

Katalóg požiadaviek

Endurance testing

20. novembra 2019

1 Úvod

Katalóg požiadaviek slúži na zaznamenávanie jednotlivých požiadaviek daných zadávateľom. V katalógu sa nachádzajú jednotlivé požiadavky rozdelené podľa potreby – teda či sa jedná o nutné časti tohto projektu, alebo ide skôr o voliteľné požiadavky. Po prečítaní katalógu požiadaviek by mal mať každý predstavu o tom, ako bude daný softvér fungovať. Katalóg požiadaviek je určený všetkým osobám, ktoré sa zapájajú do tvorby systému a taktiež slúži ako záväzná dohoda medzi zadávateľom a vývojármi.

1.1 Cieľ projektu

Cieľom projektu je vytvorenie aplikácie na archiváciu údajov o výrobkoch a testoch životnosti pre spoločnosť BOGE Rubber & Plastics. Tento systém bude uľahčovať archiváciu jednotlivých produktov ako aj spätné získavanie informácií. Do aplikácie budú nahraté všetky poskytnuté údaje, následne sa budú dať nové produkty pridávať cez vytvorený UI. V aplikácii budú uchované všetky diely, všetky dokumenty a súbory, ktoré k nim patria. Aplikácia ich bude organizovať v stromovej štruktúre v súborovom systéme.

1.2 Rozsah systému

Funkcionalita aplikácie pokrýva uchovávanie dielov a ich testov, dokumentácie a fotodokumentácie. Umožňuje manuálne nahrávanie súborov do vopred vytvorenej štruktúry. Ďalšie možné rozšírenie projektu bude umožňovať rozvrhnutie práce jednotlivých testovacích stanovísk, táto požiadavka je však už nad rámec tohto projektu. Súbory do aplikácie vkladá používateľ.

1.3 Slovník pojmov

- (a) **UI** - User interface - je to tzv. užívateľské rozhranie. Ide v podstate o to, čo daný užívateľ vidí a priamo v tom pracuje.
- (b) **Aplikácia** - aplikáciou sa rozumie vývojármi vytvorený softvér + vývojármi vytvorená databáza
- (c) **Zeichnungsnummer** – číslo výkresu, resp. číslo dielu, podľa ktorého sa vytvorí úvodná zložka štruktúry, teda jej názov; je to dvanásťčíselné označenie dielu, v názve štruktúry sa dosádza za xxx.xxx.xxx.xxx

- (d) **Kunde** – značka resp. zákazník, ktorý si objednal testovanie daného dielu, v názve štruktúry sa dosadzuje za názovZakaznika
- (e) **Štruktúra** – prehľadné a logické členenie súborov, ktoré patria jednému testovanému dielu
- (f) **Názov štruktúry** - je názvom štruktúry a zároveň aj názvom (t.j. číslom) dielu
- (g) **Log** – záznam úprav, ktoré sa udiali v databáze, napr. nahratie dokumentácie k nejakému dielu
- (h) **Bezeichnung** - umiestnenie dielu v rámci auta (napr. predná náprava)

1.4 Prehľad kapitol

V druhej kapitole sa čitateľ oboznámi so softvérom a jeho funkčnosťou. V tretej kapitole sú spísané požiadavky zadávateľa na výsledný produkt.

2 Všeobecný opis

Aplikácia bude slúžiť na získavanie informácií o dieloch a im priradeným dokumentom a súborom. Súčasťou aplikácie bude user interface, ktorý bude slúžiť na ľahšiu komunikáciu medzi používateľom a aplikáciou.

2.1 Perspektíva softvéru

Aplikácia bude slúžiť oddeleniu testov životností na ľahšie získavanie jednotlivých údajov ku daným dielom. Zároveň bude slúžiť aj na to, aby aj všetci zamestnanci oddelenia testov životností mali prístup ku všetkým dokumentom, ktoré nahrali do systému ich kolegovia.

2.2 Funkcie softvéru

Užívateľ bude môcť pomocou aplikácie a jej UI pristupovať ku všetkým dielom a ich dokumentom. Zároveň bude môcť do aplikácie nahrávať nové dokumenty, sťahovať staré dokumenty, alebo mazať dokumenty.

2.3 Charakteristika používateľa

Softvér bude slúžiť zamestnancom firmy BOGE Rubber & Plastics - konkrétne oddeleniu testov životností. Používateľmi budú traja zamestnanci tohto oddelenia. Títo používatelia budú mať rovnaké práva v aplikácii. Používatelia sa budú prihlasovať do aplikácie prihlasovacími údajmi - menom a heslom.

2.4 Všeobecné obmedzenia

Pri tvorbe nášho softvéru sa musíme držať už zaužívaného názvoslovie daného oddelenia, a taktiež zachovať používanú hierarchiu.

2.5 Predpoklady a závislosti

Aplikácia bude pracovať na operačnom systéme Windows, na databázu, ktorá je súčasťou aplikácie, zadávateľ odporúča použiť MariaDB. Aplikácia sa bude pripájať do firemnej siete, dáta nebudú prístupné mimo tejto siete. Dohodnutým programovacím jazykom na vývoj aplikácie je Java.

3 Zoznam požiadaviek

3.1 Vytvorenie štruktúry

Aplikácia umožní vytvoriť štruktúru v aplikácii tak, ako to vidíme v prílohe 4.3. Pokiaľ sa v aplikácii nachádza daný diel (to platí pre 3.1.1 aj 3.1.2) a vytvára sa iba jeho verzia (druhé trojčíslenie), tak sa vynecháva úroveň *a1* a začína sa od úrovne *a2*, pričom táto úroveň sa ukladá pod už existujúcu úroveň *a1* v aplikácii.

