

Technická dokumentácia

Evidencia verzií výrobkov

Projekt z predmetu Tvorba Informačných Systémov

vypracovali: *Lukáš Kostrian, Jakub Krištof, Peter Verčimák, Barbora Vicianová*

Obsah

1. Katalóg požiadaviek	4
1.1 Úvod	5
1.1.1 Účel tohto katalógu požiadaviek	5
1.1.2 Rozsah využitia systému	5
1.1.3 Slovník pojmov	5
1.1.4 Odkazy a referencie	5
1.1.5 Prehľad nasledujúcich kapitol	6
1.2 Všeobecný popis	6
1.2.1 Perspektíva Systému	6
1.2.2 Funkcie systému	6
1.2.3 Charakteristika používateľov	6
1.2.4 Predpoklady, závislosti a obmedzenia	6
1.3 Špecifikácie požiadaviek	7
1.3.1 Nahrávanie PDF súboru	7
1.3.2 Vytiahnutie špecifických dát z PDF súboru	7
1.3.3 Zápis dát do databázy	7
1.3.3.1 Nahratie obrázku	7
1.3.3.2 Uloženie dát do databázy	7
1.3.4 Nahratie DVP tabuľky do aplikácie	7
1.3.5 Uloženie dát z DVP tabuľky do databázy	7
1.3.6 Zobrazenie histórie súčiastky	7
1.3.7 Zobrazenie údajov verzie súčiastky	7
1.3.8 Prostredie na vytvorenie Excel šablóny pre klienta	8
1.3.8.1 Nahratie Excel tabuľky dodanej klientom	8
1.3.8.2 Definovanie šablóny	8
1.3.8.3 Export testov podľa šablóny	8
1.3.9 Úprava záznamov v databáze	8
1.3.10 Zobrazenie záznamu úprav databázy	8
2. Návrh	9
2.1 Úvod do návrhu	10
2.2 Zameranie	10
2.3 Výber použitých technológií	10
2.4 Špecifikácia vonkajších interfejsov	10
2.5 Návrh používateľského rozhrania	10
2.6 Perzistentné údaje a dátový model	11

2.7 Moduly	12
2.8 UML Diagramy	12
2.8.1 Component diagram	12
2.8.2 Class diagram	13
2.8.3 Sequence diagram	13
2.9 Plán implementácie	13
2.10 Testovacie scenáre	14
2.10.1 Upload PDF + screenshot CATIA súčiastky	14
2.10.2 Upload DVP tabuľky	14
2.10.3 Vyhľadávanie a prezeranie záznamov	14
2.10.4 Prezeranie DVP výsledkov	14
2.10.5 Zmena údajov v záznamoch	14
2.10.6 Export DVP tabuľky do šablóny	15
2.10.7 Vytvorenie šablóny pre export DVP tabuliek	15
2.10.8 Zobrazenie záznamov o úprave databázy	15
2.10.9 Zmena používateľa	15

1. Katalóg požiadaviek

1.1 Úvod

1.1.1 Účel tohto katalógu požiadaviek

Tento dokument slúži na špecifikáciu všetkých požiadaviek pre informačný systém slúžiaci na evidenciu verzií výrobkov. Tento systém vznikol ako projekt v rámci predmetu Tvorba Informačných Systémov na Fakulte matematiky, fyziky a informatiky UK v Bratislave. Dokument vznikol na základe požiadaviek od zadávateľa a slúži ako východiskový materiál pre následnú implementáciu systému, a je prístupný všetkým osobám, ktoré sa zapoja do vývoja, budú ho obsluhovať alebo používať. Taktiež slúži ako záväzná dohoda medzi zadávateľom a riešiteľmi.

1.1.2 Rozsah využitia systému

Systém je vyvíjaný pre firmu Boge Elastmetal. Cieľom projektu je vyvinúť systém slúžiaci na evidenciu verzií výrobkov, ich testov, a ich vytiahnutie do tabuliek dodanými klientom. Firma má v súčasnosti dáta uložené vo viacerých Excel tabuľkách, a zápis nových verzií je časovo náročný a repetitívny. Vyvinutý systém vyrieši duplicitné zadávanie dát do viacerých tabuliek, a niekoľkonásobne zrýchli či už evidenciu, ale aj vytiahnutie údajov do formátov špecifikovaných klientami.

