejvyšší kvalitu výsledku. Celá scéna je popsána v textovém souboru, který svou podobou připomíná jazyk C. POV-Ray takto popsanou scénu přečte, provede vykreslení a výsledek uloží ve formě rastrového obrázku. Podobu příkazu pro vykreslení scény na pozadí je možné vidět v ukázce kódu 1.5.

```
1 $ povray +W900 +H600 +FN -D +I"input.pov" +O"output.png"
```

**Ukázka kódu 1.5:** Příklad použití programu POV-Ray

### 1.3.2 Stl2pov

Stl2pov je program umožňující převod *STL* souboru na POV-Ray objekt, který je možné použít do specifikace POV-Ray scény k vykreslení.



---

## Požadavky

V této sekci jsou shrnuty požadavky kladené na vytvářenou aplikaci specifikované v zadání této práce. Požadavky jsou rozděleny na funkční a nefunkční.

### 2.0.1 Funkční požadavky

#### FP-1 Procházení, vyhledávání a prohlížení součástí a stavebnic

Aplikace musí umožňovat procházení, vyhledávání a prohlížení součástí z knihovny LDraw a stavebnic z databáze Brickset.

#### FP-2 Stažení 3D modelů

3D modely součástí musí být možno stáhnout samostatně i v sadách modelů obsažených v jednotlivých stavebnicích. Stavebnice navíc musí být možno stáhnout i ve verzi roztríděné podle barev součástí.

#### FP-3 Zajištění převodu formátu LDraw na formát vhodný pro 3D tisk

Převod formátu bude zajištěn využitím vhodné existující implementace konverze.

#### FP-4 Interaktivní prohlížení součástí ve 3D náhledu

Aplikace musí umožňovat interaktivní prohlížení součástí ve 3D náhledu.

### FP-5 Automatická aktualizace databáze LDraw

Databázi součástek z knihovny musí být možné automaticky aktualizovat.

## 2.0.2 Nefunkční požadavky

### NP-6 Open-source kód

Zdrojový kód aplikace musí být vydán pod open-source licenci.

### NP-7 Využití WebGL pro 3D náhled součástek

K interaktivnímu prohlížení součástek ve 3D náhledu musí být využito Web Graphics Library (WebGL).

### NP-8 Dobře strukturovaný, řádně otestovaný, vhodně okomentovaný a dokumentovaný zdrojový kód

### NP-9 Komentáře i dokumentace v anglickém jazyce



---

## Návrh

Důležitou součástí každého vývoje je návrhová fáze. V této fázi jsou využity znalosti získané během analytické a rešeršní části práce. Návrh je později využit při implementační fázi vývoje. Tato kapitola se nejprve zabývá výběrem vhodných technologií pro implementační část práce. Dále se zabývá návrhem architektury aplikace a návrhem uživatelského rozhraní.

### 3.1 Technologie

Jednou ze základních otázek při tvorbě webové aplikace je volba technologií, které budou při implementaci využity. Od volby se odvíjí jak návrh, tak implementace samotná.

#### 3.1.1 PHP

Pro tvorbu webové aplikace jsem se rozhodl využít programovacího jazyka PHP: Hypertext Preprocessor (PHP). Jedním z hlavních kritérií zohledněných při výběru technologie byla jistá předchozí zkušenost s programovacím jazykem PHP, který patří mezi nejpoužívanější technologie pro tvorbu webových aplikací. V současné době je v jazyce PHP napsáno více než 82 % webových stránek se skriptováním na straně serveru [40]. Vzhledem k vysoké rozšířenosti je o jazyce PHP dostupné velké množství informací, které usnadňují vývoj.

### 3.1.2 Framework Symfony 3

Po provedení průzkumu mezi PHP frameworky jsem se rozhodl svou práci vyvíjet s využitím frameworku Symfony 3. Hlavním důvodem pro výběr Symfony byla nativní podpora konzolových příkazů [41], kvalitně zpracovaná dokumentace a využití knihovny Doctrine Object-relational mapping (ORM) [42].

## 3.2 Doménový model

Doménový model (na obrázku 3.1 a 3.2) popisuje strukturu dat a vazby mezi jednotlivými entitami. Je důležitý pro učinění rozhodnutí jaké objekty a jejich vztahy bude nutné v aplikaci uchovávat.

Při vytváření doménového modelu je vycházeno ze tří zdrojů dat, které jsou stanoveny v rešeršní části práce. Z hlediska domény je tedy vhodné rozdělit entity do balíčků podle zdroje, ze kterého pochází. Využité zdroje dat jsou:

- knihovna LDraw,
- Rebrickable,
- Brickset.

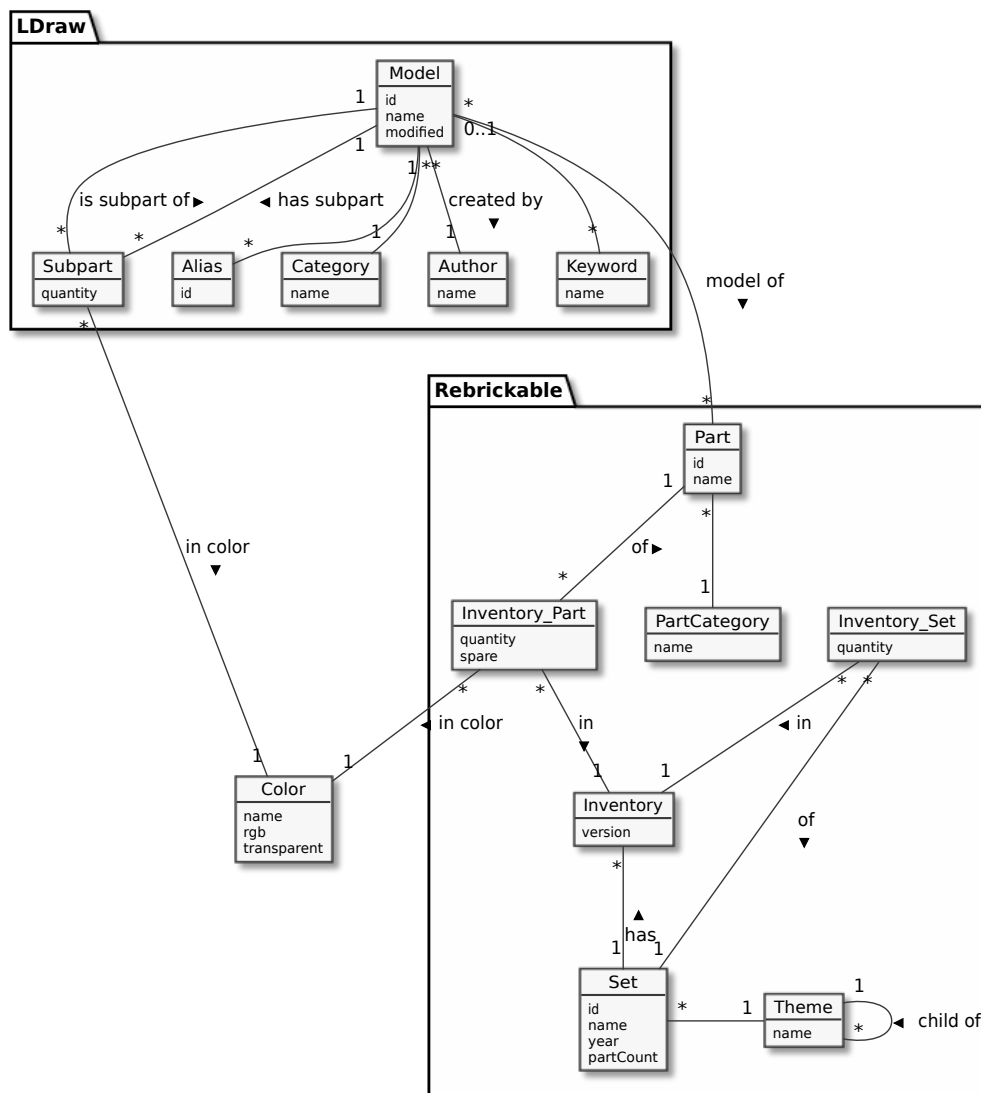
Pro větší přehlednost jsem doménový model rozdělil na dva diagramy. Diagram na obrázku 3.1 popisuje lokálně ukládaná data knihovny LDraw a služby Rebrickable. Diagram na obrázku 3.2 data dostupná přes API Brickset.

### 3.2.1 LDraw

První balíček obsahuje entity týkající se součástek z knihovny LDraw.

#### Model

Entita *Model* (tabulka 3.1) reprezentuje jednu součástku z knihovny LDraw. Atributy entity vychází z dat obsažených v jednotlivých souborech knihovny a ze specifikace hlavičky, která byla představena v podsekci *Hlavička souboru* sekce 1.1.1.



Obrázek 3.1: Doménový model: LDraw a Rebrickable

### 3. NÁVRH

---

U každého *Modelu* musí být evidován jeho autor z důvodu možnosti dodržení licence CC-BY [21], pod kterou jsou zveřejněny všechny součástky v oficiální knihovně LDraw.

**Tabulka 3.1:** Přehled atributů entity *Model*

Atribut	Popis
id:	Unikátní textový identifikátor modelu
name:	Jméno modelu
modified:	Datum poslední úpravy modelu
partCount:	Počet součástí ve stavebnici
path:	Cesta k souboru 3D modelu na serveru

#### **Author**

Entita *Author* reprezentuje autora, který svými modely přispívá do knihovny LDraw. Aby mohl být model zahrnut do oficiální knihovny, musí být autor registrován a projevit souhlas s Dohodou přispěvatelů Ldraw.org [43].

#### **Alias**

Entita *Alias* reprezentuje alternativní identifikátor modelu, na který má vazbu.

Uchování alternativních identifikátorů modelů slouží ke sjednocení tvarově identických součástí. Uchování každé součástky by z hlediska 3D tisku nemělo žádný význam, protože jejich rozdíly (například potisk) zaniknou okamžikem převodu do formátu *STL*.

#### **Subpart**

Protože formát LDraw umožňuje i vytváření součástí typu *Shortcut*, které jsou definovány v podsekci *Typy prvků* sekce 1.1.1, je nutné mít možnost uchovat informaci o vztahu mezi jednotlivými modely. Toto je umožněno entitou *Subpart*.

Entita *Subpart* kromě vazeb na *Model* obsahuje atribut specifikující četnost výskytu podsoučástky a vazbu na entitu *Color*, která specifikuje barvu.

### **Category**

Každý model je zařazen do kategorie, která je určena v hlavičce souboru.

### **Keyword**

K možnosti vyhledávání v knihovně LDraw mohou součástky a podsoučástky definovat klíčová slova.

## **3.2.2 Rebrickable**

Druhý balíček sdružuje entity ze služby Rebrickable, která poskytuje inventáře součástek a stavebnic.

### **Set**

Entita *Set* reprezentuje jednu stavebnici LEGO. Každá stavebnice má své unikátní id určené přímo společností LEGO.

### **Part**

Entita *Part* reprezentuje jednu unikátní součástku LEGO z databáze Rebrickable.

### **Inventory**

Stavebnice může být během času vydána v novější verzi, ve které je například vyměněna pouze jedna součástka. To vede k potřebě zvláštní entity *Inventory*, která reprezentuje tyto verze inventářů.

### **Inventory\_Part**

Součástky se ve stavebnicích mohou vyskytovat v různých barevných provedeních. Dále je běžné, že stavebnice LEGO obsahují některé součástky navíc, které nejsou nezbytné k jejich sestavení. Proto je v nutná entita *Inventory\_Part*, která umožňuje zaznamenání tohoto vztahu mezi součástkou a inventářem, včetně informace o počtu, barvě a typu součástky.