3.1.1 Manuálne vytvorenie štruktúry

Aplikácia umožní vytvoriť štruktúru manuálne, zadaním Zeichnungsnummer, Kunde a Bezeichnung.

3.1.2 Automatické vytvorenie štruktúry

Softvér umožní vytvoriť štruktúru automaticky, nahraním .xlsx (pozri 4.2) súboru do aplikácie, z ktorého sa prečíta Zeichnungsnummer, Kunde a Bezeichnung.

3.1.3 Obsah štruktúry

V štruktúre sa uchovávali nasledujúce položky:

verzia dielu - reťazec Zeichnungsnummer spojený s názvom zákazníka, pričom trojčíslenie za prvou bodkou odlišuje jednotlivé verzie toho istého dielu.

Ku každej verzii sa ukladajú nasledujúce položky, kde každú z nich tvorí 0..N ľubovoľných súborov:

dokumentácia - (typicky PDF, DOCX),

fotodokumentácia - (typicky PNG, JPG),

iterácia cylinder

program

signály:

originálny

editovaný

signál

Položka *signály* je rovnocenná položka, hoci obsahuje tri tiež rovnocenné podpoložky.

3.1.4 Názov štruktúry

Názov štruktúry vzniká prečítaním Zeichnungsnummer a Kunde z .xlsx (pozri 4.2) súboru alebo ich ručným zadáním do aplikácie. Názov potom vyzerá ako:

xxx.xxx.xxx.xxxNazovZakaznika

kde xxx.xxx.xxx.xxx je číslovanie dielu a nazovZakaznika je názov zadávateľa resp. zákazníka. Prvé šesťčísle sa vytvára podľa platného číslovania dielov (pozri 4.1), druhé šesťčísle sa automaticky generuje v softvéri SAP. Treba však zohľadniť aj starý formát číslovania dielov, ktorý má inú štruktúru názvu (pozri 3.1.5) ako nové diely. Názov štruktúry je zároveň názov dielu.

3.1.5 Starý názov štruktúry

Starý názov štruktúry (dielov) vznikal podľa iných pravidiel ako názov štruktúry v 3.1.4. Štruktúra však ostáva nezmenená, bude sa vytvárať ručne, tak ako je popísané v 3.1.1. Starý názov je postupnosť čísel a písmen oddelených pomlčkou. Výskyt starých názvov je zriedkavý.

3.2 Nastavenie dátumu testovania

Keď používateľ vytvára novú štruktúru (manuálne alebo automaticky), nastavuje dátum, kedy sa príslušná verzia daného dielu testovala / bude testovať. Predvolená hodnota je aktuálny dátum, keď sa štruktúra vytvára.

3.3 Nastavenie čísla zostavy

Pri vytváraní štruktúry sa bude manuálne nastavovať aj číslo zostavy (reťazec 1..? znakov), na ktorej sa daný diel testoval/bude testovať (číslo zostavy zodpovedá pracovisku/stroju, na ktorom sa súčiastka testuje).

3.4 Nahrávanie súborov do štruktúry

Používateľ môže kedykoľvek do vytvorenej štruktúry nahrávať súbory z jeho lokálneho disku. Aplikácia uloží kópiu každého takto nahratého súboru na serveri, kde bude prístupný aj pre ostatných používateľov aplikácie.

3.5 Vymazanie súborov

Používateľ môže kedykoľvek vymazať súbory zo štruktúry. Súbory sa vymažú z aplikácie a nie je možné ich obnoviť. Pred vymazaním sa aplikácia opýta užívateľa, či chce súbor naozaj vymazať. Až po potvrdení sa súbor vymaže.

3.6 Nahradenie súborov

Používateľ môže kedykoľvek nahradiť súbory v štruktúre. Nahradením sa rozumie nahratie súboru s rovnakým menom, aké má už nahratý súbor v aplikácii. Pred nahradením sa aplikácia opýta užívateľa, či chce súbor naozaj nahradiť. Až po potvrdení sa nový súbor nahrá do aplikácie, starý súbor sa vymaže a nie je možné ho obnoviť.

3.7 Vymazanie štruktúry verzie dielu

Používateľ môže kedykoľvek vymazať štruktúru verzie dielu (4.3 a2). Pred vymazaním sa aplikácia opýta užívateľa, či chce štruktúru naozaj vymazať. Až po potvrdení sa štruktúra vymaže. Vymazaním štruktúry verzie dielu sa vymažú všetky podzložky (dokumentácia, fotodokumentácia,...) vrátane všetkých súborov, ktoré v nich boli uložené. Obnovenie súborov a štruktúry po vymazaní nie je možné.

3.8 Vymazanie štruktúry dielu

Používateľ môže kedykoľvek vymazať štruktúru dielu (4.3 a1). Pred vymazaním sa aplikácia opýta užívateľa, či chce štruktúru naozaj vymazať. Až po potvrdení sa štruktúra vymaže. Vymazaním štruktúry dielu sa vymažú všetky štruktúry verzie dielu a ich súbory. Obnovenie súborov a štruktúry po vymazaní nie je možné.