1.1.3 Slovník pojmov

<i>CATIA</i> -	3D softvér používaný firmou
<i>DVP tabuľka</i> -	tabuľka obsahujúca fyzikálne hodnoty získané z testov jednotlivých komponentov súčiastky, dodané externe z laboratória vo formáte Excel
<i>Súčiastka</i> -	“výrobok”, evidujeme jej verzie
<i>Podsúčiastka</i> -	súčasť súčiastky, no taktiež má svoju históriu záznamov, v podstate súčiastka = podsúčiastka

1.1.4 Odkazy a referencie

Odkaz na GitHub repozitár - <https://github.com/TIS2022-FMFI/form-versions>
CATIA dokumentácia - http://maruf.ca/files/catiahelp/CATIA_P3_default.htm

1.1.5 Prehľad nasledujúcich kapitol

V nasledujúcich kapitolách bude všeobecne popísaný vyvíjaný systém, jeho funkcionality, charakteristika používateľov a taktiež samotné špecifické požiadavky.

1.2 Všeobecný popis

1.2.1 Perspektíva Systému

Systém bude slúžiť na zjednotenie ukladania dát verzií výrobkov, nahradí aktuálne ukladanie dát do viacerých tabuliek v programe Excel a zjednoduší prácu s evidenciou. Zachová aktuálnu štruktúru dát, no eliminuje potrebu manuálneho zadávania dát.

1.2.2 Funkcie systému

Výsledným systémom bude Windows aplikácia, napojená na spoločnú databázu, uloženú na lokálnom serveri firmy. Pri prvom spustení zamestnanec zadá svoje meno/ID, pod ktorým budú uložené jeho zmeny. Pri upravení 3D modelu výrobku zamestnanec nahradí PDF export z CATIA súbor do aplikácie, a tá automaticky vytiahne potrebné dáta a uloží ich do databázy. V aplikácii sa následne bude dať vyhľadať špecifická súčiastka, zobrazíť jej "podsúčiastky", históriu zmien a odkaz na výsledky testov. Výsledky testov bude možné nahráť vo formáte Excel tabuľky, a údaje sa zapíšu do databázy. Testy bude možné exportovať do formátov špecifikovaných klientami. Firma si sama v rozhraní aplikácie nahradí šablónu od klienta, definuje políčka, kde sa majú uložiť dáta z DVP tabuľky, a súbor aj so šablónou sa uloží do spoločnej databázy a neskôr budú použiteľné ihneď. Všetky záznamy bude možné upravovať aj po nahrať. Zoznam všetkých úprav bude dostupný, a bude možné vidieť kto, kedy spravil akú zmenu.

1.2.3 Charakteristika používateľov

Zamestnanec - má možnosť nahráť PDF súbor, nahráť súbor s testami, vytvoriť šablónu pre export, vytvoriť export pre firmu a taktiež bude vedieť zobrazíť záznam zmien v databáze

Správca - nasadí databázu na lokálny server a rozdistribuje systém do počítačov zamestnancov

1.2.4 Predpoklady, závislosti a obmedzenia

Systém bude realizovaný ako Windows aplikácia s databázou riešenou pomocou MariaDB, zdieľanou pre všetkých zamestnancov, uloženú na lokálnom serveri firmy.

1.3 Špecifikácie požiadaviek

1.3.1 Nahrávanie PDF súboru

- Používateľ bude schopný do aplikácie nahráť PDF súbor

1.3.2 Vytiahnutie špecifických dát z PDF súboru

- Vložený PDF súbor bude aplikáciou spracovaný, a "vytiahnu" sa potrebné dáta

1.3.3 Zápis dát do databázy

1.3.3.1 Nahratie obrázku

- Používateľ bude vedieť nahráť obrázok modelu z programu CATIA, ktorý bude evidovaný spolu s danou verziou

1.3.3.2 Uloženie dát do databázy

- Po spracovaní PDF súboru sa dáta uložia do databázy

1.3.4 Nahratie DVP tabuľky do aplikácie

- Používateľ bude schopný do aplikácie nahráť DVP tabuľku

1.3.5 Uloženie dát z DVP tabuľky do databázy

- Aplikácia získa dáta z DVP tabuľky o vykonaných testoch, a podľa ID súčiastky ich priradí a pridá k zodpovedajúcemu záznamu v databáze.