#### **Inventory\_Set**

Stavebnice se nemusejí skládat pouze ze součástí, ale mohou sdružovat i větší množství jiných stavebnic. Proto je nutná entita *Inventory\_Set*, která reprezentuje tento vztah mezi stavebnicí a inventářem.

#### **Theme**

Každá stavebnice náleží do série. Série je reprezentována entitou *Theme*. Série může být zároveň podřazena jiné sérii. Zpravidla je zanoření maximálně tři úrovně hluboké.

#### **Color**

Entita *Color* (tabulka 3.2) je společná pro oba balíčky. To je dáno faktem, že Rebrickable se svými daty vychází právě z knihovny LDraw, jak bylo zmíněno v sekci 1.1.3.

**Tabulka 3.2:** Přehled atributů entity *Color*

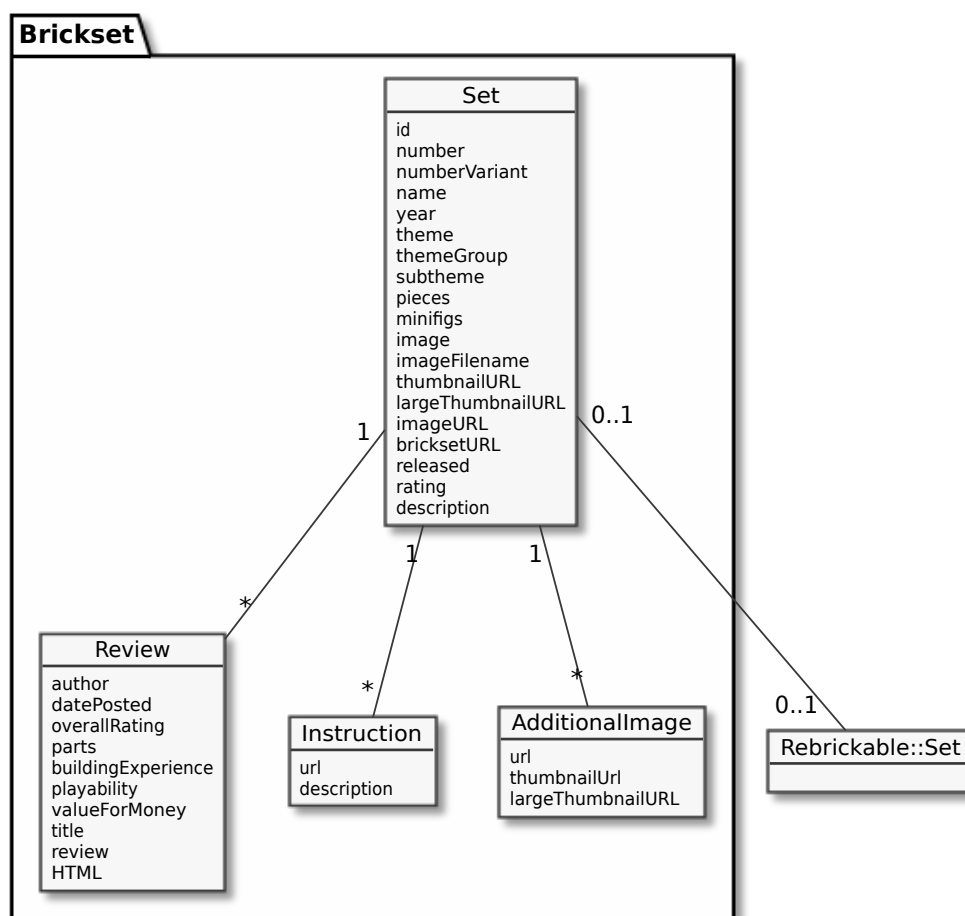
Atribut	Popis
id:	Unikátní LDraw id barvy [6]
name:	Jméno barvy
rgb:	Hexadecimální RGB kód barvy
transparent:	Indikátor o transparentnosti barvy

#### **3.2.3 Brickset**

Třetí balíček popisuje strukturu dat dostupných přes Brickset API.

#### **Set**

Entita *Set* reprezentuje jednu stavebnici LEGO. Tato entita odpovídá právě jedné nebo žádné entitě *Set* z balíků Rebrickable.



Obrázek 3.2: Doménový model: Brickset

## Review

Brickset umožňuje uživatelům vytvářet recenze na stavebnice. Tyto recenze jsou reprezentovány entitou *Review*. Recenze kromě textového popisu obsahuje i číselné hodnocení hlavních kritérií, znázorněných v diagramu.

## Instruction

Instrukce k postavení stavebnice je reprezentována entitou *Instruction*.

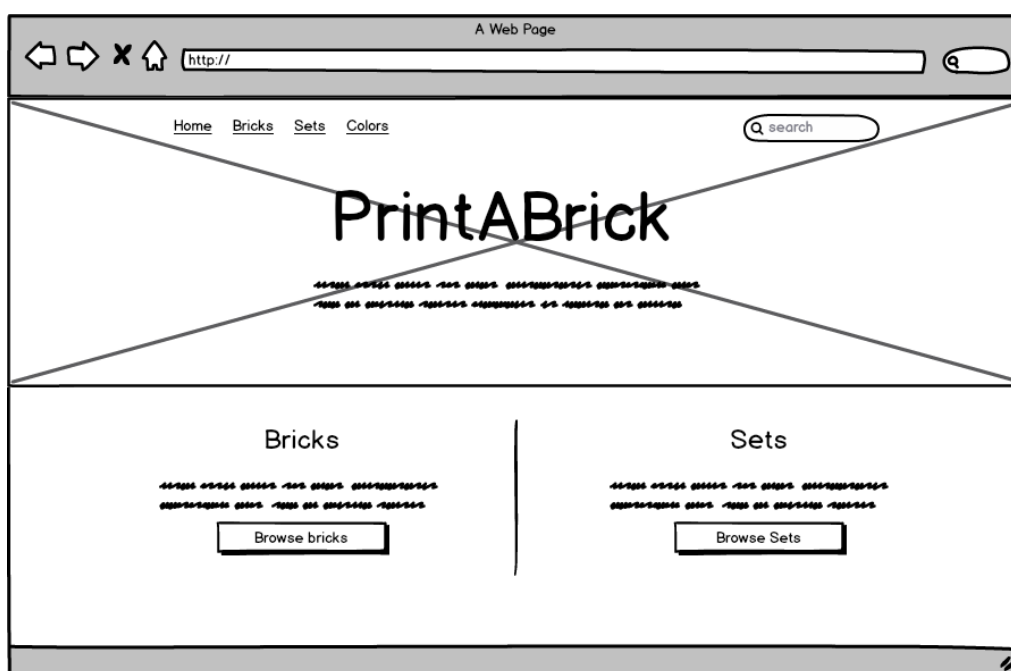
### AdditionalImage

Entita *AdditionalImage* reprezentuje jeden obrázek stavebnice. Obrázky jsou dostupné kromě původní velikosti i v podobě miniatur.

## 3.3 Uživatelské rozhraní

### 3.3.1 Domovská stránka

Návrh domovské stránky (obrázek 3.3) je velmi minimalistický. Stránka slouží pouze k seznámení uživatele s významem aplikace a k navigaci na další stránky pomocí dvou velkých bloků umístěných pod úvodní fotografií. V pravém horním rohu se nachází vyhledávací pole, které umožňuje uživateli rychlou navigaci na součástky a stavebnice.



Obrázek 3.3: Návrh hlavní stránky

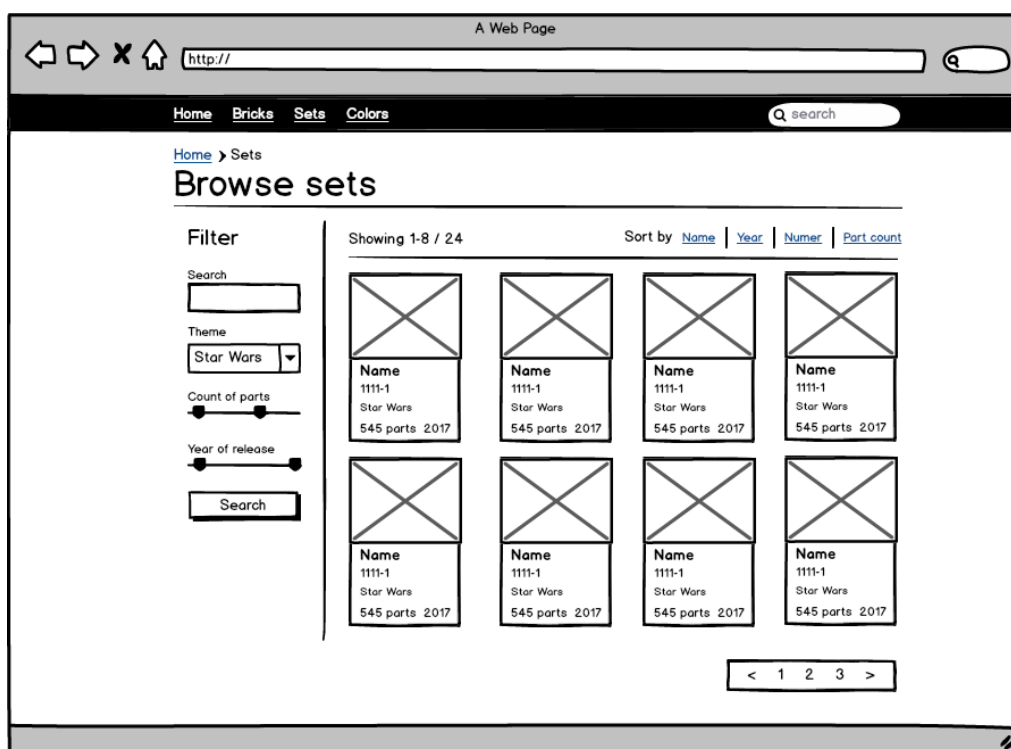
### 3.3.2 Výpis stavebnic

Na obrázku 3.4 je možné vidět návrh stránky výpisu stavebnic. Tento výpis je stránkovaný a je možné ho řadit podle hlavních atributů stavebnic. Uživatel



může specifikovat kritéria filtrování v levém sloupečku. Sloupeček obsahuje textové pole pro zadání vyhledávaného výrazu, výběr kategorie a posuvníky určující počet součástí a rok vydání stavebnice.

Výpis stavebnic je zobrazen v mřížce. Každý blok reprezentující stavebnici obsahuje obrázek a hlavní atributy, které mohou být uživateli nápomocné při výběru.