3.9 Export všetkých súborov jednej štruktúry

Aplikácia dokáže na požiadavku používateľa vyexportovať ZIP-súbor so všetkými súbormi prislúchajúcimi k zvolenej štruktúre rozdelené do súborových priečinkov podľa štruktúry.

3.10 GUI

3.10.1 Prihlasovanie do systému

Užívateľ sa pri zapnutí aplikácie bude musieť prihlásiť. Po zapnutí aplikácie sa zobrazí prihlasovací formulár - meno a heslo. Po vyplnení prebehne verifikácia prihlasovacích údajov a ak boli správne, tak sa zobrazí GUI. Ak nie, tak sa zobrazí chybová hláška, ktorá vyzve užívateľa na opätovné zadanie prihlasovacích údajov.

3.10.2 Zobrazenia

Aplikácia bude mať (po úspešnom prihlásení) dve zobrazenia: úvodné zobrazenie a zobrazenie na pridanie štruktúr.

3.10.2.1 Úvodné zobrazenie

V úvodnom zobrazení budú zobrazené nasledujúce položky:

1. nastavenie vyhľadávania (filter) a tlačidlo na spustenie vyhľadávania (pozri 3.10.2.2)

2. zobrazenie štruktúr a súborov (pozri 3.10.2.3)
3. riadiaci prvok (napr. tlačidlo) na export záznamu zmien údajov uchovávaných v aplikácii (pozri 3.10)
4. riadiaci prvok na vytvorenie štruktúry (pozri 3.1)

3.10.2.2 Vyhľadávanie

Filtrovacími kritériami sú:

- (a) interval dátumu testovania (od - do)
- (b) zákazník
- (c) číslo dielu
- (d) umiestnenie dielu (napr. predná náprava)
- (e) číslo zostavy (na ktorej zostave sa testoval daný diel)
- (f) podľa čísla objednávky

Používateľ vyplní jeden alebo viacero filtrov, v tom prípade sa aplikujú súčasne (platia zároveň).

3.10.2.3 Zobrazenie štruktúr a súborov

Používateľovi sa zobrazia štruktúry usporiadané v stromovom zobrazení. Pred aplikáciou filtra sa zobrazia všetky štruktúry, po aplikácii filtra sa zobrazia iba štruktúry spĺňajúce kritériá vyhľadávania. Používateľ môže v tomto zobrazení prezerať, pridávať, nahradzovať, alebo odstraňovať súbory v jednotlivých štruktúrach, prípadne zmazať samotnú štruktúru. Počas zobrazenia štruktúry sa v GUI nachádza riadiaci prvok na export všetkých súborov štruktúry (pozri 3.X).

3.10.2.4 Zobrazenie na pridanie štruktúr


Po kliknutí na riadiaci prvok na vytvorenie štruktúry sa používateľovi zobrazí nové zobrazenie, kde bude možnosť nahráť súbor .xlsx na automatické vytvorenie štruktúry, alebo používateľ vytvorí štruktúru manuálne, zadанím príslušných informácií (pozri 3.1.1) do formulára.


3.11 História úprav

V histórii úprav (logu) sa budú nachádzať záznamy všetkých úprav (nahrávanie súborov, mazanie súborov,...), ktoré boli vykonané. V logu sa bude vyskytovať meno užívateľa (teda meno, pod ktorým sa prihlásil do aplikácie a presný čas).