1.3.6 Zobrazenie histórie súčiastky

- Používateľ bude vedieť vyhľadať súčiastku podľa ID, a zobraziť jej históriu verzií, jej obrázok a dátum

1.3.7 Zobrazenie údajov verzie súčiastky

- Používateľ bude vedieť zvoliť verziu súčiastky, a zobraziť, aké zmeny boli spravené

1.3.8 Prostredie na vytvorenie Excel šablóny pre klienta

1.3.8.1 Nahratie Excel tabuľky dodanej klientom

- Používateľ bude vedieť nahráť súbor s vopred preddefinovanou šablónou od klienta vo formáte Excel

1.3.8.2 Definovanie šablóny

- Používateľ bude vedieť definovať špecifické políčka v Excel tabuľke z bodu 3.8.1 (podľa súradníc), do ktorých sa zapíšu vybrané ľubovoľné údaje z DVP tabuľky, prípadne názov, ID a dátum

1.3.8.3 Export testov podľa šablóny

- Používateľ bude vedieť exportovať Excel tabuľku s testami danej súčasti, a bude si vedieť zvoliť, podľa ktorej šablóny

1.3.9 Úprava záznamov v databáze

- Používateľ bude vedieť dodatočne zmeniť všetky polia ľubovoľného zvoleného záznamu v databáze

1.3.10 Zobrazenie záznamu úprav databázy

- Používateľ bude vedieť zobraziť všetky zmeny vykonané v databáze, a to presne dátum, čas, čo sa zmenilo/nahrало a kto zmenu vykonal

2. Návrh

2.1 Úvod do návrhu

Dokument slúži ako detailný návrh informačného systému pre firmu BOGE na evidovanie verzií súčiastok. Obsahuje dôležité informácie potrebné na pochopenie funkcionality aj jej vysvetlenie a spôsob implementácie systému. Dokument je primárne určený pre vývojárov. Tento dokument obsahuje všetky požiadavky z Katalógu požiadaviek

2.2 Zameranie

Na prácu s dokumentom je potrebné byť oboznámený s Katalógom požiadaviek pre jednoduchšie pochopenie. Návrh spracováva požiadavky Katalógu a obsahuje kompletný a detailný postup implementácie systému. V nasledujúcich kapitolách sa nachádza špecifikácia vonkajších interfejsov, dátové modely a návrh používateľského rozhrania vrátane vizuálnej časti.

2.3 Výber použitých technológií

Aplikácia bude nainštalovaná na všetky Windows počítače zamestnancov firmy BOGE. Aplikácia bude napísaná v jazyku Java, a bude komunikovať s MariaDB databázou na lokálnom serveri firmy.

2.4 Špecifikácia vonkajších interfejsov

Aplikácia bude spustená lokálne v zariadení a s používateľom bude komunikovať pomocou používateľského grafického rozhrania. Aplikácia bude komunikovať s MariaDB databázou umiestnenou na lokálnom serveri pomocou Java package mariadb a importu java.sql. Ďalej bude spracovávať dva druhy Excel súborov pomocou Apache POI, jeden Excel súbor bude exportovaný zo softvéru CATIA a druhý Excel súbor je v stanovenom formáte DVP. Taktiež pomocou Apache POI bude aplikácia exportovať Excel súbory.

Využitá technológia:

- Java - aplikácia bude napísaná v jazyku Java najmä kvôli veľkej dostupnosti knižníc ku danému projektu
- MariaDB - databáza na uchovanie údajov na lokálnom serveri
- MariaDB Connector/J - adaptér na komunikáciu s MariaDB databázou
- Apache POI - knižnica na spracovanie a úpravu Excel súborov
- JavaFX - knižnica na vytvorenie GUI
- IMGScalr - knižnica na úpravu veľkostí obrázkov
- JMetro - knižnica na CSS GUI

2.5 Návrh používateľského rozhrania

[odkaz](#)