Obrázek 3.4: Návrh stránky výpisu stavebnic

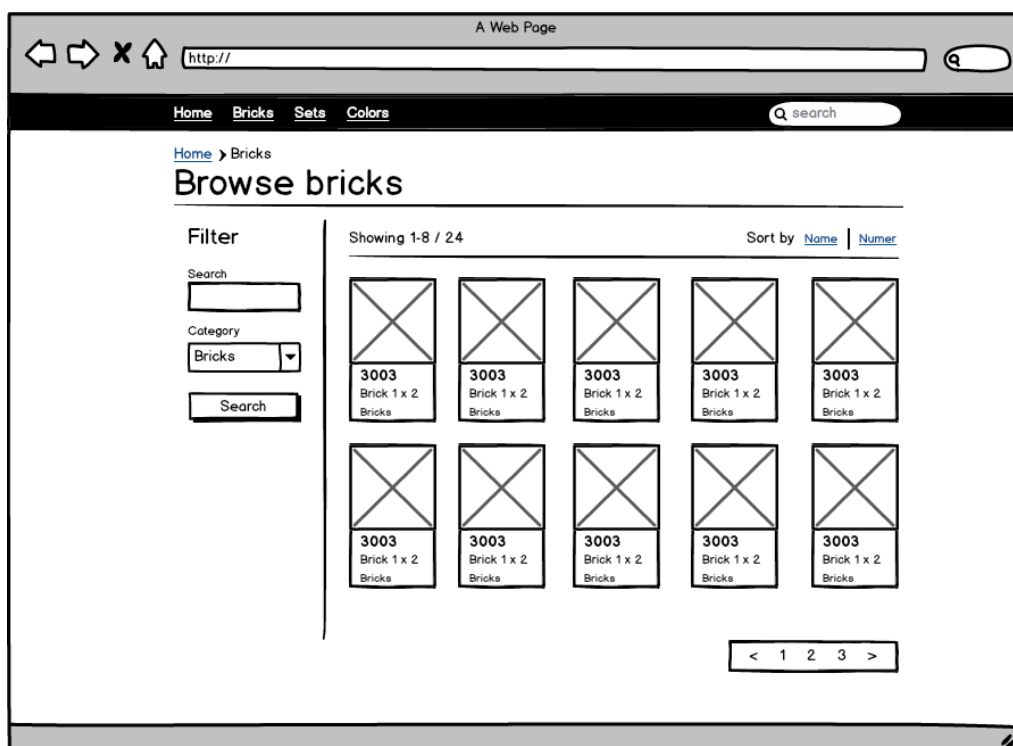
### 3.3.3 Výpis součástek

Stránka výpisu součástek (obrázek 3.5) je rozložením prvků navržena stejně jako stránka výpisu stavebnic.

### 3.3.4 Detail stavebnice

Stránka detailu stavebnice (obrázek 3.6) představuje uživateli veškeré dostupné informace o stavebnici. Stránka je graficky rozdělena na dva bloky.

### 3. NÁVRH



Obrázek 3.5: Návrh stránky výpisu součástek

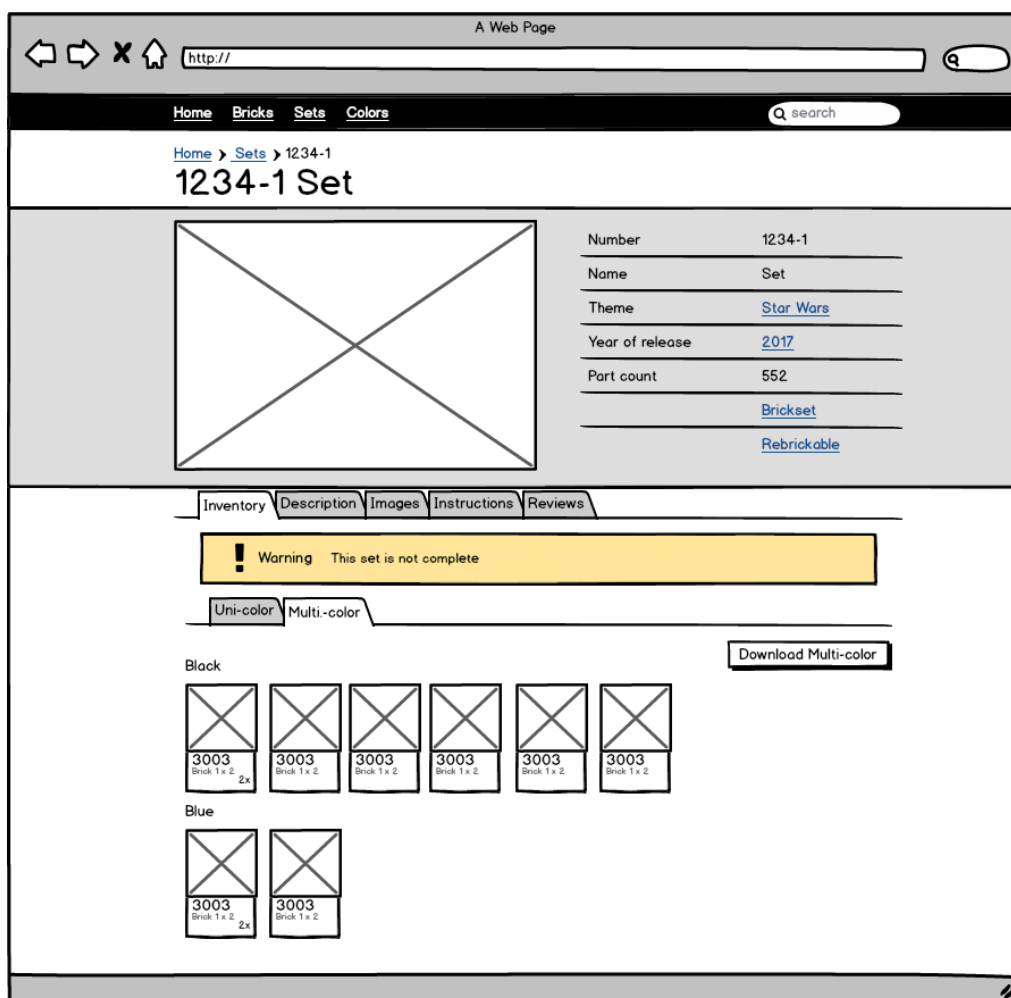
Hlavní blok obsahuje obrázek stavebnice a atributy zobrazené v tabulce. Dále obsahuje odkazy na služby ze kterých pochází zobrazená data.

Druhý blok je pro lepší přehlednost rozdělen do podstránek, které je možné přepínat pomocí horizontálního menu. Výchozí podstránkou je výpis součástek obsažených ve stavebnici. Tento výpis je možné zobrazit jak ve variantě ignorující barvy součástek, tak ve variantě roztříděné podle barev. Každá z těchto variant obsahuje tlačítko pro stažení 3D modelů součástek.

Pokud pro stavebnici nejsou dostupné 3D modely všech součástek, je na stránce zobrazeno varování.

#### 3.3.5 Detail součástky

Stránka detailu součástky (obrázek 3.7) je navržena podobně jako stránka detailu stavebnice.

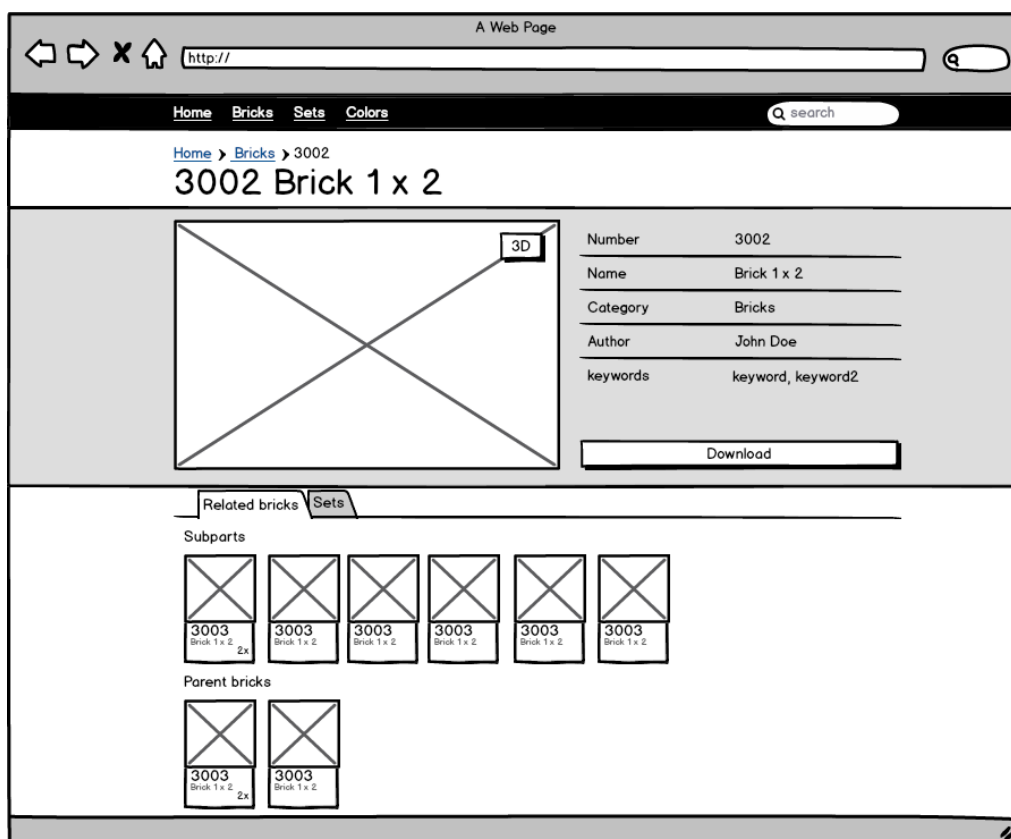


Obrázek 3.6: Návrh stránky detailu stavebnice

Hlavní blok obsahuje obrázek součástky a přehled atributů. Obrázek součástky má v pravém horním rohu tlačítko, kterým se spouští interaktivní 3D náhled. Dále blok obsahuje tlačítko pro stažení 3D modelu.

Druhý blok obsahuje dvě podstránky. První podstránka zobrazuje všechny příbuzné součástky. V druhé je možné prohlížet veškeré stavebnice, ve kterých se součástka vyskytuje.

### 3. NÁVRH

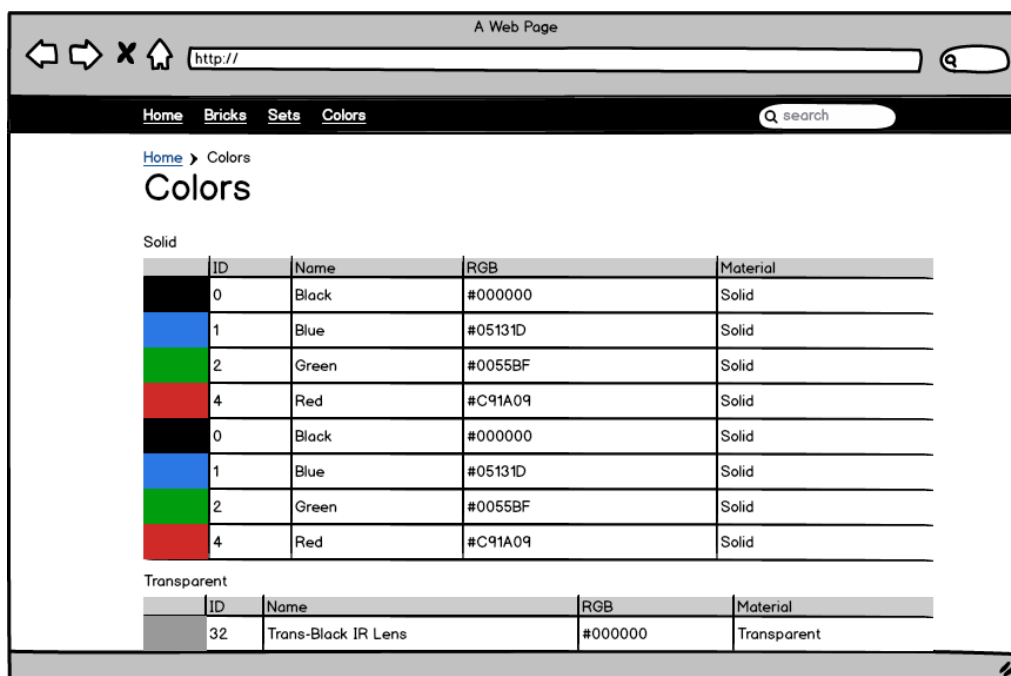


Obrázek 3.7: Návrh stránky detailu součástky

#### 3.3.6 Seznam barev

Stránka seznamu barev (obrázek 3.8) slouží uživatelům k seznámení se s možnými barevnými variantami součástek. Barvy jsou zobrazeny ve dvou tabulkách. První tabulka sdružuje klasické barvy a druhá transparentní barvy.