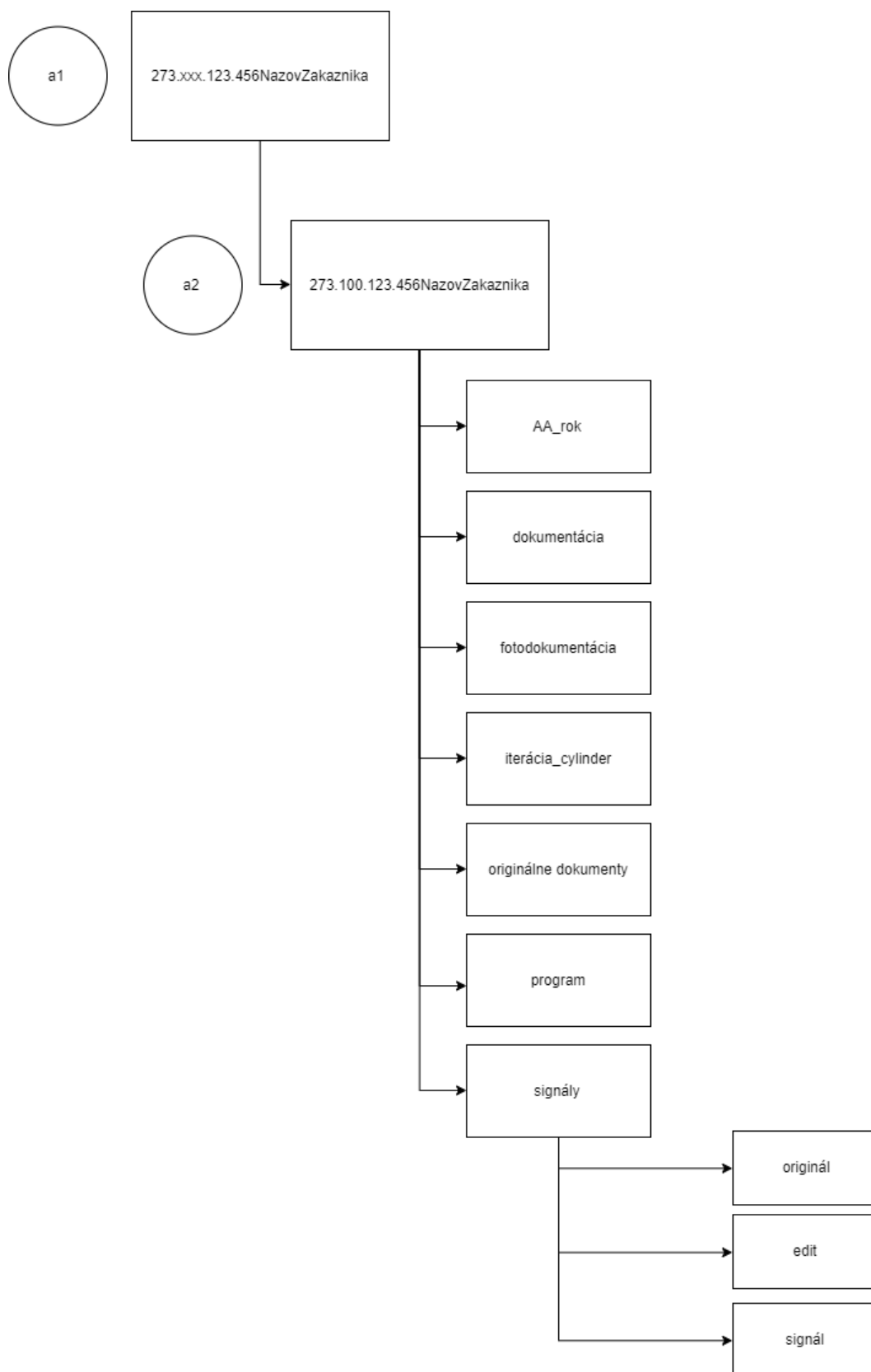
3.11.1 Zobrazenie histórie úprav

V úvodnom zobrazení môže používateľ zvoliť riadiaci prvok log a tým stiahnuť históriu úprav (log) do jeho počítača. Súbor sa po stiahnutí automaticky otvorí predvoleným programom na zobrazenie súborov typu .txt.

	A	B	C	D	E	F	G	H	I	J	K
1	Nr.	477015	Index	11	Arbeitsauftrag				 BOGE RUBBER & PLASTICS		
2	Kunde:			Bezeichnung			Zeichnungsnummer:		E-Stufe:	Datum:	
3	JLR			BSHG FRT SUSP LWR ARM MTNG BLT			273.300.012.409		0	4.9.2019	
4	Kunden-Zeichnungsnr.:			Konstruktionsstufe Kunde:		Zielfahrzeug:		Kennzeichnung		Auftraggeber:	
5	M8A2-3069-AAPROTO			D7a		X761 / L560 21MY		VB		Pleva	
6	Verrechnungsnummer				Bestelldaten:			Termin:	Σ Menge	Stk.-Kunde:	Stk.-Versuch:
7	CH-060675				6713519 / 2498320			KW37	2	0	2
8	Freigabe:										
9	Zur Vulkanisation			Ja		Nein					
10								Unterschrift			
11	Zum Versand			Ja		Nein					
12								Unterschrift			

	A	B	C	D	E	F	G	H	I	J	K
1	Nr.	480659	Index	0	Arbeitsauftrag				 BOGE RUBBER & PLASTICS		
2	Kunde:			Bezeichnung			Zeichnungsnummer:		E-Stufe:	Datum:	
3	Aston Martin			Single Hydro			735.910.155.201_C		1	5.9.2019	
4	Kunden-Zeichnungsnr.:			Konstruktionsstufe Kunde:		Zielfahrzeug:		Kennzeichnung		Auftraggeber:	
5	MY83-3069-540-16Hz			2PT		AM800		OT + proto assy		Zverbik	
6	Verrechnungsnummer				Bestelldaten:			Termin:	Σ Menge	Stk.-Kunde:	Stk.-Versuch:
7	CH-060020.01.02.TR							KW42	4	0	4
8	Freigabe:										
9	Zur Vulkanisation			Ja		Nein					
10								Unterschrift			
11	Zum Versand			Ja		Nein					
12								Unterschrift			

4.3 Štruktúra



Návrh systému

Filip Eliaš, Tomáš Kosec, Maroš Malý

Úvod

Účel dokumentu

Tento dokument slúži na detailný popis návrhu pre systém Databázy testov životnosti. Pomocou UML diagramov a popisov jednotlivých častí tried je v dokumente vysvetlené, akým spôsobom bude systém fungovať.

Zameranie a rozsah návrhu

Pred prečítaním je potrebné si prečítať katalóg požiadaviek, aby bol čitateľ detailne oboznámený s funkcionalitou systému.

