

# **Katalóg požiadaviek**

## **Audit prepráv**

**Tvorba informačných systémov**  
**2022/2023**

**Roman Božik**

**Jozef Kolek**

**Terézia Strišovská**

**Michal Uhrin**

## Obsah

<b>1. Úvod</b>	<b>3</b>
1.1. Účel katalógu požiadaviek	3
1.2. Rozsah využitia systému	3
1.3. Slovník pojmov	3
1.4. Odkazy a referencie	4
1.5. Prehľad nasledujúcich kapitol	4
<b>2. Všeobecný popis</b>	<b>5</b>
2.1. Perspektíva systému	5
2.2. Funkcie systému	5
2.3. Charakteristika používateľov	6
2.4. Všeobecné obmedzenia	7
2.5. Predpoklady a závislosti	7
<b>3. Špecifikácie a požiadavky</b>	<b>7</b>
3.1. Funkčné požiadavky	7
3.1.1. Import dát	7
3.1.2. Pracovníci	8
3.1.3. Audit vozidla	8
3.1.4. Riešenie chýb	9
3.1.5. Rozšírené požiadavky (voliteľné)	10
3.2. Kvalitatívne požiadavky	10

## 1. Úvod

### 1.1. Účel katalógu požiadaviek

Tento dokument predstavuje súhrn všetkých požiadaviek pre projekt Audit prepráv, ktorý vznikol ako projekt v rámci predmetu Tvorba informačných systémov na Fakulte matematiky, fyziky a informatiky UK v Bratislave. Dokument je určený všetkým osobám zapojeným do vývoja systému, jeho správy a užívania. Slúži ako záväzná dohoda medzi vývojármi a zadávateľom projektu.

### 1.2. Rozsah využitia systému

Cieľom projektu je aplikácia na kontrolu vozíkov v sklade, ktorý patrí Gefco Slovakia s.r.o. . Aplikácia bude kontrolovať poradie vozíkov, prvý a posledný produkt umiestnený na ňom. Na kontrolu sa použijú dáta exportované zo systému, alebo z papierov ktoré sú prikladané k vozíkom.

### 1.3. Slovník pojmov

TLS – opakujúca sa sekvencia čísel v rozsahu 0-9999

IONO – osemmiestne číslo ktoré je unikátne pre každé jedno vozidlo

Report - vstupný súbor obsahujúci informácie o tovaroch v .CSV formáte. Mal by obsahovať carriagekey, TLS, vehicle\_id, position.

Export - výstupný súbor obsahujúci údaje o kontrole tovaru naloženého do jedného vozidla

Zákazník - zákazník zadávateľa

Užívateľ - osoba poverená na obsluhu systému na základnej úrovni

Administrátor - osoba poverená na spravovanie systému

Kontrolór - osoba oprávnená na vykonanie manuálneho auditu prepráv

Pattern - určuje koľko vozíkov konkrétného typu sa zmestí do daného vozidla, môže obsahovať viac rôznych typov

Config - každý zákazník má v databáze uložené vozidlo, do ktorého sa zvyčajne nakladá jeho zásielka a tiež zoznam pracovníkov, ktorí môžu počas auditu tohto zákazníka znovu otvoriť vozidlo.

## 1.4. Odkazy a referencie

### 1.4.1. vzorová ukážka reportu (viď v slovníku pojmov)

[Report](#)

[Popis výstupu](#)

[Etiketa na vozíku](#)

## 1.5. Prehľad nasledujúcich kapitol

V druhej kapitole je popísaný všeobecný popis informačného systému, ktorý zahŕňa perspektívu a funkcionality systému.

Tretia kapitola sa venuje špecifikácii požiadaviek, konkrétne funkčným, kvalitatívnym a používateľským.

## 2. Všeobecný popis

### 2.1. Perspektíva systému

Produkt bude mobilná (androidová) aplikácia na kontrolu vozíkov pri nakladaní vozidiel. Cieľom aplikácie bude vedieť pomocou skenovania kontrolovať položky v aute na základe reportov. Údaje o kontrole naloženého vozidla sú uložené do databázy a budú môcť byť exportované vo formáte PDF . Aplikácia bude čítať zoznam pracovníkov, vozidiel, zákazníkov spolu s ich patternami z databázy.

Exportované súbory budú dostupné v nato určenej webovej aplikácii. Vo webovej aplikácii bude taktiež možné pridávať, odstraňovať pracovníkov a meniť ich údaje.

### 2.2. Funkcie systému

Aplikácia bude ovládaná užívateľmi, ktorí sa budú prihlasovať. Takisto bude pravidelne aktualizovať súbor na kontrolu údajov, ktorý bude získavať z pravidelného reportu z OneDrive.

Pred začatím skenovania sa používateľ prihlási pomocou čiarového kódu. Ďalej si zvolí zákazníka, pre ktorého je zásielka určená. A nakoniec si vyberie ŠPZ vozidla, do ktorého sa budú nakladať kontrolované vozíky. Každý zákazník má svoj vlastný pattern, ktorý je uložený v databáze.