Tato stránka podle původního návrhu sloužila k možnosti dohledání přesné barvy součástky podle jména barvy. Později byla však tato informace zahrnuta přímo na místa, kde se vyskytuje zmínka o barvě. Tato stránka tedy postrádá svůj původní význam a mohla by být odstraněna.



Obrázek 3.8: Návrh stránky seznamu barev



---

# Implementace

## 4.1 Použité technologie

### 4.1.1 Framework Symfony

Symfony je sada znovupoužitelných komponent a PHP framework pro tvorbu webových projektů [44]. Jedná se o jeden z celosvětově nejpoužívanějších nástrojů pro tvorbu webových stránek [45].

### 4.1.2 Komponenty

Při implementaci aplikace byla kromě základních komponent frameworku Symfony využita celá řada dalších komponent třetích stran:

**KnpmenuBundle** integruje do Symfony PHP knihovnu Knpmenu, která usnadňuje vytváření navigací v aplikaci. [46]

**KnppaginatorBundle** integruje do Symfony PHP knihovnu KnppaginatorComponent, která usnadňuje stránkování kolekcí. [47]

**LiipImagineBundle** poskytuje nástroj pro manipulaci s obrázky. Umožňuje vytváření miniatur, změnu velikosti obrázku a další transformace. [48]

**OneupFlysystemBundle** je abstrakce souborového systému, která umožňuje snadnou práci se soubory a jednoduchou výměnu lokálního souborového systému za vzdálený. [49]

**FOSElasticaBundle** poskytuje integraci knihovny Elastica [50], zajišťující komunikaci s vyhledávačem Elasticsearch [51] do Symfony. [52]

**DoctrineCacheBundle** umožňuje využívání různých cachovacích systémů poskytovaných knihovnou Doctrine Cache. [53]

### 4.1.3 Databáze

Framework Symfony používá pro persistenci dat ORM Doctrine 2. Dotazy na databázi se provádějí přes entity manager a třídy repozitářů (ve kterých jsou definované složitější dotazy), které entity manager využívají. V Doctrine 2 mají třídy anotované atributy, které se mají persistovat. Pomocí anotací jsou definované vztahy mezi třídami, datové typy sloupců databáze a mapování.

Pro vývoj a běh aplikace jsem použil MySQL, který je druhým nejpoužívanějším databázovým serverem [54]. Zároveň by však vzhledem k abstrakci poskytované knihovnou Doctrine 2 bylo možné zvolit i jiný databázový server (s nutností zohlednění při implementaci načítání CSV tabulek Rebrickable, popsané v sekci 4.3.2).

### 4.1.4 Frontend

Pro usnadnění práce při tvorbě projektu poskytujícího uživatelské rozhraní je vhodné využití frontend frameworku. Frameworky definují vzhled a chování nezákladnějších, často používaných prvků.

Pro tvorbu práce jsem se rozhodoval kromě využití pro mě již známého frameworku Bootstrap 3 také nad možností využití dalších nejpoužívanějších alternativ – Foundation a Semantic UI [55]. Pro mou aplikaci jsem se nakonec rozhodnul využít frameworku Semantic UI. Semantic UI při definování tříd používá přirozené názvy vycházející z angličtiny, tudíž je jeho použití velmi intuitivní a naučení používání by mělo být velmi snadné. V ukázce kódu 4.1



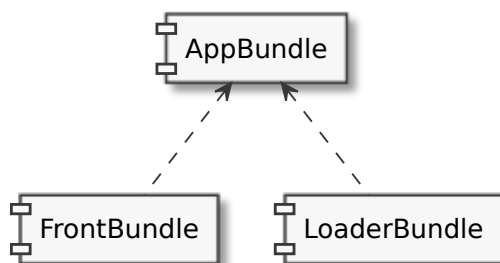
je možné vidět příklad definice mřížky se třemi sloupečky pomocí Semantic UI.

```
1 <div class="ui three column grid">
2   <div class="row">
3     <div class="column"></div>
4     <div class="column"></div>
5     <div class="column"></div>
6   </div>
7 </div>
```

**Ukázka kódu 4.1:** Ukázka definice mřížky pomocí Semantic UI

## 4.2 Architektura

Architektura aplikace je členěna do tří balíků reprezentujících logicky odlišné funkční celky. Hierarchie balíků je znázorněna na diagramu 4.1.



**Obrázek 4.1:** Architektura komponent

**AppBundle** je základním balíkem aplikace obsahujícím implementaci datové vrstvy aplikace, zodpovědné za persistenci dat a jejich zpřístupnění. Dále obsahuje implementaci klienta pro přístup k datům Brickset přes API a implementaci business logiky aplikace.

**FrontBundle** obsahuje implementaci uživatelského rozhraní aplikace v podobě *Controller* tříd a *Twig* šablon sloužících k zobrazení dat uživateli.

**LoaderBundle** zastřešuje logiku aplikace pro načítání součástí knihovny LDraw a databáze Rebrickable z CSV souborů za pomoci konzolových příkazů [41].

### 4.3 Načítání dat

#### 4.3.1 Načítání knihovny LDraw

Mechanismus načítání a aktualizace knihovny LDraw je implementován s využitím Symfony komponenty Console. Administrátor aplikace může spouštět jednotlivé příkazy podle instrukcí uvedených v dodatku B. Na základě těchto příkazů jsou následně načítána data do databáze a prováděn převod formátu. V ukázce kódu 4.2 je možné vidět volání příkazu pro načtení knihovny LDraw.

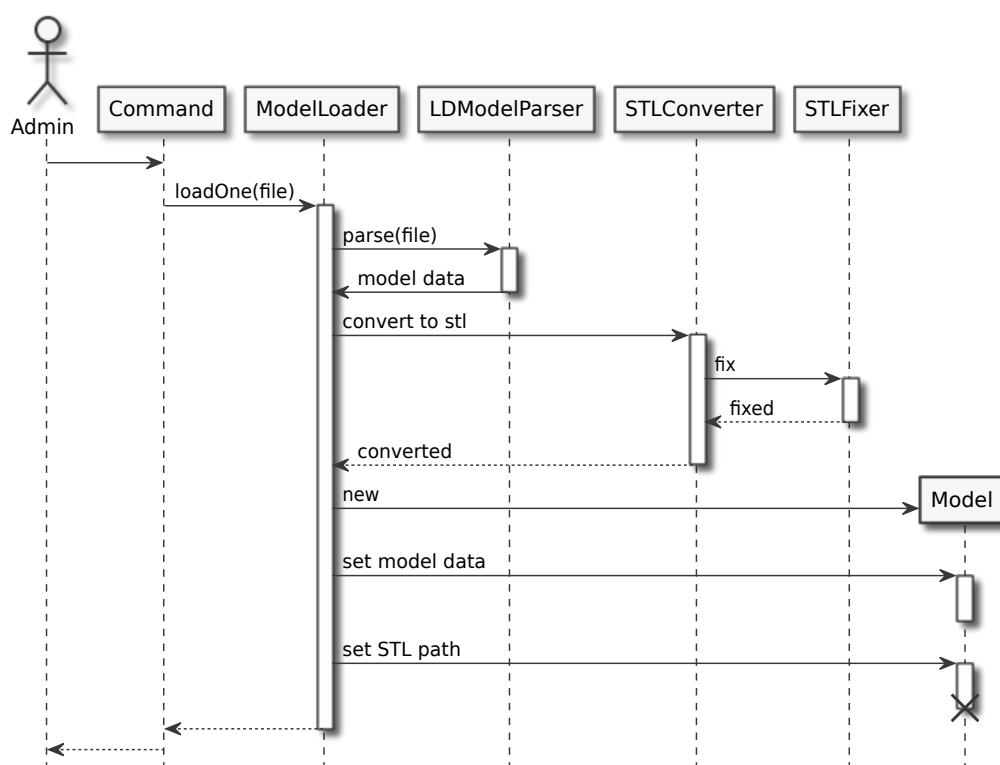
```
1 $ bin/console app:load:ldraw --all
2 -----
3 Downloading LDraw library
4 -----
5 Loading file from: http://www.ldraw.org/library/updates/complete.zip
6 [33.93 MB / 33.93 MB] [=====] 100% (1 min/1 min)
7 LDraw library downloaded
8 -----
9 Loading models from LDraw library: /tmp/printabrick.HIH6lN/ldraw/
10 -----
11 [9782 / 9782] [=====] 100% (6 mins/6 mins)
12 Done with "0" errors.
```

**Ukázka kódu 4.2:** Ukázka příkazu pro načtení knihovny LDraw

Aplikace umožňuje načítání a aktualizaci celé knihovny i jednotlivých součástí. Proces načítání jednotlivé součástky je znázorněn na diagramu 4.2 a podrobněji popsán v následujících podsekcích.

#### Parsování součástí LDraw

Parsování součástí formátu LDraw má na starost třída *LDMoDelParser*. Třída na základě specifikace formátu LDraw představeného v sekci 1.1.1 parsuje



Obrázek 4.2: Diagram načítání součástky knihovny LDraw

hlavičku souboru i reference na podsoučástky. Data jsou následně v podobě asociativního pole vrácena k dalšímu zpracování.

### Převod formátu

Pro spouštění programů zajišťujících převod formátu LDraw na formát *STL* jsem využil komponenty Symfony Process [56]. Ta umožňuje spouštění příkazů přímo z kódu PHP (ukázka kódu 4.3).

Převod formátu je nejprve zajištěn třídou *STLConverter*, která zastřešuje volání programu LDView (představeného v sekci 1.2.2). Následně třída *STLFixer* provede opravu *STL* souboru pomocí volání programu ADMesh (představeného v sekci 1.2.3).