Používané technológie

Java

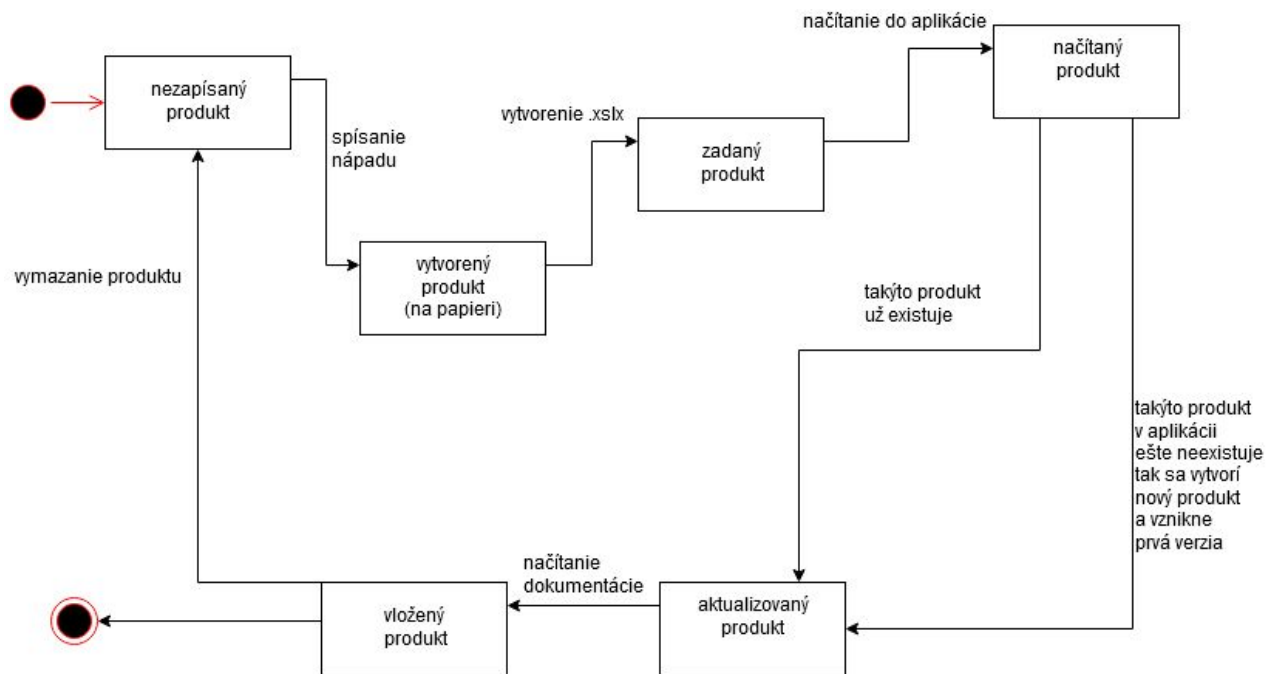
V projekte sa využíva programovací jazyk Java vo verzii 1.8. Pomocou nej sa vytvorí aplikácia a taktiež sa bude využívať na komunikáciu medzi aplikáciou a databázou.

MariaDB

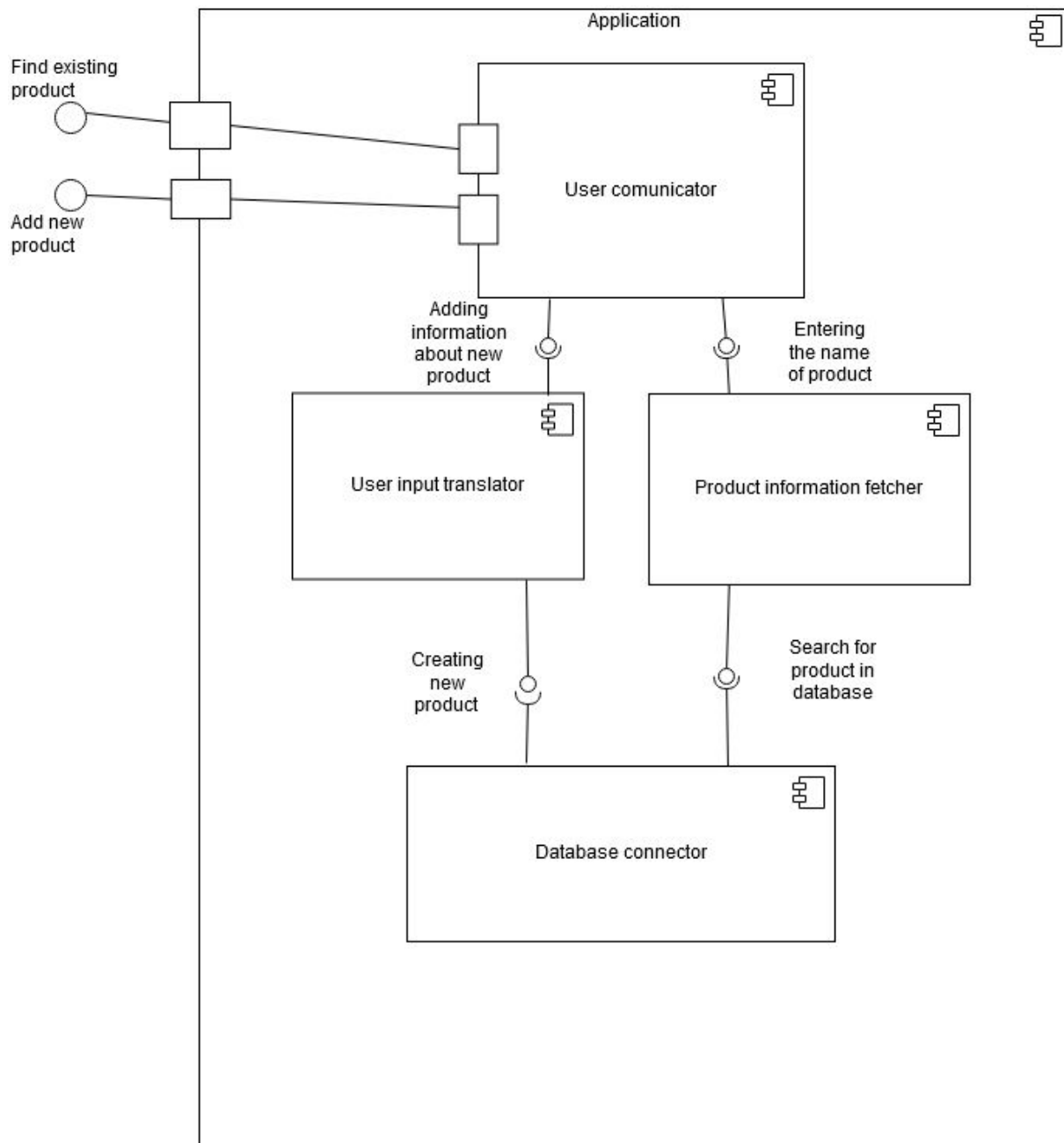
Na vytvorenie databázy sa používa MariaDB.