Po začatí skenovania vozidla bude možné skenovanie údajov vozíka, prvého produktu vozíka a posledného produktu vozíka. Podľa týchto údajov aplikácia kontroluje zhodu s reportom. Po skončení práce bude môcť pracovník použiť uzavretie vozidla.

V prípade chyby pri skenovaní aplikácia na chybu upozorní, následne sa umožní jej odstránenie alebo potvrdenie od

kontrolóra, že táto chyba je v poriadku. Chybu si aplikácia zaznamenáva do databázy.

Taktiež v prípade, že užívateľ uzavrie vozidlo priskoro, aplikácia umožní znovu otvorenie vozidla, ale len do začatia skenovania ďalšieho vozidla.

Aplikácia exportuje údaje o vozíkoch naložených do vozidla po ukončení skenovania tohto vozidla alebo po začatí skenovania ďalšieho vozidla.

System bude možno nainštalovať na viacero zariadení. Každé zariadenie obsluhuje prevažne jeden užívateľ.

System si udržiava záznamy činností v textovom súbore v databáze.

Administrátor a kontrolór majú možnosť konfigurovať zákazníka s patternom a vozidlá. Administrátor môže navyše upravovať zoznam pracovníkov.

## 2.3. Charakteristika používateľov

S aplikáciou budú interagovať používatelia, ktorí majú jednu z nasledujúcich 3 rolí: administrátor, užívateľ a kontrolór. Cieľom rozdelenia je, aby niektorí pracovníci mohli systém spravovať v prípade personálnych, alebo iných zmien.

**Užívateľ:**

Pracovník, ktorý môže aplikáciu používať na skenovanie produktov, kvôli ich kontrole. Môže zatvárať a otvárať autá.

**Kontrolór:**

Má práva užívateľa. V prípade nedostupnosti reportu má oprávnenie vykonať potvrdenie manuálnej kontroly. Vie

nakonfigurovať patterny a config v databáze pomocou aplikácie.

Administrátor:

Má práva kontrolóra. Navyše môže v aplikácii zadávať, vykonávať personálne zmeny.

## 2.4. Všeobecné obmedzenia

System vyžaduje prístup na internet, aby naša mobilná aplikácia mohla komunikovať s databázou (pripojenie na sieť, kábel).

## 2.5. Predpoklady a závislosti

Aplikácia predpokladá, že report bude exportovaný každých 15 minút. Aplikácia vyžaduje manuálne uzavretie každého vozidla. Bude pracovať len s poslednou verziou reportu, lebo obsahuje všetky aktuálne platné informácie.

# 3. Špecifikácie a požiadavky

## 3.1. Funkčné požiadavky

### 3.1.1. Import dát

- 3.1.1.1. Aplikácia získava report vo formáte .csv s dátami o vozíkoch každých 15 minút z OneDrive alebo zo servera.
- 3.1.1.2. Ak report neexistuje alebo je prázdny, je možné vykonávať audit, ale správnosť každého vozíka musí potvrdiť pracovník s rolou kontrolóra naskenovaním svojho kódu.
- 3.1.1.3. Aplikácia bude komunikovať s databázou cez HTTPS protokol, ktorý bude zabezpečený pomocou API kľúča.

- 3.1.1.4. Údaje o zákazníkoch (pattern, ŠPZ vozidla) bude aplikácia získavať z databázy.
- 3.1.1.5. Prihlasovacie údaje o pracovníkoch bude (číslo čiarového kódu, meno, rola) aplikácia získavať z databázy.

### 3.1.2. Pracovníci

- 3.1.2.1. Zoznam pracovníkov je uložený v databáze obsahujúcej ich číslo čiarového kódu, meno, priezvisko a rolu (užívateľ, administrátor, kontrolór).
- 3.1.2.2. Administrátor môže pridávať do databázy, odstraňovať z databázy pomocou aplikácie nové vozidlo (ŠPZ), pracovníka (číslo čiarového kódu, meno, rola), nového zákazníka s configom a patternom.
- 3.1.2.3. Administrátor, kontrolór môže pristupovať k exportom pomocou webovej aplikácie. Samozrejme prístup je na úrovni čítania.

### 3.1.3. Audit vozidla

#### 3.1.3.1. Začiatok auditu

- 3.1.3.1.1. Pre jedného zákazníka môže prebiehať audit len jedného vozidla, vykonáva ho jeden pracovník.
- 3.1.3.1.2. Pracovník zodpovedný za audit sa prihlasuje naskenovaním svojho čiarového kódu.
- 3.1.3.1.3. Pracovník zodpovedný za audit si zvolí zákazníka pre ktorého bude zásielka určená.
- 3.1.3.1.4. Pracovník zodpovedný za audit si zvolí vozidlo na základe ŠPZ do ktorého bude skenovať vozíky.