2.6 Perzistentné údaje a dátový model

2.6.1 Popis jednotlivých tabuliek

- **bom**
 - int *id* **PK** = id v databáze
 - varchar(20) *parent* = id "rodiča" (zodpovedá *part.part_id*)
 - varchar(20) *child* = id "dieťaťa" (zodpovedá *part.part_id*)
- **coordinates**
 - int *id* **PK** = id
 - int *template_id* **FK** = id šablóny (zodpovedá *table.id*)
 - int *row* = riadok
 - int *col* = stĺpec
 - int *sheet* = sheet
 - int *test_type* **FK** = id typu testu (zodpovedá *test_types.id*)
- **db_log**
 - int *id* **PK** = id
 - varchar(40) *user_id* = meno užívateľa
 - timestamp *time* = čas úpravy
 - varchar(100) *value* = úprava
- **part**
 - int *id* **PK** = id v databáze
 - varchar(20) *part_id* = id súčiastky
 - varchar(20) *type* = typ súčiastky (model/prototyp/séria)
 - varchar(30) *date* = dátum vzniku/úpravy
 - varchar(250) *comment* = komentár k verzii
 - longblob *image* = obrázok
 - varchar(30) *developed_from* = id súčiastky z ktorej je vyrobená
 - varchar(30) *name* = názov súčiastky
- **template**
 - int *id* **PK** = id
 - varchar(100) *name* = názov šablóny definovaný užívateľom
 - longblob *excel* = prázdna Excel šablóna

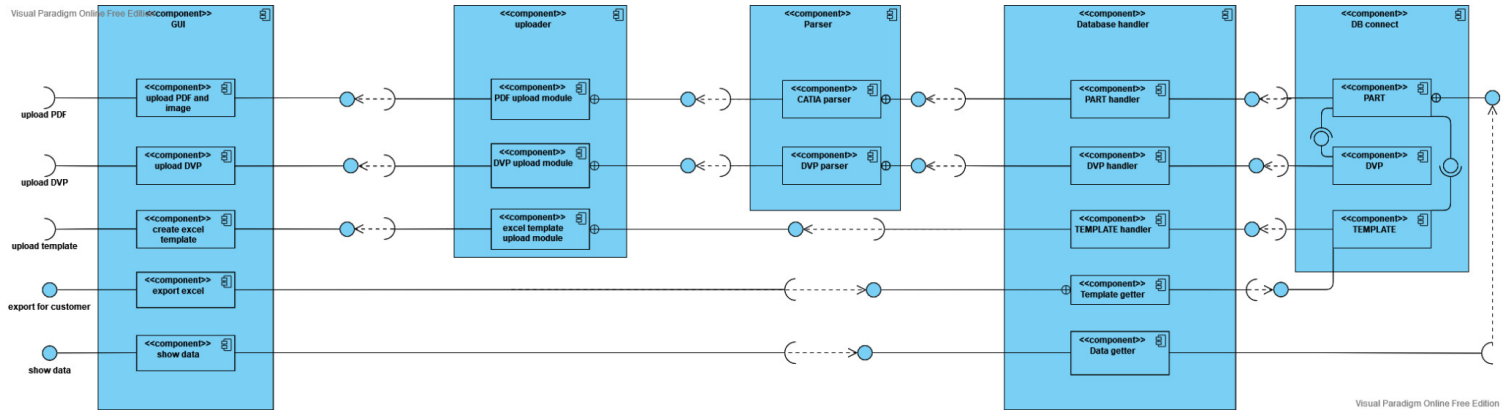
- **test**
 - int *id* **PK** = id
 - varchar(50) *date* = dátum
 - varchar(50) *customer_id* = id zákazníka
 - varchar(100) *aa* = aa
 - varchar(20) *part_id* = id súčiastky (zodpovedá *part.part_id*)
- **test_result**
 - int *id* **PK** = id
 - varchar(50) *est_type* = id testu (zodpovedá test_types.id)
 - varchar(50) *test_result* = výsledok testu
 - varchar(50) *test_soll* = soll testu
 - varchar(50) *test_soll_plus* = soll + testu
 - varchar(50) *test_soll_minus* = soll - testu
 - int *test_id* = id testu (zodpovedá *test.id*)
- **test_types**
 - int *id* **PK** = id
 - varchar(100) *name* = názov
- **users**
 - int *id* **PK** = id
 - varchar(50) *mail* = mail užívateľa
 - varchar(50) *psswd* = heslo užívateľa v MD5 šifrovaní
 - int *admin* = 0/1 bool či je user admin alebo nie