UML diagramy

UML state diagram



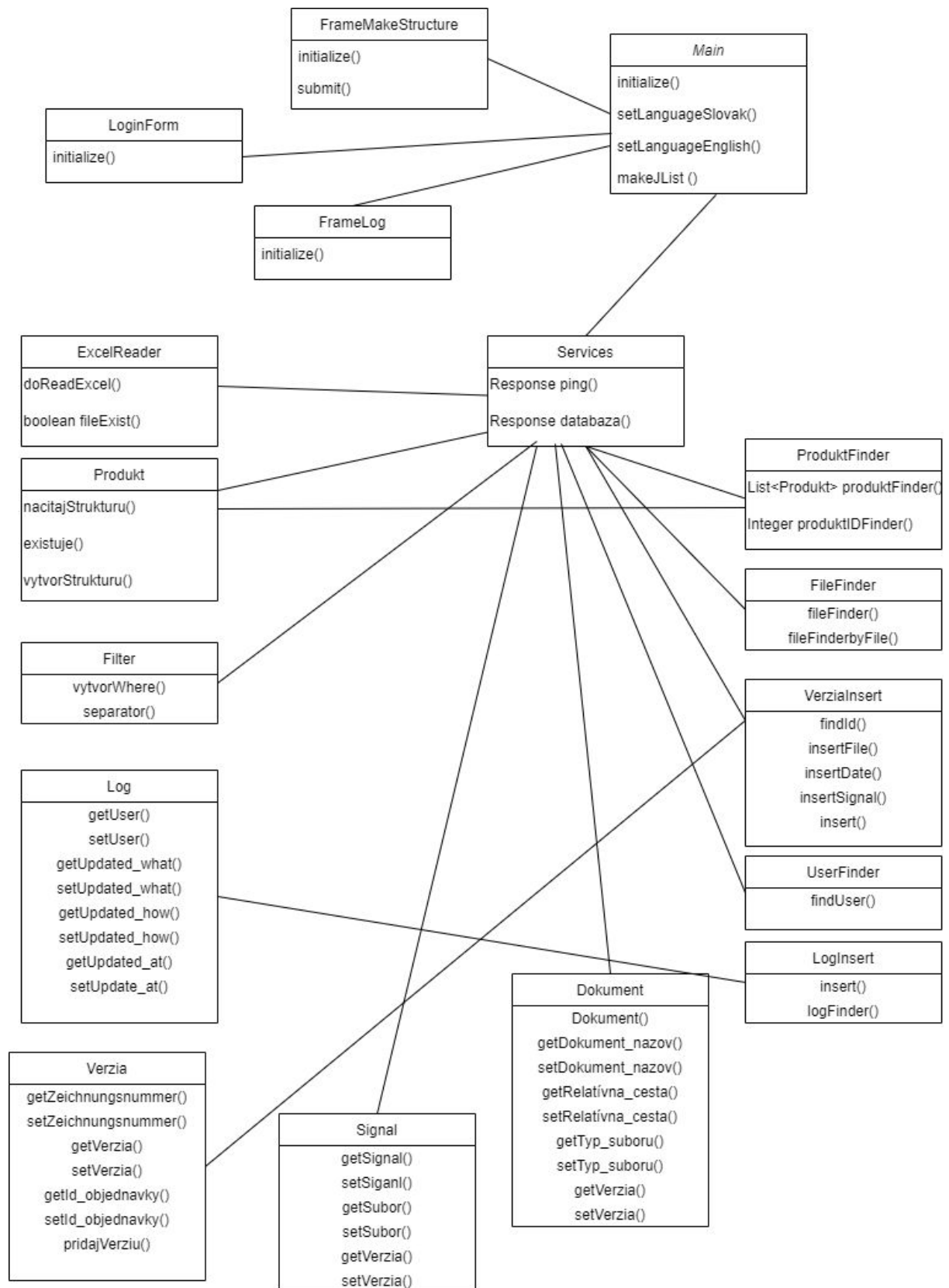
UML component diagram



Tento diagram zobrazuje 2 rôzne situácie, ktoré môžu v aplikácii nastať a Komponenty, ktoré ovplyvňujú.

Situácie, ktoré popisuje sú hľadanie existujúceho produktu a pridávanie nového produktu. Ak hľadáme existujúci product tak si v User Communicatore zadávame meno tohto produktu do Product information fetcher. Ten už komunikuje s Database connectorom a ten hľadá v database. Ak pridávame nový product, tak jeho informácie zadávame do User input translator. Ten do Database connectora pošle podnet na vytvorenie nového produktu.