- 3.1.3.2. Aplikácia zobrazuje počet už naložených vozíkov aj maximálny možný počet vozíkov na základe patternu.

### 3.1.3.3. Kontrola vozíka

- 3.1.3.3.1. Aplikácia si zapamätá čas začiatku auditu vozíka.
- 3.1.3.3.2. Aplikácia si uloží kód vozíka naskenovaný pracovníkom, následne kód prvého a posledného výrobku vo vozíku.
- 3.1.3.3.3. Aplikácia porovná zistené kódy s údajmi z reportu. Pozri 3.1.4.1.
- 3.1.3.3.4. Aplikácia zároveň kontroluje či užívateľ neprekročil kapacitu vozidla. Pozri 3.1.4.2.
- 3.1.3.3.5. Aplikácia si zapamätá čas ukončenia auditu vozíka.

### 3.1.3.4. Zatvorenie vozidla

- 3.1.3.4.1. Pracovník môže v priebehu auditu zatvoriť vozidlo, aplikácia si zaznamenáva čas posledného uzavretia.
- 3.1.3.4.2. Pracovník môže po uzatvorení vozidla znovu otvoriť vozidlo.

### 3.1.3.5. Koniec auditu

- 3.1.3.5.1. Audit uzavrie pracovník.
- 3.1.3.5.2. Aplikácia na začiatku auditu nového vozidla, odhlásení užívateľa alebo vypnutí aplikácie bude exportovať súbor vo formáte PDF .

## 3.1.4. Riešenie chýb

- 3.1.4.1. Keď aplikácia objaví nezhodu zistených údajov s dátami z reportu (prvé a posledné TLS vo vozíku, zlé poradie vozíkov v rámci

konkrétneho typu vozíkov), upozorní na to pracovníka.

- 3.1.4.2. Záznam o chybe je evidovaný v databáze.
- 3.1.4.3. Chybu je možné odstrániť alebo kontrolór umožní pokračovať v audite ostatných vozíkov rovnakého typu naskenovaním svojho kódu.
- 3.1.4.4. Keď pracovník odstráni chybu, má možnosť znovu naskenovať opravený vozík a pokračovať v audite jeho typu.
- 3.1.4.5. Ak bol prekročený maximálny počet naskenovaných vozíkov, alebo bol z určitého typu naložený menší počet vozíkov, ako je stanovené v patterne, na konci auditu bude aplikácia vyžadovať naskenovanie kódu kontrolóra.
- 3.1.5. Užívateľ sa môže odhlásiť po zavretí vozidla, čím sa automaticky ukončí audit vozidla 3.1.3.5.3 .
- 3.1.6. Užívateľ môže vypnúť aplikáciu po zavretí vozidla, čím sa automaticky ukončí audit vozidla 3.1.3.5.3 .

### 3.1.7. Rozšírené požiadavky (voliteľné)

- 3.1.7.1. Sledovanie pohybu jednotlivých vozíkov medzi zadávateľom a jeho zákazníkmi (kontrola opotrebenia vozíkov).
- 3.1.7.2. Aplikácia na základe dĺžky kódu zvolí typ kódu TLS alebo IONO.
- 3.1.7.3. Možnosť nastavenia ukladania exportov do systému microsoft onedrive.

## 3.2. Kvalitatívne požiadavky

- 3.2.1. Systém musí byť intuitívny a zároveň musí byť časovo efektívny.
- 3.2.2. Systém musí obsahovať konzistenciu dát a to v takom rozsahu, aby aj po náhlom výpadku a opätovnom obnovení napájania zariadenia mohol pracovník naďalej pokračovať v predošlom stave.

- 3.2.3. Systém musí obsahovať funkčnú a v rámci prostredia systému neobíditeľnú autorizáciu / autentifikáciu užívateľa.
- 3.2.4. Systém musí udržiavať dátovú redundanciu a to v takom rozsahu, aby sa celý obsah databázy udržiaval aj v zariadení na ktorom beží aplikácia a v prípade nesúladu zosynchronizovala zmeny.
- 3.2.5. Systém musí obsahovať možnosť preložiť všetky, bežnému užívateľovi, videné hlášky.

Návrh

## **Audit prepráv**

Tvorba informačných systémov

2022/2023

**Roman Božik**

**Jozef Kolek**

**Terézia Strišovská**

**Michal Uhrin**

<b>1. Úvod</b>	<b>4</b>
1.1. Účel dokumentu	4
1.2. Zameranie a rozsah	4
1.3. Prehľad nasledujúcich kapitol	4
<b>2. Podrobná špecifikácia vonkajších interfejsov</b>	<b>5</b>
2.1. Komunikácia s inými zariadeniami	5
<b>3. Dátový model</b>	<b>8</b>
3.1. Entitno relačný model auditu prepráv	8
3.2. Súbory	10
<b>4. Používateľské rozhranie</b>	<b>11</b>
<b>5. UML Diagramy</b>	<b>19</b>
5.1. UML Component diagram	19
5.2. UML Class diagram	20
5.3. UML State diagram	22
<b>6. Návrh implementácie</b>	<b>23</b>
6.1. Všeobecný popis vývoja aplikácie	23
6.2. Front-end	23
6.3. Back-end	23