2.7 Moduly

- **Modul spracovania PDF súborov z CATIA** - pomocou Apache POI načítaný PDF súbor vo formáte z Catia nahrá do databázy pomocou package mariadb a java.sql
- **Modul spracovania Excel súborov DVP** - pomocou Apache POI načítaný Excel súbor vo formáte DVP tabuľky nahrá do databázy pomocou package mariadb a java.sql
- **Modul GUI** - pomocou java.swing vytvorené používateľské rozhranie
- **Modul zobrazenia údajov** - zobrazenie údajov z databázy pomocou package mariadb a java.sql
- **Modul vytvorenia exportovacieho Excel formátu** - pomocou Apache POI sa vytvorí požadovaný Excel formát a ten sa pomocou package mariadb a java.sql uloží v databáze
- **Modul exportovania Excelu** - pomocou Apache POI sa vytvorí a exportuje Excel vo formáte zadefinovaného v databáze

2.8 UML Diagramy

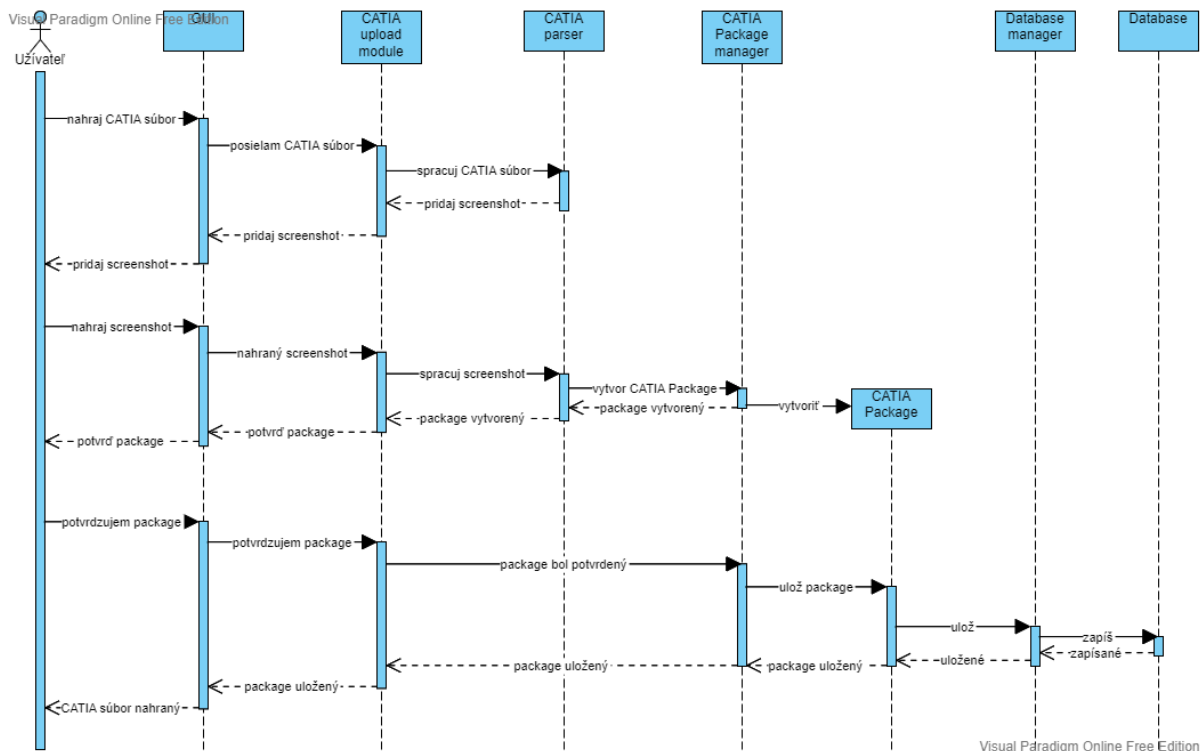
2.8.1 Component diagram



2.8.2 Class diagram

[odkaz](#)