UML class diagram



Návrh implementácie

ExcelReader

readExcel (String path)

Táto metóda sa volá pre zapnutie parsera názvu.

fileExist()

Metóda zisťuje, či zadaný súbor existuje.

doReadExcel()

Metóda prečíta dané políčka z priloženého .xlsx súboru a uloží ich do triednych premenných

getKunde()

Metóda vráti značku resp. meno zákazníka (Kunde).

getBezeichnung()

Metóda vráti miesto, kde sa nachádza daný diel.

getZeichnungsnummer()

Metóda vráti označenie/číslo dielu.

getNr()

Metóda vráti číslo objednávky.

getIndex()

Metóda vráti index objednávky.s

LogInsert

Integer insert(String user, String what, String how)

Metóda vloží do tabuľky **log** riadok obsahujúci informácie o vykonanej zmene.

List<Log> logFinder()

Metóda vráti zoznam všetkých riadkov z tabuľky **log** (čiže všetky vykonané zmeny v databáze)

ProduktFinder

List<Produkt> produktFinder(String kunde, String zeichnungsnummer, String bezeichnung, String nr)

Metóda vráti konkrétny produkt z tabuľky **produkt**, ktorý má dané parametre.

Integer produktIDFinder(String kunde, String zeichnungsnummer, String bezeichnung, String nr)

Metóda vráti ID daného produktu.

List<Produkt> produktAllFinder(String where)

Metóda vráti všetky produkty z tabuľky **produkt**, ktoré spĺňajú danú podmienku.

UserFinder

Boolean findUser(String user, String pass)

Metóda vráti true/false v prípade či sa daný user nachádza v tabuľke **uzivatelia**.

VerzialInsert

Integer insert(String zeichnungsnummer, String verzia, Integer id_objednavky)

Metóda vloží danú verziu do tabuľky **verzia**.

Integer findId(String zeichnungsnummer)

Metóda vráti id verzie s daným *zeichnungsnummer*.

insertFile(String nazov,String relnazov, String typ, String cesta, String user, String verzia)

Metóda vloží súbor do tabuľky **subory_k_verzii** pre danú verziu.

insertDate(String nazov, Date datum, Integer id)

Metóda nastaví dátum v tabuľke **verzie** pre danú verziu.

insertSignal(String file, String signal, String verzia, String user)

Metóda vloží súbor do tabuľky **signaly** pre danú verziu.

Filter

List<HashMap<String, String>> vytvorWhere()

Metóda vytvorí *where* podmienku na základe vyplnených údajov a následne zavolá *produktAllFinder* pre danú podmienku.

separator(String text, String varName)

Metóda vráti upravený text pri multi-množinovom vstupe atribútov (napr. keď používateľ zadá viac značiek, ktoré chce vyhľadať)

Produkt

nacitajStrukturu()

Ak daná štruktúra existuje metóda pridá novú verziu, ak neexistuje zavolá vytvorStrukturu().

boolean existuje()

Metóda zistí či daný produkt už existuje.

vytvorStrukturu()

Metóda vytvorí novú štruktúru.

String odstran_verziu_z_zeichnungsnummer(String zeichnungsnummer)

Metóda odstráni verziu z daného reťazca. Výstupom teda je reťazec v tvare
XXX.____.XXX.XXX

insert(String kunde, String zeichnungsnummer, String bezeichnung,
String nr, String zostava)

Metóda vloží produkt do tabuľky **produkt**.

update()

Metóda nastaví v tabuľke **produkt** stĺpec zostava.

Verzia

pridajVerziu()

Metóda vytvorí novú verziu a zároveň vytvorí na disku štruktúru priečinkov.

Services

Táto trieda obsahuje všetky service, ktoré sa používajú pri prepojení front endu a back endu. Jednotlivé metódy využívajú back endové triedy a ich metódy.

Response verzie(@Context javax.ws.rs.core.HttpHeaders hh)

Metóda vráti JSON obsahujúcu všetky verzie daného produktu.

Response allprodukt(@Context javax.ws.rs.core.HttpHeaders hh)

Metóda vráti JSON obsahujúcu všetky produkty.

Response newProdukt(final MultivaluedMap<String, String> params)

Metóda vytvorí nový produkt.

Response date(final MultivaluedMap<String, String> params)

Metóda nastaví dátum testovania pre danú verziu.

Response zostava(final MultivaluedMap<String, String> params)

Metóda nastaví zostavu pre daný produkt.

Response user(@Context javax.ws.rs.core.HttpHeaders hh)

Metóda zistí či sa daný užívateľ nachádza v databáze.

Response log()

Metóda vráti všetky záznamy z tabuľky **log**.

Functions.js

nacitaj()

Metóda vyberie údaje z inputboxov a pošle ich na server, kde sa údaje spracujú do SQL WHERE podmienky, podľa nej sa vráti JSON vybraných produktov.

log()

Metóda si vypýta všetky údaje z log tabuľky, dostane naspäť JSON.

login()

Metóda kontroluje zadané údaje od používateľa, kontroluje či je meno a heslo správne.

newProduct()

Metóda vyberie z inputboxov údaje o produkte, ktoré odošle na server, ktorý následne vytvorí nový produkt.

fillData()

Metóda načíta vstupný xlsx súbor a načítané hodnoty vloží do príslušných inputboxov.

download()

Metóda stiahne informáciu o úpravách používateľov (log) v podobe textového dokumentu.

setDatum()

Metóda nastaví dátum testovania pre vybranú verziu.

setZostava()

Metóda nastaví verziu testovania pre vybraný produkt.

showSetters()

Metóda zobrazí element na nastavenie dátumu a verzie.

hideSetters()

Metóda skryje element na nastavenie dátumu a verzie. Podľa vyplnených polí pre verziu a dátum zavolá funkcie na nastavenie dátumu a verzie.