1. Úvod
  - 1.1. Účel dokumentu
  - 1.2. Zameranie a rozsah
  - 1.3. Prehľad nasledujúcich kapitol
2. Podrobná špecifikácia vonkajších interfejsov
  - 2.1. Komunikácia s inými zariadeniami
3. Dátový model
  - 3.1. Entitno relačný model linky
  - 3.2. Súbory
4. Používateľské rozhranie
5. UML Diagramy
  - 5.1. UML Component diagram
  - 5.2. UML Class diagram
  - 5.3. UML State diagram
6. Návrh implementácie
  - 6.1. Všeobecný popis vývoja aplikácie
  - 6.2. Front-end
  - 6.3. Back-end
7. Plán implementácie

# 1 Úvod

## 1.1 Účel dokumentu

Tento dokument slúži ako kompletný a detailný návrh informačného systému auditu prepráv. Obsahuje všetky informácie potrebné pre vysvetlenie a pochopenie funkcionality ako aj spôsobu implementácie systému. Tento dokument je primárne určený pre vývojárov. Obsah v tomto dokumente zahŕňa všetky požiadavky z katalógu požiadaviek.

## 1.2 Zameranie a rozsah

Pre prácu s týmto dokumentom sa vyžaduje oboznámenie sa s katalógom požiadaviek, ktorý s návrhom úzko súvisí. Tento dokument špecifikuje všetky požiadavky z katalógu požiadaviek. Ďalej špecifikuje vonkajšie interfejsy, formáty súborov a komunikačné protokoly. Obsahuje taktiež celkový návrh používateľského prostredia vrátane vizualizácie. Návrh implementácie opisuje diagramy konkrétne komponentný diagram, stavový diagram a class diagram.

## 1.3 Prehľad nasledujúcich kapitol

Nasledujúce kapitoly sa venujú vonkajším interfejsom, dátovému modelu, používateľskému rozhraniu a jeho vizualizácií.

## 2. Špecifikácia vonkajších interfejsov

### 2.1 Komunikácia s inými zariadeniami

#### **REST API**

REST API je rozhranie, vďaka ktorému je možné bezpečne komunikovať medzi viacerými rozhraniami za použitia https požiadaviek.

#### **PostgreSQL**

PostgreSQL je relačná databáza slúžiaca ako perzistentné dátové úložisko aplikácie. Aplikácia bude s databázou komunikovať prostredníctvom REST API.

#### **WSGI**

WSGI je univerzálne rozhranie medzi webovým serverom a webovou aplikáciou.

#### **Apache**

Apache je webový server, ktorý bude prijímať https požiadavky a presmerovávať ich podľa potreby na WSGI alebo aplikácie priamo bežiacie na webovom servery.

#### **Microsoft OneDrive**

Microsoft OneDrive je cloudové úložisko z ktorého si aplikácia bude sťahovať reporty.



## **Webová administrácia**

Pomocou webovej administrácie si bude možné vybrať export vozidla a zároveň konkrétne informácie o danom audite.

### **Očakávané entity:**

Vzdialený server – Bude server prístupný z internetu na ktorom bude bežať databáza, webový server, WSGI a webová administrácia.

Zariadenie – teda každé zariadenie, na ktorom bude bežať aplikácia. Očakáva sa, že každé zariadenie sa nachádza v sieti s veľmi limitovaným prístupom (iba HTTPS).

Produkty z rodiny Microsoft – V tejto chvíli sa očakáva použitie služby microsoft one drive ako úložisko pre reporty.

### **Očakávané scenáre:**

Sťahovanie reportov z cloudového úložiska bude aplikácia vykonávať každých 15 minút za pomoci python knižnice Micorosft Graph.

Aplikácia si bude v databáze držať dáta o pracovníkoch, logy, dáta o vozidlách a exportoch. Pristupovať k nim bude cez HTTPS požiadavky za pomoci python knižnice flask. Toto API bude bežať na vzdialenom serveri s nainštalovaným Apache a WSGI