## 4. IMPLEMENTACE

---

```
1  /**
2   * Call LDView process with $arguments.
3   *
4   * @param array $arguments
5   */
6  private function runLDView(array $arguments)
7  {
8      $builder = new ProcessBuilder();
9      $process = $builder
10         ->setPrefix($this->ldview)
11         ->setArguments($arguments)
12         ->getProcess();
13
14     $process->run();
15
16     if (!$process->isSuccessful()) {
17         throw new ProcessFailedException($process);
18     }
19 }
```

**Ukázka kódu 4.3:** Ukázka použití komponenty Symfony Process

### Logování

Protože během procesu načítání knihovny dochází ke zpracování velkého množství dat a ne vždy se dá spoléhat na validitu vstupu, bylo třeba implementovat mechanismus logování událostí. Významné události a chyby jsou logovány pomocí knihovny Monolog [57]. Ukázku přidávání události do logu je možné vidět v ukázce kódu 4.4 a podobu souboru logu v ukázce kódu 4.5.

```
1  try {
2      $modelArray = $this->ldModelParser->parse(file_get_contents($file));
3  } catch (\Exception $e) {
4      $this->logger->error($e->getMessage(), [$file]);
5
6      return null;
7  }
```

**Ukázka kódu 4.4:** Ukázka použití knihovny Monolog

```

1 [2017-05-27 22:54:05] loader.INFO: Model skipped.
  ↳ {"number":"004695a","type":"Sticker"} []
2 [2017-05-27 22:54:05] loader.INFO: Model skipped.
  ↳ {"number":"004695b","type":"Sticker"} []
3 [2017-05-27 22:54:05] event.ERROR: Subpart file not found
  ↳ {"subpart":"2-4cyli","model":"3820"} []

```

**Ukázka kódu 4.5:** Ukázka logu načítání

### 4.3.2 Načítání dat Rebrickable

Načítání velkého množství dat z CSV souborů do aplikace může být dle [58] velmi problematické. Načítání databáze Rebrickable má na starost služba *RebrickableLoader*. Ta pomocí volání MySQL funkce `LOAD DATA LOCAL INFILE` (viz ukázka kódu 4.6) načítá velmi rychle CSV soubory Rebrickable.

```

1 private function loadCsvFile($file, $table, $columns)
2 {
3     $query = sprintf("LOAD DATA LOCAL INFILE '%s'
4         REPLACE INTO TABLE %s
5         CHARACTER SET UTF8
6         FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY '\"'
7         LINES TERMINATED BY '\\n'
8         IGNORE 1 LINES %s", addslashes($file), $table, $columns);
9
10    return $this->em->getConnection()->prepare($query)->execute();
11 }

```

**Ukázka kódu 4.6:** Ukázka načítání tabulek CSV

### 4.3.3 Mapování LDraw a Rebrickable

Jedním z problémů, který vyvstává při využití různých zdrojů dat pro obdržení 3D modelů součástek a databáze stavebnic, je vzájemné provázání těchto dat. Namapování součástek má na starost třída *RelationMapper*, která provádí přiřazení součástek z knihovny LDraw k součástkám z databáze Rebrickable.

Protože obě služby používají lehce odlišný systém číslování součástek (i přes fakt, že číslování Rebrickable vychází z LDraw), nelze při zohlednění všech

pravidel LDraw [59] i Rebrickable [60] automaticky rozpoznat vazby všech součástí.

K umožnění manuálního přiřazení součástí LDraw k součástkám Rebrickable jsem implementoval mechanismus umožňující definování vazeb pomocí YAML Ain't Markup Language (YAML) souboru *app/Resources/reasons/part\_model.yml*. K načtení souboru je využita komponenta Symfony Yaml Component [61]. Mapování součástky na model je možné vidět v ukázce kódu 4.7. Tyto vazby jsou následně zohledněny při běhu příkazu pro spárování součástí.

```
1 #rebrickable_part_ID: ldraw_model_ID
2 970d00: 970c00
3 298c04: 4592c01
```

**Ukázka kódu 4.7:** Ukázka part\_model.yml

### 4.3.4 API Brickset

Dalším zdrojem dat využívaným pro obdržení informací o stavebnicích je API Brickset. Jak již bylo zmíněno v sekci 1.1.2, API Brickset je implementováno jako služba postavená nad protokolem SOAP. V PHP je pro komunikaci se SOAP servery k dispozici třída *SOAPClient* [62], která za nás obstarává komunikaci na základě definice v podobě Web Services Definition Language (WSDL) souboru.

Při získávání statických dat přes API je vhodné využít možnosti cachování k minimalizaci počtu dotazů na server a k urychlení načítání dat. K tomuto jsem využil komponenty *DoctrineCacheBundle*. V ukázce kódu 4.8 je zobrazena metoda *getSetInstructions* třídy *BricksetManager*, která načítá instrukce k postavení stavebnice z cache (pokud se zde data již vyskytují) nebo pomocí Brickset API.

## 4.4 3D náhled

3D náhled součástí LEGO je implementován s využitím javascriptového frameworku Three.js. Tento framework disponuje velmi rozsáhlou a přehlednou

```

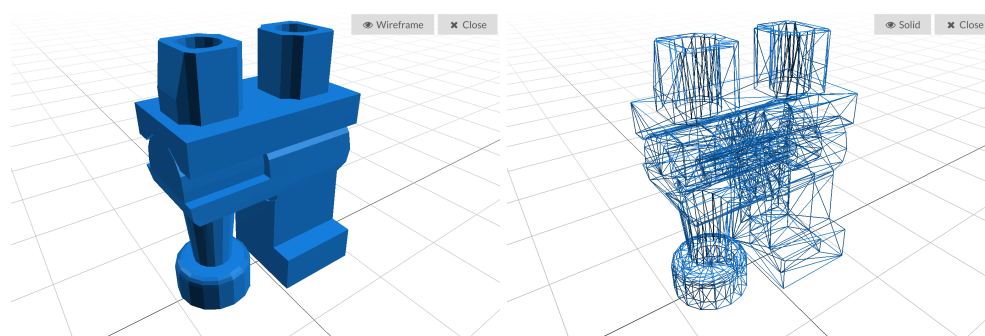
1 public function getSetInstructions($id)
2 {
3     $key = 'instructions-'. $id;
4
5     if (!$data = unserialize($this->cache->fetch($key))) {
6         $data = $this->bricksetClient->getInstructions($id);
7         $this->cache->save($key, serialize($data),
8             ↪ self::CACHE_LIFETIME);
9     }
10
11     return $data;
12 }

```

**Ukázka kódu 4.8:** Ukázka načtení instrukcí stavebnice

dokumentací, včetně ukázek použití a je nejpoužívanějším javascriptovým WebGL frameworkem [63]. Na základě těchto důvodů bylo rozhodnuto o jeho využití v tomto projektu.

Na obrázku 4.3 je možné vidět ukázkou a možnosti implementovaného 3D náhledu.



**Obrázek 4.3:** 3D náhled součástky

## 4.5 Stažení součástek

Poskytnutí ZIP archivů s modely součástek má na starost služba *ZipService*. Ta s využitím knihovny třídy PHP *ZIPArchive* umožňuje vytvoření archivů, které jsou následně poskytnuté uživateli ke stažení.

V ukázce adresářové struktury 4.1 je možné vidět strukturu archivu stavebnice roztríděné podle barev.

```
2000416-1_Duck(Multi-Color)
├── Red_ (#C91A09) ..... složka se součástkami v červené barvě
│   ├── 3021_ (2x).stl
│   └── Yellow_ (#F2CD37) ..... složka se součástkami ve žluté barvě
│       ├── 3001_ (1x).stl
│       ├── 3003_ (2x).stl
│       └── 3004_ (1x).stl
└── LICENSE.txt ..... licenční ujednání se seznamem autorů součástí
```

**Adresářová struktura 4.1:** Obsah archivu 2000416-1\_Duck(Multi-Color).zip

## 4.6 Obrázky součástek a stavebnic

### 4.6.1 Renderování součástek

Jak již bylo zmíněno v sekci 1.3, Rebrickable poskytuje většinu obrázků vyrenderovaných součástek z knihovny LDraw volně ke stažení. Pro zbytek součástek jsem implementoval mechanismus vykreslení s využitím programů stl2pov [37] a POV-Ray [38].

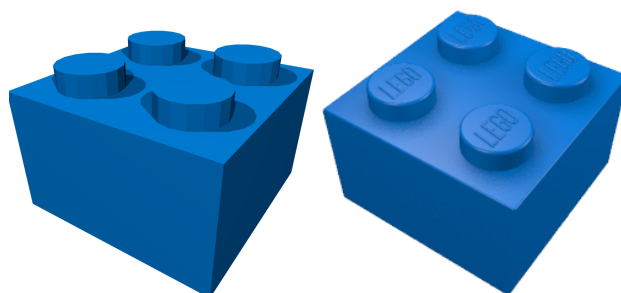
Vykreslení součástek zajišťuje třída *STLRenderer*. Třída nejprve připraví kompletní scénu, obsahující součástku převedenou pomocí stl2pov do POV-Ray formátu. Dále scéna obsahuje definici pozice světla a kamery, specifikované v souboru *app/Resources/povray\_layout/layout.tmpl*. Následně je pomocí volání programu POV-Ray vykreslen obrázek.

Porovnání vyrenderované součástky s obrázkem součástky z Rebrickable je možné vidět na obrázku 4.4.

### 4.6.2 Cachování obrázků

Pro urychlení načítání webových stránek jsem implementoval mechanismus cachování miniatur obrázků součástek a stavebnic. K práci s obrázky v Symphony slouží komponenta LiipImagineBundle [48]. LiipImagineBundle umožňuje





**Obrázek 4.4:** Porovnání renderu součástky a Rebrickable [8]

definování různých filtrů a transformací, které jsou aplikovány na zobrazované obrázky. Transformované obrázky jsou následně uloženy do cache, ze které jsou při dalším požadavku načtené.

Proces vytváření transformovaných obrázků se skládá ze tří kroků:

1. načtení obrázku,
2. aplikace filtrů,
3. uložení transformovaného obrázku do cache.

### **Načtení obrázku**

Načítání obrázků mají na starosti třídy *PartImageLoader* a *SetImageLoader*, které implementují rozhraní *LoaderInterface* a na základě vstupu vrací binární obsah obrázku.

### **Aplikace filtrů**

Filtry jsou na obrázky aplikovány na základě konfigurace (ukázka kódu 4.9).

## **4.7 Vyhledávání**

Podle funkčního požadavku *FP-1* musí aplikace umožňovat vyhledávání a procházení součástek a stavebnic. Jak již bylo zmíněno v podkapitole 4.1, k realizaci tohoto požadavku jsem využil fulltextového vyhledávače Elasticsearch [51]. Pro implementaci konkrétně balíček FOSElasticaBundle [52], který integruje PHP

```
1 liip_imagine:
2     filter_sets:
3         part_min:
4             quality: 90
5             data_loader: part_image_loader
6             cache: ~
7             default_image: "/resources/images/noimage_min.png"
8             filters:
9                 upscale: { min: [230, 230] }
10                thumbnail: { size: [230, 230], mode: inset,
11                            ↪ allow_upscale: true }
12                background: { size: [250, 250], position: center, color:
13                             ↪ '#FFFFFF' }
```

**Ukázka kódu 4.9:** Ukázka konfigurace filtru LiipImagineBundle

knihovnu pro komunikaci s Elasticsearch serverem přes Representational State Transfer (REST) API do frameworku Symfony.

### 4.7.1 Nastavení indexování

Vyhledávání a filtrování kolekce probíhá na základě nastavených pravidel indexování entit. Nastavení indexování entity *Set* je možné vidět v ukázce kódu 4.10. Nastavení probíhá pomocí konfiguračního souboru ve formátu YAML a určuje atributy společně s metodou indexace.

### 4.7.2 Vyhledávání

Vyhledávání je v aplikaci zajišťováno službou *SearchService*. Služba na základě formuláře s využitím repozitářů připraví dotaz, který je odeslán na vyhledávací server Elasticsearch v podobě zobrazené v ukázce kódu 4.11. Následně jsou data získaná z odpovědi automaticky namapována na entity s využitím Doctrine 2 a je vrácen výsledek.

## 4.8 Náhled pro sociální síť

Významným zdrojem návštěvnosti stránek mohou v dnešní době být i sociální síť. Aby byl sdílený odkaz na stránku dostatečně zajímavý je třeba stránky

```
1 fos_elastica:
2   indexes:
3     app:
4       types:
5         set:
6           mappings:
7             id:
8               type: "keyword"
9             fields:
10              ngrams: { type: 'text', analyzer: id_analyzer }
11            name:
12              analyzer: name_analyzer
13              search_analyzer: "standard"
14            year: { type: integer }
15            partCount: { type: integer }
16            theme:
17              type: "object"
18              properties: { id: ~ }
19          persistence:
20            driver: orm
21            model: AppBundle\Entity\Rebrickable\Set
22            ...
23            repository: AppBundle\Repository\Search\SetRepository
24            ...
```

**Ukázka kódu 4.10:** Ukázka nastavení mapování entity *Set*

rozšířit o patřičné meta informace. Facebook a většina sociálních sítí ke specifikování náhledu využívá protokol Open Graph představený firmou Facebook v roce 2010 [64].