`setActualRow()`

Metóda nastaví globálnu premennú na hodnotu aktuálneho označeného riadka

`vypisLog(data)`

Metóda vypíše log.

Test otvorenie aplikácie:

1. Používateľ klikne na ikonu programu, prípadne stlačí enter na označenej ikone
2. Spustí sa aplikácia

Test prihlásenia:

1. Používateľ zadá prihlasovacie meno do text boxu
2. Používateľ zadá heslo
3. Používateľ klikne na tlačidlo "Login"
4. Používateľovi sa zobrazí hlavná obrazovka

Test odhlásenia:

1. Používateľ klikne na tlačidlo "Log out"
2. Používateľ sa odhlási z aplikácie
3. Používateľovi sa zobrazí Login formulár

Test filtra:

1. Používateľ zadá do jednotlivých text boxov atribúty vyhľadávania (1-7)
2. Používateľ klikne na tlačidlo "Filter"
3. Používateľovi sa zobrazia výsledky vyhľadávania

Test vypísania všetkých súborov:

1. Používateľ nevyplní žiaden text box (nevyplní žiaden atribút vyhľadávania)
2. Používateľ klikne na tlačidlo "Filter"
3. Používateľovi sa zobrazia všetky uložené súbory

Test nahrávania dielov:

1. Používateľ klikne na tlačidlo "Vytvor Štruktúru" ("Make Structure")
2. Používateľovi sa zobrazí okno kde môže nahrať excel, prípadne manuálne vyplniť požadované atribúty
3. Vytvorí sa štruktúra

Test nahrávania súborov do štruktúry:

1. Používateľ si v štruktúre nájde daný diel
2. Používateľ klikne na meno daného dielu
3. Používateľ klikne na tlačidlo nahratia súboru "Insert"
4. Používateľ zvolí súbor na nahranie
5. Súbor sa nahrá do štruktúry

Test mazania súborov zo štruktúry:

1. Používateľ si v štruktúre nájde súbor, ktorý chce vymazať
2. Používateľ klikne na meno daného súboru
3. Používateľ klikne na tlačidlo "Delete"

4. Súbor sa vymaže zo štruktúry

Test nahradenia súboru:

1. Používateľ si v štruktúre nájde daný diel
2. Používateľ klikne na meno daného dielu
3. Používateľ klikne na tlačidlo nahratia súboru "Update"
4. Používateľ zvolí súbor na nahranie
5. Používateľovi sa zobrazí okno či si je istý, že chce vybraný súbor nahradiť
6. Ak používateľ zvolí možnosť, že si nepraje nahradiť, nenastane žiadna zmena
7. V opačnom prípade sa súbor nahradí v štruktúre

Test vymazanie štruktúry dielu:

1. Používateľ si v štruktúre nájde štruktúru dielu, ktorú chce vymazať
2. Používateľ klikne na meno danej štruktúry
3. Používateľ klikne na tlačidlo "Delete"
4. Používateľovi sa zobrazí okno či si je istý, že chce vybranú štruktúru vymazať
5. Ak používateľ zvolí možnosť, že si nepraje štruktúru vymazať nenastane žiadna zmena
6. V opačnom prípade sa štruktúra nenávratne vymaže

Test vymazania štruktúry verzie dielu:

1. Používateľ si v štruktúre nájde štruktúru verzie dielu, ktorú chce vymazať
2. Používateľ klikne na meno danej štruktúry verzie dielu
3. Používateľ klikne na tlačidlo "Delete"
4. Používateľovi sa zobrazí okno či si je istý, že chce vybranú štruktúru verzie dielu vymazať
5. Ak používateľ zvolí možnosť, že si nepraje štruktúru verzie dielu vymazať nenastane žiadna zmena
6. V opačnom prípade sa štruktúra verzie dielu vymaže zo štruktúry

Test exportu všetkých súborov jednej štruktúry:

1. Používateľ si v štruktúre nájde štruktúru verzie dielu, ktorú chce exportovať
2. Používateľ klikne na meno danej štruktúry verzie dielu
3. Používateľ klikne na tlačidlo "Export"
4. Používateľovi sa zobrazí okno pre výber miesta uloženia súborv štruktúry na lokálnom disku
5. Používateľ zvolí miesto uloženia
6. Súbory štruktúry sa uložia na používateľom vybrané miesto na disku v zip archíve

Test zmeny jazyka používateľského rozhrania:

1. Používateľ klikne na položku v menu lište "Jazyk" ("Language")
2. Používateľ zvolí preferovaný jayzk z dvojice "SLK" pre slovenský jazyk a "ENG" pre anglický jazyk
3. Jazyk používateľského rozhrania sa nastaví podľa výberu používateľa

Test zobrazenia histórie úprav:

1. Používateľ klikne na položku v menu lište "LOG"
2. Používateľovi sa zobrazí textové okno so záznamom histórie úprav
3. Používateľ klikne na tlačidlo "Export"
4. Používateľovi sa zobrazí okno pre výber miesta uloženia txt súboru na lokálnom disku
5. Používateľ zvolí miesto uloženia
6. Log sa uloží na disk vo forme textového súboru