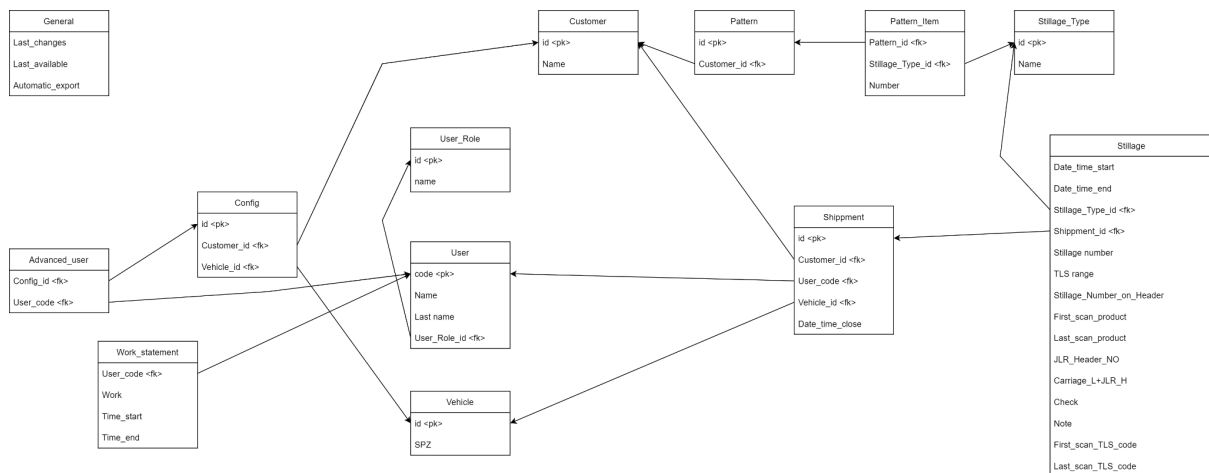
Kód pre predstavu:

<https://github.com/TIS2022-FMFI/transport-audit/pull/11/files>

Aplikácia bude po audite posielat' do databázy podrobné dáta o prebehnutom audite. Tieto dáta si bude webová administrácia na to určená z databázy ťahať a umožní si užívateľovi zvoliť ľubovoľné z nich a tie preklopiť do formátu PDF s možnosťou si výsledné PDF stiahnuť.

Aplikácia bude obsahovať pre prípad výpadku internetového spojenia internú databázu, ktorá sa bude synchronizovať s PostgreSQL databázou bežiacou na servery. V prípade dátového konfliktu bude riešenie buď na výbere pracovníka, alebo sa automaticky uprednostnia dáta vytvorené neskôr v závislosti na konkrétnej situácii.

## 3. Dátový model



### 3.1. Entitno relačný model auditu prepráv

General: všeobecné nastavenia a informácie

Last\_changes – čas kedy boli posledné zmeny vykonane v databáze

Last\_avaliable – čas kedy bola databáza naposledy online.

Automatic\_export – či sa po uzatvorení vozidla bude exportovať záznam vo formáte pdf a odošle niekam.

Advanced\_user: pracovník ktorý má právo na nakladanie vozidla

Config\_id – natiahne si pomocou configu ktoré vozidlo pre akého zákazníka môže skenovať

User\_code – natiahne sa kód užívateľa, jeho vlastná identita

Work\_statement: určuje ktorý užívateľ koľko času skenoval vozidla

User\_code – ktorý užívateľ to bol

Work – typ pracovnej činnosti, ktorú užívateľ v aplikácii robil (menil patterny, upravoval zoznam užívateľov etc.).

Time\_start – začatie vykonávania činnosti

Time\_end – ukončenie činnosti

Config: tabuľka údaje sú súčasťou výstupu, ktorý bude zadávateľ exportovať z databázy

Id – jeho vlastná identita

Customer\_id – zákazník ktorému sa vozidlo nakladá

User\_id – id užívateľa ktorý dane vozidlo nakladá

Vehicle\_id – poznávacia značka vozidla do ktorého sa nakladá

User\_Role: určuje práva užívateľov

Id – id roly

Name – meno užívateľskej roly

User: obsahuje zoznam všetkých užívateľov ktorí budú mať do aplikácie prístup

Code – unikátny kód pomocou ktorého sa prihlásia do aplikácie

Name – meno užívateľa

Last\_name – Priezvisko užívateľa

User\_role\_id – užívateľská rola

Vehicle: Zoznam vozidiel do ktorých sa nakladalo

Id - vozidla

SPZ – poznávacia značka vozidla

Pattern: kapacita vozidla, v počte vozíkov ktoré sa do neho zmestia

Id – neskôr potrebujeme

Customer\_id – pre akého zákazníka tento pattern je používaný, pretože každý zákazník má nejaké typické patterny ktoré opakovane používa

## 3.2. Súbory

[Datovy model](#)

[Výstupy ktore som zahríňal pri navrh](#)