Vše co je pro vytvoření náhledu pro sociální sítě potřeba, je přidání patřičných Open Graph meta tagů do hlavičky stránky. Pro tento účel jsem vytvořil znovupoužitelné makro šablonovacího systému *Twig* (ukázka kódu 4.12). Výsledný náhled při sdílení stránky stavebnice na sociální síti Facebook je zobrazen na obrázku 6.2.

## 4. IMPLEMENTACE

---

```
1 {
2   "query": {
3     "bool": {
4       "must": [{
5         "multi_match": {
6           "fields": [ "name", "id", "id.ngrams" ],
7           "query": "falcon",
8           "fuzziness": 0.7,
9           "minimum_should_match": "80%",
10          "operator": "and"
11        }
12      ]},
13     "filter": [
14       { "match": { "theme.id" : 158 }},
15       { "range": { "partCount" : { "gte" : "0", "lte" : "5922" } }},
16       { "range": { "year" : { "gte" : "1950", "lte" : "2017" } }}
17     ]
18   }
19 },
20 "size": 500
21 }
```

**Ukázka kódu 4.11:** Ukázka JSON dotazu Elasticsearch

```
1 {% macro meta(title,url,description,image,imageHeight,imageWidth) %}
2   <meta property="og:title" content="{{ title }}">
3   <meta property="og:description" content="{{ description }}">
4   <meta property="og:url" content="{{ url }}">
5   <meta property="og:image" content="{{ image }}">
6   <meta property="og:image:width" content="{{ imageWidth }}">
7   <meta property="og:image:height" content="{{ imageHeight }}">
8   <meta property="og:site_name" content="{{ name }}">
9
10  <meta property="twitter:card" content="summary">
11 {% endmacro %}
```

**Ukázka kódu 4.12:** *Twig* makro pro přidání Open Graph meta tagů



**Obrázek 4.5:** Náhled sdílení stavebnice 2000416-1 Duck



---

# Testování a nasazení

## 5.1 Testování

V této sekci jsou popsány způsoby, jakými byla aplikace testována. K testování bylo využito jak metodik statické analýzy kvality kódu, tak i automatické testování pomocí jednotkových a integračních testů.

Statická analýza kvality kódu byla provedena nástrojem PHPSpec [65]. Testování pomocí jednotkových a integračních testů bylo provedeno nástrojem PHPUnit [66]. Výstup testování pomocí PHPUnit se nachází v ukázce kódu 5.1.

```
1 $ php vendor/bin/phpunit
2 PHPUnit 6.2.2 by Sebastian Bergmann and contributors.
3
4 ..... 50 / 121 ( 41%)
5 ..... 100 / 121 ( 82%)
6 ..... 121 / 121 (100%)
7
8 Time: 41.4 seconds, Memory: 102.00MB
9
10 OK (121 tests, 191 assertions)
```

**Ukázka kódu 5.1:** Výstup PHPUnit testů

### 5.1.1 Jednotkové testy

Jednotkové testy jsou zaměřeny na dílčí části aplikace. Měly by být maximálně izolované od ostatních částí. Hlavní podstatou jednotkových testů je kontrola jedné komponenty, bez ohledu na funkčnost ostatních komponent, u kterých předpokládáme, že fungují korektně. Z tohoto důvodu se při jednotkovém testování ostatní komponenty nahrazují falešnými objekty. [67]

Implementovaný jednotkový test příkazu pro načtení knihovny LDraw je možné vidět v ukázce kódu 5.2.

```
1 public function testLoadAll()
2 {
3     self::bootKernel();
4     $application = new Application(self::$kernel);
5
6     $modelLoader = $this->createMock(ModelLoader::class);
7     $modelLoader->expects($this->once())->method('loadAll');
8
9     $application->add(new LoadLdrawCommand(null, $modelLoader));
10
11     $command = $application->find('app:load:ldraw');
12
13     $tester = new CommandTester($command);
14     $tester->execute(
15         ['--ldraw' => 'path', '--all' => true]
16     );
17 }
```

**Ukázka kódu 5.2:** Příklad jednotkového testu příkazu načtení knihovny LDraw



## 5.1.2 Integrační testy

Integrační testy kontrolují, jak spolu třídy nebo moduly navzájem spolupracují. Testují, zda jeden modul splňuje požadavky jiného. Na rozdíl od jednotkových testů mohou využívat databázi a souborový systém. [67]

Veškeré implementované integrační testy aplikace rozšiřují třídu *BaseTest*, která připravuje prostředí pro běh integračních testů. Aby bylo možné integrační testy provést, jsou vytvořena testovací data. V Symfony je k tomuto účelu určena komponenta *DoctrineFixturesBundle* [68].













Pomocí integračních testů je testována veškerá business logika aplikace, včetně testů volání programů třetích stran, využívaných k práci se soubory *STL*.

```
1 public function testConvertToStl()
2     {
3         $adapter = new Local(__DIR__.'/fixtures/ldraw');
4         $ldrawLibraryContext = new Filesystem($adapter);
5         $this->stlConverter-
6             ↪ >setLDrawLibraryContext($ldrawLibraryContext);
7
8         $this->assertNotNull($this->stlConverter-
9             ↪ >datToStl(__DIR__.'/fixtures/ldraw/parts/983.dat'));
10
11         // Check if stl file exists
12         $this->assertTrue($this->filesystem->has('models/983.stl'));
13     }
```

**Ukázka kódu 5.3:** Příklad integračního testu

### 5.1.3 Pokrytí kódu testy

Na obrázku 5.1 je možné vidět pokrytí kódu testy, vygenerované pomocí PHPUnit.

	Code Coverage								
	Progress	Lines	Count	Progress	Functions and Methods	Count	Progress	Classes and Traits	Count
Total		85.59%	1354 / 1582		72.27%	245 / 339		55.13%	43 / 78
AppBundle		76.09%	490 / 644		67.86%	152 / 224		48.94%	23 / 47
FrontBundle		91.23%	312 / 342		80.00%	36 / 45		70.59%	12 / 17
LoaderBundle		92.62%	552 / 596		81.43%	57 / 70		57.14%	8 / 14

Obrázek 5.1: Pokrytí kódu testy

## 5.2 Nasazení

Aplikace byla nasazena na školní Virtuální privátní server (VPS), který byl speciálně zřízen pro potřeby mé aplikace. Na serveru běží operační systém CentOS 7 [69]. Vzhledem k tomu, že jsem obdržel administrátorská práva k systému, bylo možné nainstalovat všechny potřebné závislosti, kterých aplikace využívá.

Při instalaci bylo postupováno podle instalační příručky (příloha B), včetně úspěšného běhu testů.

Aplikace je nasazena a připravena k používání na adrese <http://printabrick.org>.



---

## Právní aspekty

V této kapitole budou rozebrány právní aspekty výsledné aplikace, aby mohlo být rozhodnuto o možnosti zveřejnění do sítě Internet.

### 6.1 Zdroje dat

Při analyzování právních aspektů aplikace je v první řadě nutné zhodnotit veškeré využití zdroje dat v aplikaci.

#### 6.1.1 LDraw

Jak již bylo uvedeno v sekci 1.1.1, všechny součástky knihovny LDraw jsou sdíleny pod licencí CC-BY [21].

Aby byla dodržena licence knihovny LDraw při sdílení 3D modelů součástí, je nutné spolu se staženými soubory jednoznačně uvést autora dle licence. Toho je docíleno přidáním souboru LICENSE.txt do sdíleného archivu. Příklad tohoto souboru je možné vidět v ukázce kódu 6.1 zobrazující licenční soubor, zahrnutý v archivu stavebnice *2000416-1 Duck*. Dále je autor společně s licencí uveden na stránce detailu součástky.

#### 6.1.2 Rebrickable

Jak již bylo řečeno v kapitole 1.1.3, Rebrickable poskytuje svá data volně bez nároků a omezení. Autoři Rebrickable pouze žádají o uvedení původu

```
1 All stl files in this archive were converted by LDView from
2 official LDraw Library http://www.ldraw.org/
3
4 Files are licensed under the Creative Commons - Attribution license.
5 http://creativecommons.org/licenses/by/2.0/
6
7 Attribution:
8
9 3021 - "Plate 2 x 3" by James Jessiman
10 3001 - "Brick 2 x 4" by James Jessiman
11 3003 - "Brick 2 x 2" by James Jessiman
12 3004 - "Brick 1 x 2" by James Jessiman
```

### Ukázka kódu 6.1: Ukázka souboru LICENSE.txt

využívaných dat. Toto přání jsem splnil přidáním prohlášení do patičky každé stránky.

### 6.1.3 Brickset

API služby Brickset je volně dostupné k vytváření doplňkových webových stránek nebo služeb bez uvedení omezujících podmínek. [70]

## 6.2 Obchodní značka LEGO

Obchodní značka LEGO může být použita v případě dodržení určitých podmínek i na neoficiálních webových stránkách, na kterých jsou zobrazovány nebo diskutovány prvky LEGO. Instrukce vydané společností LEGO jsou dostupné na [71].

Shrnutí instrukcí:

- logo LEGO nesmí být použito na neoficiální webové stránce,
- obchodní značka LEGO by se měla při každém použití vždy zobrazovat se symbolem ®,
- musí být zřejmé, že stránka není sponzorována nebo autorizována společností LEGO,

- obchodní značka nesmí být zvýrazňována,
- značka LEGO nemůže být použita v internetové adrese,
- stránka by měla obsahovat prohlášení o popření právního nároku.

Všechny tyto instrukce byly zohledněny při tvorbě aplikace, včetně přidání prohlášení o popření právního nároku do patičky (viz ukázka kódu 6.2). Při tvorbě tohoto prohlášení jsem se inspiroval prohlášením z patičky stránky služby Brickset [23].

- 1 LEGO®, the LEGO logo, the Minifigure, and the Brick and Knob
- 2 configurations are trademarks of the LEGO Group of Companies which
- 3 does not sponsor, authorize, or endorse this site.

**Ukázka kódu 6.2:** Prohlášení o popření právního nároku v patičce stránky

## 6.3 Závěr

Při implementaci byly zohledněny licenční podmínky všech služeb, využitých balíčků i instrukce společnosti LEGO pro neoficiální webové stránky a nebyly shledány žádné překážky k zveřejnění. Výsledná webová aplikace tedy může být zveřejněna do sítě Internet.





---

## Závěr

Cílem práce bylo navrhnout a implementovat webovou aplikaci umožňující stažení 3D modelů dílů LEGO. Tento cíl byl úspěšně naplněn včetně zveřejnění aplikace do sítě internet.

V rešeršní části práce byly nejprve představeny služby poskytující data o součástkách a stavebnicích LEGO, dále byly představeny programy pro převod formátu LDraw na formát *STL*.

Na základě rešerše byl následně proveden návrh architektury aplikace a uživatelského rozhraní, ze kterého bylo vycházeno při samotné implementaci.