2.8.3 Sequence diagram



2.9 Plán implementácie

- Prototyp verzia GUI pre realizáciu a testovanie funkčnosti kódu.
- Vytvorenie Create Scriptu databázy
- Spracovanie údajov z PDF súboru
- Pripravenie dát na vloženie do databázy
- Spracovanie dát z Excel tabuľky(DVP)
- Vloženie spracovaných dát z Catie a Excel tabuliek do databázy
- Zobrazovanie údajov
- Vyhľadávanie súčiastok
- Zobrazenie výsledkov testov
- Vytvorenie šablón na export dát z databázy
- Export dát z databázy do Excelu vo vopred vytvorenom formáte

2.10 Testovacie scenáre

2.10.1 Upload PDF + screenshot CATIA súčiastky

Scenár: Užívateľ chce pridať PDF súbor so súčiastkou do databázy. Zadá preto do aplikácie daný PDF súbor, priloží k nemu obrázok, a stlačí tlačidlo na upload.

Výstup: Po stlačení tlačidla sa súbor spracuje a s obrázkom sa vloží do databázy. Zmena sa zapíše do záznamu zmien. Ak užívateľ nepriložil PDF súbor alebo obrázok, užívateľovi sa zobrazí varovanie, že mu chýba časť.

2.10.2 Upload DVP tabuľky

Scenár: Užívateľ chce nahrať DVP tabuľku vo formáte Excel. Zadá do aplikácie daný Excel súbor a stlačí tlačidlo na upload.

Výstup: Po stlačení tlačidla sa všetky existujúce výsledky testov spracujú, skontrolujú či už sú v databáze a ak nie, nahrajú sa do databázy. Zmena sa zapíše do záznamu zmien.

2.10.3 Vyhľadávanie a prezeranie záznamov

Scenár: Užívateľ si chce prezrieť záznam o súčiastke, ktorej ID pozná. Do aplikácie teda zadá je ID, a klikne na tlačidlo vyhľadať.

Výstup: Po stlačení tlačidla sa mu zobrazia všetky verzie súčiastky, a následne ak na nejakú klikne, zobrazí sa jej komentár a jej zloženie.

2.10.4 Prezeranie DVP výsledkov

Scenár: Užívateľ si chce prezrieť výsledky testov danej súčiastky. Najprv postupuje podľa bodu 2.10.3., a následne zvolí tlačidlo “Zobraziť DVP výsledky”

Výstup: Po stlačení tlačidla sa mu zobrazia všetky testy vykonané na danej verzii súčiastky.

2.10.5 Zmena údajov v záznamoch

Scenár: Užívateľ chce zmeniť ľubovoľný záznam, najprv si ho vyhľadá podľa bodu 2.10.3, prípadne 2.10.4, a následne bude vedieť dvojklikom zvoliť údaj a prepísať ho.

Výstup: Po stlačení tlačidla sa všetky existujúce výsledky testov spracujú, skontrolujú či už sú v databáze a ak nie, upraví sa databáza. Zmena sa zapíše do záznamu zmien.

2.10.6 Export DVP tabuľky do šablóny

Scenár: Užívateľ si chce vyexportovať DVP testy pre danú verziu súčiastky. Najprv postupuje podľa bodu 2.10.3, a následne po zvolení danej verzie stlačí tlačidlo “Export DVP”

Výstup: Po stlačení tlačidla užívateľ zvolí daný dátum testu (ak je pre danú súčiastku viac testov), a následne sa mu zobrazí okno, kam zadá lokáciu uloženia, a ak zvolí tlačidlo OK, vygeneruje sa mu nový Excel súbor s dátami z databázy. Zmena sa zapíše do záznamu zmien.

2.10.7 Vytvorenie šablóny pre export DVP tabuliek

Scenár: Užívateľ zadá prázdny Excel súbor, následne pre každý požadovaný údaj zadá suradnice na ktorý sa má v danom Excel súbore zapísať, a stlačí tlačidlo "OK"

Výstup: Po stlačení tlačidla sa prázdny Excel súbor uloží v databáze, spolu so všetkými súradnicami.

2.10.8 Zobrazenie záznamov o úprave databázy

Scenár: Užívateľ zvolí kategóriu "História"

Výstup: Užívateľovi sa následne zobrazí tabuľka všetkých úprav databázy, spolu s časom úpravy a jej autorom.