## 4. Používateľské rozhranie

### 4.1. Prihlásenie

SCREEN č. 1



Po naskenovaní kódu prebehne prihlásenie a dostanem na screen č.2

Kliknutím sa zobrazí možnosť na výber jazykov

SCREEN č. 2



Kliknutím sa prepne na screen č.8

Kliknutím sa prepne na screen č.11

Kliknutím sa prepne na screen č. 13

Kliknutím sa premene na screen č.15

Kliknutím sa premene na screen č. 3

Kliknutím sa užívateľ odhlási a prepne sa na screen č.1

## 4.2. Audit vozidla

SCREEN Č. 3

Zakaznik

Vozidlo

POKRAČOVAŤ

ODHLÁSIŤ

Scroll okno na výber zákazníka

Scroll okno na výber ŠPZ vozidla.  
Zobrazí sa po výbere zákazníka

Kliknutím sa pokračuje

Kliknutím sa odhlási

SCREEN Č. 4

Naskenuj vozík 0

Naskenuj prvý produkt 0

Naskenuj posledný produkt 0

SPÄŤ

UZAVRIEŤ KAMIÓN

PRIDAŤ/UBRAŤ

Grafické zobrazenie naplnenosti vozidla

Zobrazí sa číslo naskenovaného vozíka

Zobrazí sa číslo naskenovaného prvého produktu

Zobrazí sa číslo naskenovaného posledného produktu

Kliknutím sa prepne na screen č.3

Kliknutím sa uzavrie vozidlo a prepne sa na screen č.7

Kliknutím môže pokračovať mimo pattern

## SCREEN Č. 5

Graphic representation of vehicle fullness

Naskenuj vozík 0

Naskenuj prvý produkt 0

Naskenuj posledný produkt 0

Vozík je v poriadku

SPÄT

UZAVRIEŤ KAMIÓN

PRIDAŤ/UBRAŤ

Grafické zobrazenie naplnenosti vozidla

Zobrazí sa číslo naskenovaného vozíka

Zobrazí sa číslo naskenovaného prvého produktu

Zobrazí sa číslo naskenovaného posledného produktu

Oznámenie že sken prebehol úspešne

Kliknutím sa prepne na screen č.3

Kliknutím sa uzavrie vozidlo a prepne sa na screen č.7

Kliknutím môže pokračovať mimo pattern

## SCREEN Č. 6

Graphic representation of vehicle fullness

Naskenuj vozík 0

Naskenuj prvý produkt 0

Naskenuj posledný produkt 0

Chybná kontrola

SPÄT

NASLEDUJÚCI SKEN

PRIDAŤ/UBRAŤ

Grafické zobrazenie naplnenosti vozidla

Zobrazí sa číslo naskenovaného vozíka

Zobrazí sa číslo naskenovaného prvého produktu

Zobrazí sa číslo naskenovaného posledného produktu

Oznámenie že sken neprebehol úspešne

Kliknutím sa prepne na screen č.3

Kliknutím sa umožní ďalší sken.

Kliknutím môže pokračovať mimo pattern

Zobrazí sa TOAST, pre otázku či sa chyba bude odstraňovať chybu alebo nie



## SCREEN č. 7



Kliknutím sa otvorí vozidlo a prepne sa na screen č.4

Kliknutím skončí audit vozidla a prepne sa na screen č.4

## 4.3. Zamestnanci

## SCREEN č. 8



Scroll zamestnancov s číslom, rolou a možnosťou označiť jedného

Kliknutím sa prepne na screen č.9

Kliknutím sa prepne na screen č.10 , musí byť označený jeden zamestnanec

Kliknutím sa odstráni zamestnanec, musí byť označený jeden zamestnanec

Kliknutím sa prepne na screen č.2

## SCREEN č. 9



Číslo: \_\_\_\_\_

Rola: \_\_\_\_\_

Meno: \_\_\_\_\_

PRIDAJ

SPÄŤ

Textový input len pre čísla

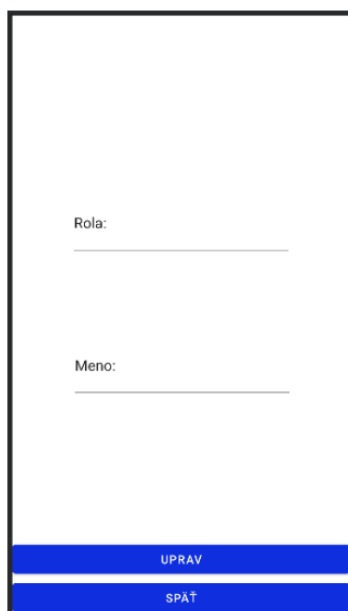
Scroll možných rol

Textový input

Kliknutím sa pridá zamestnanec, prípadne upozorní na chybu alebo korektné pridanie

Kliknutím sa prepne na screen č.8

## SCREEN č. 10



Rola: \_\_\_\_\_

Meno: \_\_\_\_\_

UPRAV

SPÄŤ

Scroll možných rol

Textový input

Kliknutím sa upraví zamestnanec, prípadne upozorní na chybu alebo korektnú úpravu

Kliknutím sa prepne na screen č.9

## 4.4. Zákazníci

SCREEN Č. 11

Cislo zakaznika    Označ

PRIDAJ

VYMAŽ

SPÄT

Scroll zákazníkov s číslov  
a možnosťou označiť jedného

Kliknutím sa odstráni zákazník, musí byť označený jeden zákazník

Kliknutím sa prepne na screen č.2

SCREEN Č. 12

Číslo:

Meno:

PRIDAJ

SPÄT

Vygenerované číslo/možnosť zmeny

Názov zákazníka

Kliknutím sa pridá zákazník, prípadne upozorní na chybu alebo korektné pridanie

Kliknutím sa prepne na screen č.11



## 4.6. Pattern

### SCREEN č. 15

Cislo patternu   Zákazník   Označ

PRIDAJ

VYMAŽ

SPÄŤ

Scroll patternov s číslom, zákazníkom  
a možnosťou označiť jedného

Kliknutím sa prepne na screen č.16

Kliknutím sa odstráni pattern, musí byť označený jeden pattern

Kliknutím sa prepne na screen č.2

### SCREEN č. 16

Zákazník:

Vlož patern:

PRIDAJ

SPÄŤ

Scroll zákazníkov

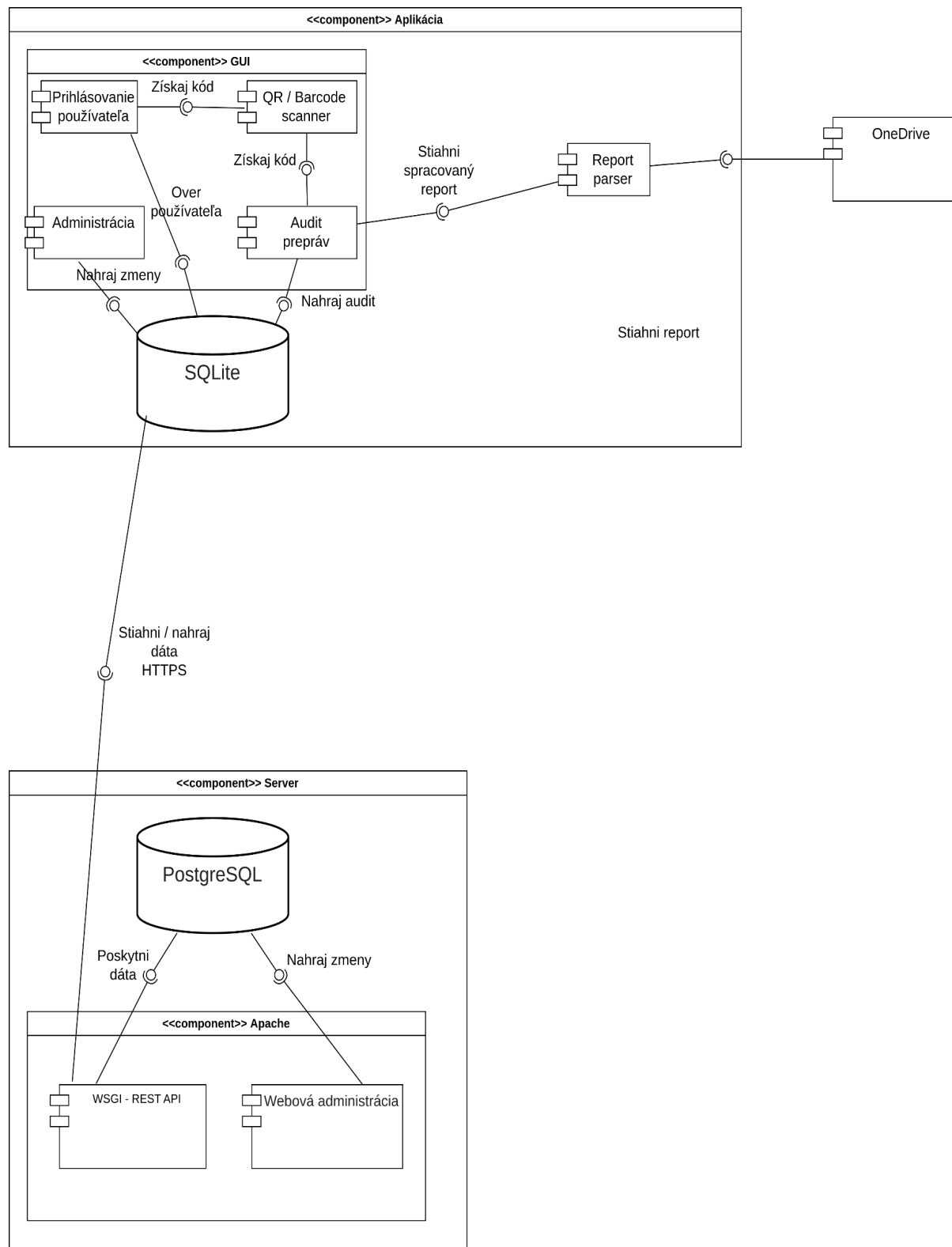
Scroll type stillage, po vybratí sa  
vygeneruje text pole s tipom  
a číselný input na počet KS