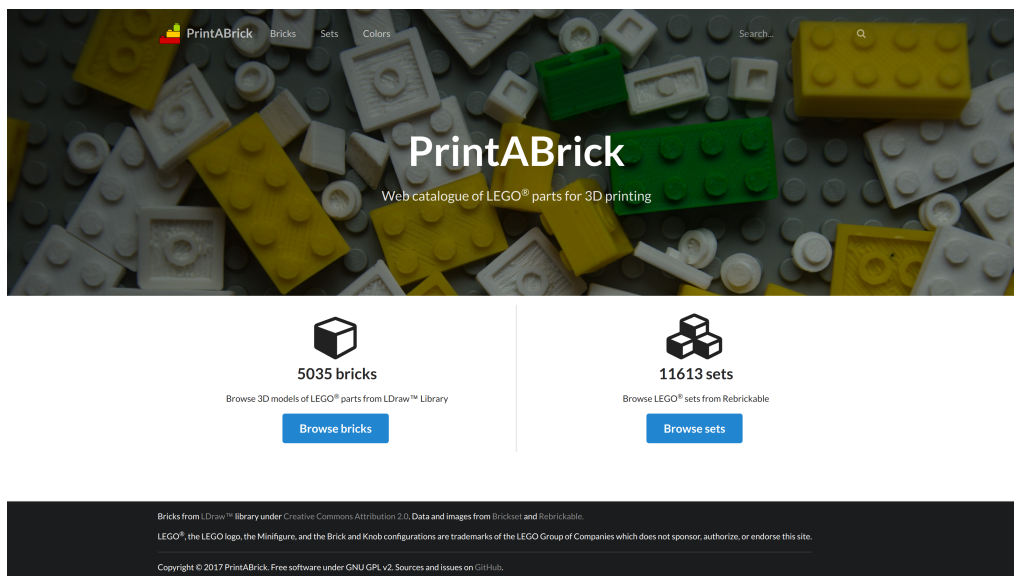
Instance aplikace je nasazena a připravena k používání na adrese <http://printabrick.org>.

Zdrojový kód aplikace je zveřejněn pod open-source licencí GPLv2 [34] a veřejně dostupný na <https://github.com/hubnedav/PrintABrick>.

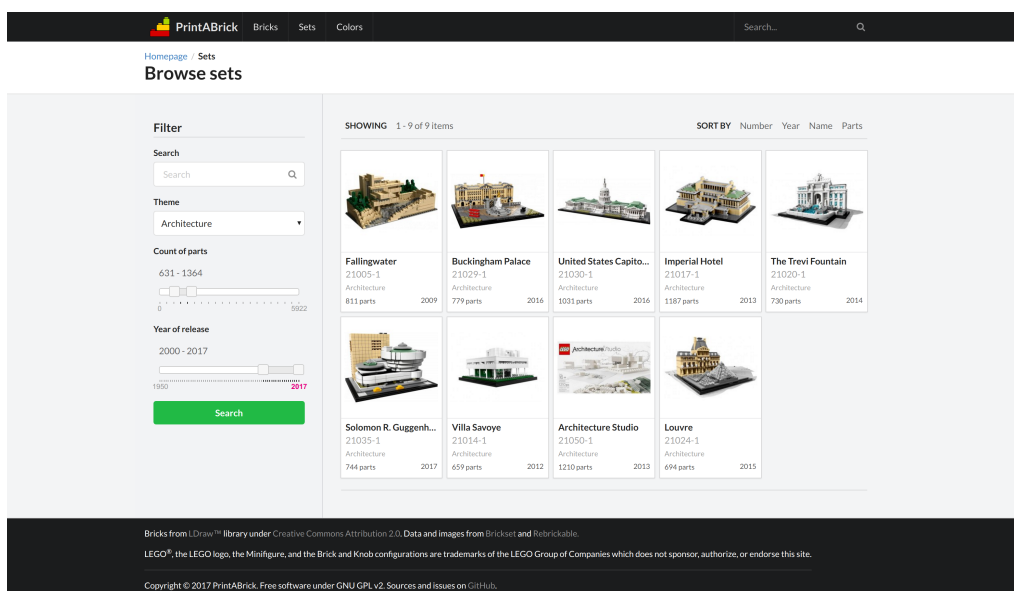
### **Možnosti dalšího rozvoje**

V rámci dalšího rozvoje aplikace vidím následující možnosti:

- výpočet odhadu množství materiálu potřebného k tisku,
- využití uživatelských dat pro správu sbírek přes API Rebrickable,
- provedení uživatelského testování s následnou opravou zjištěných závad.



Obrázek 6.1: Snímek obrazovky – domovská stránka



Obrázek 6.2: Snímek obrazovky – výpis stavebnic

---

## Zdroje

1. LDRAW.ORG. *LDraw.org Centralised LDraw Resources* [online]. 2002 [cit. 2017-06-11]. Dostupné z: <http://www.ldraw.org/>.
2. CLAGUE, K.; AGULLO, M.; HASSING, L.C. *LEGO Software Power Tools: With LDraw, MLCad, and LPub*. Syngress, 2003. ISBN 9781931836760. Dostupné také z: <https://books.google.cz/books?id=cDbda-vV3NIC>.
3. JESSIMAN, James. *Car* [Knihovna LDraw].
4. LDRAW.ORG. *LDraw.org Standards: File Format Standards* [online] [cit. 2017-06-11]. Dostupné z: <http://www.ldraw.org/documentation/ldraw-org-file-format-standards>.
5. BLISS, S.; POBURSKY, O.; TSCHAGER, W. et al. *LDraw.org Standards: File Format 1.0.2* [online]. 2007 [cit. 2017-06-11]. Dostupné z: <http://www.ldraw.org/article/218>.
6. LDRAW.ORG. *LDraw Parts: Colour Definitions* [online] [cit. 2017-06-19]. Dostupné z: <http://www.ldraw.org/article/547.html>.
7. JESSIMAN, James. *Brick 2 x 2* [online]. 2002 [cit. 2017-06-14]. Dostupné z: <http://www.ldraw.org/library/official/parts/3003.dat>.
8. REBRICKABLE, Pty Ltd. *LEGO PART 3003 Brick 2 x 2* [online] [cit. 2017-06-12]. Dostupné z: <https://rebrickable.com/media/parts/ldraw/-1/3003.png>.
9. HOWARD, W. et al. *LDraw Parts: Official Parts* [online] [cit. 2017-06-16]. Dostupné z: <http://www.ldraw.org/article/13>.

10. HOWARD, W. et al. *LDraw.org Standards: File Format Restrictions for the Official Library* [online] [cit. 2017-06-13]. Dostupné z: <http://www.ldraw.org/article/512>.
11. TSCHAGER, W.; COBBS, T.; HOWARD, W. *LDraw.org Standards: Official Library Header Specification* [online]. 2007 [cit. 2017-06-11]. Dostupné z: <http://www.ldraw.org/article/398>.
12. LDRAW.ORG. *LDraw.org Standards: Official Library Specification for Sticker Parts* [online]. 2016 [cit. 2017-06-15]. Dostupné z: <http://www.ldraw.org/article/339>.
13. LDRAW.ORG. *LLDraw Primitives Reference* [online] [cit. 2017-06-16]. Dostupné z: <http://www.ldraw.org/library/primref/>.
14. REBRICKABLE, Pty Ltd. *LEGO PART 3001 Brick 2 x 4* [online] [cit. 2017-06-15]. Dostupné z: <https://rebrickable.com/media/parts/ldraw/-1/3001.png>.
15. REBRICKABLE, Pty Ltd. *LEGO PART 3001p0b Brick 2 x 4 with Wavy Mouth and Fangs Print* [online] [cit. 2017-06-15]. Dostupné z: <https://rebrickable.com/media/parts/ldraw/-1/3001p0b.png>.
16. PTADMIN. *Moved to 6141* [online]. 2015 [cit. 2017-06-14]. Dostupné z: <http://www.ldraw.org/library/official/parts/6141.dat>.
17. REBRICKABLE, Pty Ltd. *LEGO PART 970c00 Legs and Hips [Complete Assembly]* [online] [cit. 2017-06-15]. Dostupné z: <https://rebrickable.com/media/parts/ldraw/-1/970c00.png>.
18. REBRICKABLE, Pty Ltd. *LEGO PART 3815 Hips* [online] [cit. 2017-06-15]. Dostupné z: <https://rebrickable.com/media/parts/ldraw/-1/3815.png>.
19. REBRICKABLE, Pty Ltd. *LEGO PART 3816 Leg, Right* [online] [cit. 2017-06-15]. Dostupné z: <https://rebrickable.com/media/parts/ldraw/-1/3816.png>.
20. REBRICKABLE, Pty Ltd. *LEGO PART 3817 Leg, Left* [online] [cit. 2017-06-15]. Dostupné z: <https://rebrickable.com/media/parts/ldraw/-1/3817.png>.

21. COMMONS, Creative. *Creative Commons: Attribution 2.0 Generic* [online] [cit. 2017-06-10]. Dostupné z: <https://creativecommons.org/licenses/by/2.0/>.
22. BRICKSET, Ltd. *Brickset Infographic* [online]. 2016 [cit. 2017-06-12]. Dostupné z: <https://brickset.com/infographic>.
23. BRICKSET, Ltd. *About Brickset* [online] [cit. 2017-06-13]. Dostupné z: <https://brickset.com/about>.
24. BRICKSET, Ltd. *Brickset web services API v2 documentation* [online] [cit. 2017-06-12]. Dostupné z: <https://brickset.com/tools/webservices/v2>.
25. REBRICKABLE, Pty Ltd. *Homepage* [online] [cit. 2017-06-12]. Dostupné z: <http://rebrickable.com/>.
26. REBRICKABLE, Pty Ltd. *About* [online] [cit. 2017-06-12]. Dostupné z: <http://rebrickable.com/about/>.
27. REBRICKABLE, Pty Ltd. *LEGO Database Download* [online] [cit. 2017-06-09]. Dostupné z: <http://rebrickable.com/downloads/>.
28. REBRICKABLE, Pty Ltd. *HELP - Frequently Asked Questions* [online]. 2017 [cit. 2017-06-16]. Dostupné z: <https://rebrickable.com/help/faq/>.
29. REBRICKABLE, Pty Ltd. *Terms of Service* [online] [cit. 2017-06-09]. Dostupné z: <http://rebrickable.com/terms/>.
30. BRICKLINK, Limited. *BrickLink* [online] [cit. 2017-06-16]. Dostupné z: <https://www.bricklink.com/v2/main.page>.
31. BRICK OWL, Ltd. *Brick Owl* [online] [cit. 2017-06-16]. Dostupné z: <http://www.brickowl.com/>.
32. 3DADDFAB. *What is an STL file and is it obsolete?* [online]. 2011 [cit. 2017-06-16]. Dostupné z: <http://3daddfab.com/blog/index.php?/archives/4-What-is-an-STL-file-and-is-it-obsolete.html>.
33. BURNS, Marshall. *StereoLithography Interface Specification* [online] [cit. 2017-06-18]. Dostupné z: [http://fabbers.com/tech/STL\\_format](http://fabbers.com/tech/STL_format).
34. FREE SOFTWARE FOUNDATION, Inc. *GNU General Public License* [online]. 1991 [cit. 2017-06-10]. Dostupné z: <http://www.gnu.org/licenses/old-licenses/gpl-2.0.html>.

35. COBBS, Travis; BARTFAI, Peter. *LDView* [online]. 2016 [cit. 2017-06-13]. Dostupné z: <https://tcobbs.github.io/ldview>.
36. MARTIN, A. *ADMesh* [online] [cit. 2017-06-18]. Dostupné z: <https://github.com/admesh/admesh>.
37. SMITH, Roland. *py-stl / stl2pov* [online]. 2012 [cit. 2017-06-18]. Dostupné z: <https://rsmith.home.xs4all.nl/software/py-stl-stl2pov.html>.
38. OAKS, Hallam. *POV-Ray* [online] [cit. 2017-06-18]. Dostupné z: <http://www.povray.org/>.
39. TIŠNOVSKÝ, Pavel. *Vykreslujeme 3D scény s POV-Ray* [online]. 2008 [cit. 2017-06-28]. Dostupné z: <https://www.root.cz/clanky/vykreslujeme-3d-sceny-s-pov-ray/>.
40. W3TECHS. *Usage of server-side programming languages for websites* [online]. 2017 [cit. 2017-06-28]. Dostupné z: [https://w3techs.com/technologies/overview/programming\\_language/all](https://w3techs.com/technologies/overview/programming_language/all).
41. SENSIOLABS. *Console Commands* [online] [cit. 2017-06-23]. Dostupné z: <https://symfony.com/doc/current/console.html>.
42. SENSIOLABS. *Databases and the Doctrine ORM* [online] [cit. 2017-06-28]. Dostupné z: <https://symfony.com/doc/current/doctrine.html>.
43. LDRAW.ORG. *LDraw.org Contributor Agreement* [online]. 2004 [cit. 2017-06-12]. Dostupné z: <http://www.ldraw.org/article/349>.
44. SENSIOLABS. *Symfony* [online] [cit. 2017-06-24]. Dostupné z: <http://symfony.com/>.
45. KITIPOVA, Iva. *Best PHP Frameworks of 2017: a Beginner's Guide* [online]. 2017 [cit. 2017-06-20]. Dostupné z: <https://www.webhostface.com/blog/best-php-frameworks-of-2017/>.
46. KNPLABS. *Knplabs MenuBundle* [online] [cit. 2017-06-23]. Dostupné z: <https://github.com/Knplabs/KnpMenuBundle>.
47. KNPLABS. *Knplabs PaginatorBundle* [online] [cit. 2017-06-23]. Dostupné z: <https://github.com/Knplabs/KnpPaginatorBundle>.
48. LIIP, Ltd. *LiipImagineBundle* [online] [cit. 2017-06-23]. Dostupné z: <https://github.com/liip/LiipImagineBundle>.

49. 1UP, GmbH. *OneupFlysystemBundle* [online] [cit. 2017-06-23]. Dostupné z: <https://github.com/liip/LiipImagineBundle>.
50. RUFLIN, Nicolas. *Elastica: elasticsearch PHP Client* [online] [cit. 2017-06-23]. Dostupné z: <https://github.com/ruflin/Elastica>.
51. ELASTICSEARCH. *Elasticsearch* [online] [cit. 2017-06-20]. Dostupné z: <https://www.elastic.co/products/elasticsearch>.
52. FRIENDSOFSYMFONY. *FOSElasticaBundle* [online] [cit. 2017-06-23]. Dostupné z: <https://github.com/FriendsOfSymfony/FOSElasticaBundle>.
53. SENSIOLABS. *DoctrineCacheBundle* [online] [cit. 2017-06-26]. Dostupné z: <http://symfony.com/doc/current/bundles/DoctrineCacheBundle/index.html>.
54. SOLID IT, GmbH. *DB-Engines Ranking* [online] [cit. 2017-06-24]. Dostupné z: <https://db-engines.com/en/ranking>.
55. GERCHEV, Ivaylo. *The 5 Most Popular Frontend Frameworks of 2017* [online]. 2017 [cit. 2017-06-25]. Dostupné z: <https://www.sitepoint.com/5-most-popular-frontend-frameworks-compared/>.
56. SENSIOLABS. *The Process Component* [online] [cit. 2017-06-23]. Dostupné z: <http://symfony.com/doc/current/components/process.html>.
57. SENSIOLABS. *Logging with Monolog* [online] [cit. 2017-06-25]. Dostupné z: <http://symfony.com/doc/current/logging.html>.
58. DOMRATCHEV, Anton. *Importing Large CSV files with PHP Part 1: Import using one query* [online]. 2015 [cit. 2017-06-24]. Dostupné z: <https://www.grok-interactive.com/blog/import-large-csv-into-mysql-with-php-part-1/>.
59. DEE, Chris. *FAQ for LDraw Part Numbers* [online]. 2017 [cit. 2017-06-11]. Dostupné z: <http://www.ldraw.org/library/tracker/ref/numberfaq/>.
60. NATHAN. *Part Numbering Changes* [online]. 2013 [cit. 2017-06-16]. Dostupné z: <https://rebrickable.com/forum/index.php?/topic/138-part-numbering-changes/>.
61. SENSIOLABS. *The Yaml Component* [online] [cit. 2017-06-25]. Dostupné z: <https://symfony.com/doc/current/components/yaml.html>.

62. GROUP, The PHP. *SoapClient* [online] [cit. 2017-06-26]. Dostupné z: <http://php.net/manual/en/class.soapclient.php>.
63. DEVINE, Benjamin. *WebGL Framework Comparison* [online] [cit. 2017-06-25]. Dostupné z: <http://bnjm.github.io/WebGL-framework-comparison/>.
64. KUTÁČ, Pavel. *OpenGraph - proniknutí na sociální síť* [online]. 2015 [cit. 2017-06-28]. Dostupné z: <http://www.kutac.cz/blog/weby-a-vse-okolo/opengraph-proniknuti-na-socialni-site/>.
65. PHPSPEC. *phpspec* [online] [cit. 2017-06-29]. Dostupné z: <http://www.phpspec.net/en/stable/>.
66. BERGMANN, Sebastian. *PHPUnit* [online] [cit. 2017-06-29]. Dostupné z: <https://phpunit.de/>.
67. ZAMRZLA, Josef. *Testování a tvorba testovatelného kódu v PHP* [online]. 2012 [cit. 2017-06-28]. Dostupné z: <https://www.zdrojak.cz/clanky/testovani-a- tvorba-testovatelného-kódu-v-php/>.
68. SENSIOLABS. *DoctrineFixturesBundle* [online] [cit. 2017-06-29]. Dostupné z: <http://symfony.com/doc/current/bundles/DoctrineFixturesBundle/index.html>.
69. RED HAT, Inc. *The CentOS Project* [online] [cit. 2017-06-29]. Dostupné z: <https://www.centos.org/>.
70. BRICKSET, Ltd. *Request web services key* [online] [cit. 2017-06-22]. Dostupné z: <https://brickset.com/tools/webservices/requestkey>.
71. LEGO GROUP, The. *Fair Play* [online] [cit. 2017-06-14]. Dostupné z: <https://www.lego.com/en-us/legal/legal-notice/fair-play>.
72. SENSIOLABS. *Requirements for Running Symfony* [online] [cit. 2017-06-28]. Dostupné z: <https://symfony.com/doc/current/reference/requirements.html>.
73. NPM. *npm* [online] [cit. 2017-06-28]. Dostupné z: <https://www.npmjs.com/>.
74. BOWER. *Bower - A package manager for the web* [online] [cit. 2017-06-28]. Dostupné z: <https://bower.io/>.



---

## Seznam použitých zkratk

CAD	Computer Aided Design
PHP	PHP: Hypertext Preprocessor
YAML	YAML Ain't Markup Language
ČVUT	České vysoké učení technické v Praze
FIT	Fakulta informačních technologií
MVC	Model-view-controller
API	Application Programming Interface
CSV	Comma-separated values
STL	Stereolithography format
UTF-8	UCS/Unicode Transformation Format
SOAP	Simple Object Access Protocol
XML	Extensible Markup Language
GPLv2	GNU General Public License, verze 2
CC-BY	Creative Commons Attribution 2.0 Generic
RGB	Red Green Blue
FDM	Fused Deposition Modeling
WebGL	Web Graphics Library
REST	Representational State Transfer
ORM	Object-relational mapping
WSDL	Web Services Definition Language
VPS	Virtuální privátní server



---

# Instalační příručka

Tato instalační příručka je dostupná také v anglické verzi v podobě README.md souboru v přiloženém zdrojovém kódu aplikace.

## PrintABrick

Webový katalog LEGO® dílů k 3D tisku

### Systémové požadavky

- PHP verze 7.0 nebo vyšší
- PHP rozšíření
  - FTP
  - SOAP
  - GD
  - PDO
  - Zip
- *date.timezone* nastaveno v souboru *php.ini*

Zda váš systém splňuje minimální požadavky Symfony můžete zjistit příkazem  
`$ bin/symfony_requirements`

Pro kompletní požadavky navštivte Symfony 3.3 dokumentaci [72].

### Závislosti

- Elasticsearch  $\geq 5$  [51]
- POV-Ray [38]
- stl2pov [37]
- ADMesh
- LDView OSMesa  $\geq 4.2.1$  [35]

### Instalace

#### Backend

1. Ujistěte se, že váš systém splňuje minimální požadavky.
2. Nainstalujte závislosti pomocí Composer, `$ composer install`.

#### Frontend

1. Nainstalujte npm [73] závislosti pomocí `$ npm install`.
2. Nainstalujte bower [74] závislosti pomocí `$ bower install`.
3. Zkompilujte zdroje `$ gulp default [--env production]`.

#### Inicializace

##### Nastavení databáze

1. Nastavte parametry aplikace v souboru `app/config/parameters.yml`.
2. Vygenerujte prázdnou databázi příkazem `$ bin/console doctrine:database:create`.
3. Vytvořte tabulky databáze příkazem `$ bin/console doctrine:schema:create`.
4. Načtěte výchozí data do databáze `$ bin/console doctrine:fixtures:load`.

##### Načtení dat

Kompletní inicializace dat aplikace může být provedena příkazem `$ bin/console app:init`.

---

Tento příkaz se skládá z několika podpříkazů, které mohou být spuštěny samostatně:

1. Načtení součástí knihovny LDraw  
`$ bin/console app:load:ldraw [--ldraw=PATH] [--all] [--file=FILE] [--update].`
2. Načtení databáze Rebrickable  
`$ bin/console app:load:rebrickable.`
3. Spárování součástí knihovny LDraw a databáze Rebrickable  
`$ bin/console app:load:relations.`
4. Stažení/vykreslení obrázků součástí  
`$ bin/console app:load:images [--color=INT] [--rebrickable] [--missing].`
5. Zaindexování Elasticsearch `$ bin/console fos:elastica:populate.`

## Spustění aplikace na lokálním webserveru

Aplikaci můžete spustit na lokálním PHP web serveru příkazem `$ bin/console server:start`.

## Párování součástí

Vazby mezi součástkami z knihovny LDraw a součástkami z databáze Rebrickable jsou automaticky spárovány při běhu příkazu

```
$ bin/console app:load:relations.
```

Vazby mezi součástkami, které nebyly automaticky rozpoznány je možné opravit přidáním odpovídajících ID do souboru

```
app/Resources/relations/part_model.yml.
```

## Testování

Testy využívají zvláštní databázi, která musí být před prvním spuštěním testů vytvořena. Vytvoření testovací databáze:

1. Vygenerujte prázdnou databázi příkazem  
`$ bin/console doctrine:database:create --env=test.`

### 2. Vytvořte tabulky databáze příkazem

```
$ bin/console doctrine:schema:create --env=test.
```

Kompletní systémové testy mohou být spuštěny příkazem `$ phpunit`.

Tyto testy pokrývají základní funkcionality aplikace, včetně testování úspěšnosti volání programů třetích stran, využívaných aplikací.

## Obsah přiloženého média

README.md.....	stručný popis obsahu média
src.....	zdrojové kódy implementace
thesis.....	text práce
latex.....	zdrojová forma práce ve formátu X <sub>Y</sub> L <sub>A</sub> T <sub>E</sub> X
BP_Hubner_David_2017.pdf.....	text práce ve formátu PDF