Kliknutím sa pridá pattern, prípadne upozorní na chybu alebo korektné pridanie

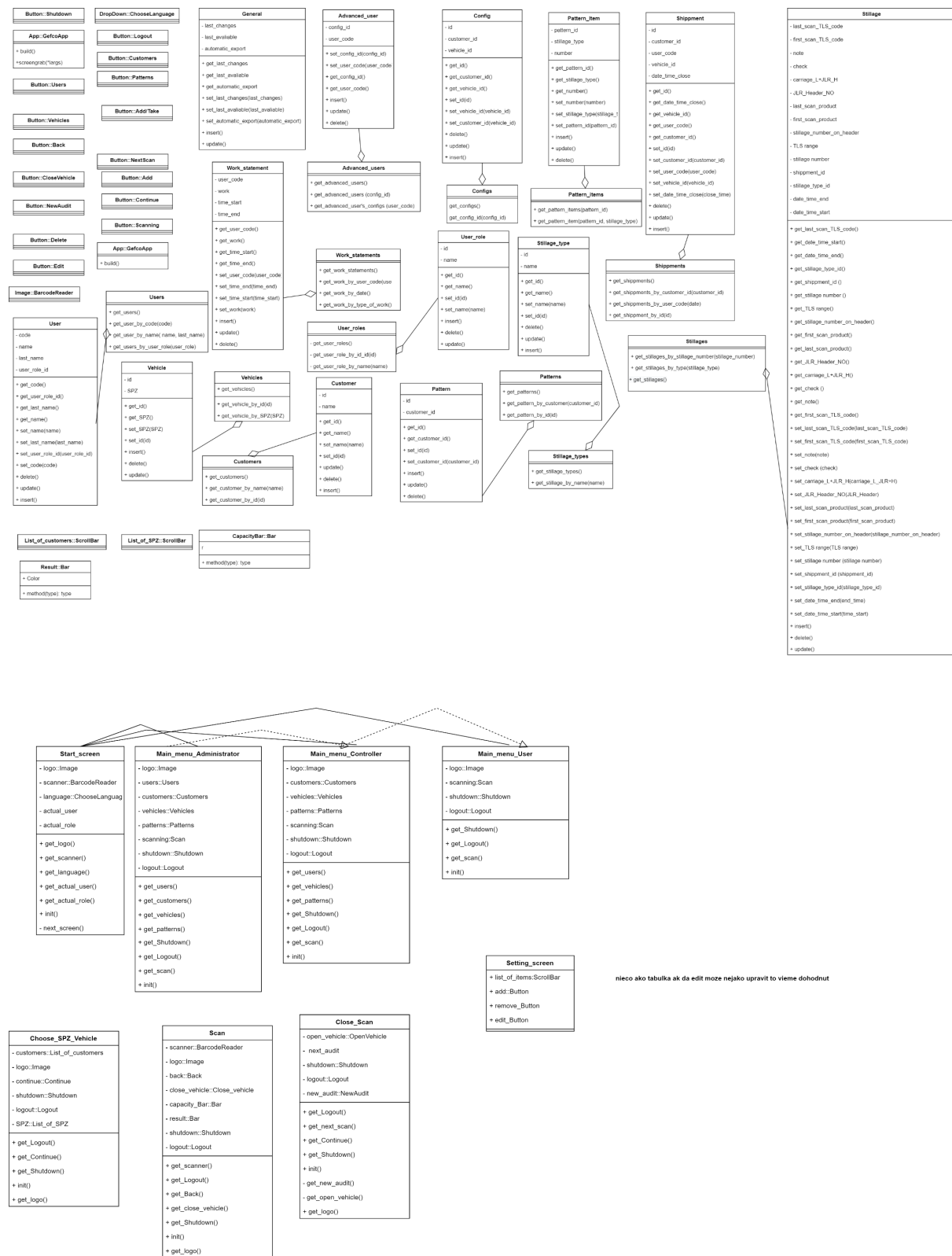
Kliknutím sa prepne na screen č.15

## 5. UML diagram

### 5.1 UML Component diagram



## 5.2. UML Class diagram



Dizajn, pozícia všetko sa nastavuje cez .kw file ktorý nesie názov app ktorý buildujeme

Buttony:

- majú class každý ma nejakú funkciu ktorá sa na klik vykoná

Dropdowns, Scroll:

- majú class slúži na select to si neskôr vytiahneme

Start-screen:

- bude si volať jednotlivé komponenty ako buttony, kameru, logo
- next\_screen() – na základe porovnania kódu s databázou ho pošle na príslušne main\_menu

Main\_menu\_Administrator, Main\_menu\_Controller,  
Main\_menu\_User:

- natiahne si buttony a na základe kliknutia sa posúva ďalej

Choose\_SPZ\_Vehicle:

scroll menu kde si pracovník vyberie zákazník a potom vozidlo

opäť natiahne si buttoniky je to v prípade že sa pracovník hocijakej pozície rozhodne skenovať

Scan:



- Potiahne kameru vyťahuje ňou kódky porovnáva
- Vola si Bar ktorý priebežne aktualizuje
- Ďalší bar zobrazuje či skenovanie bolo ok

Close\_Scan:

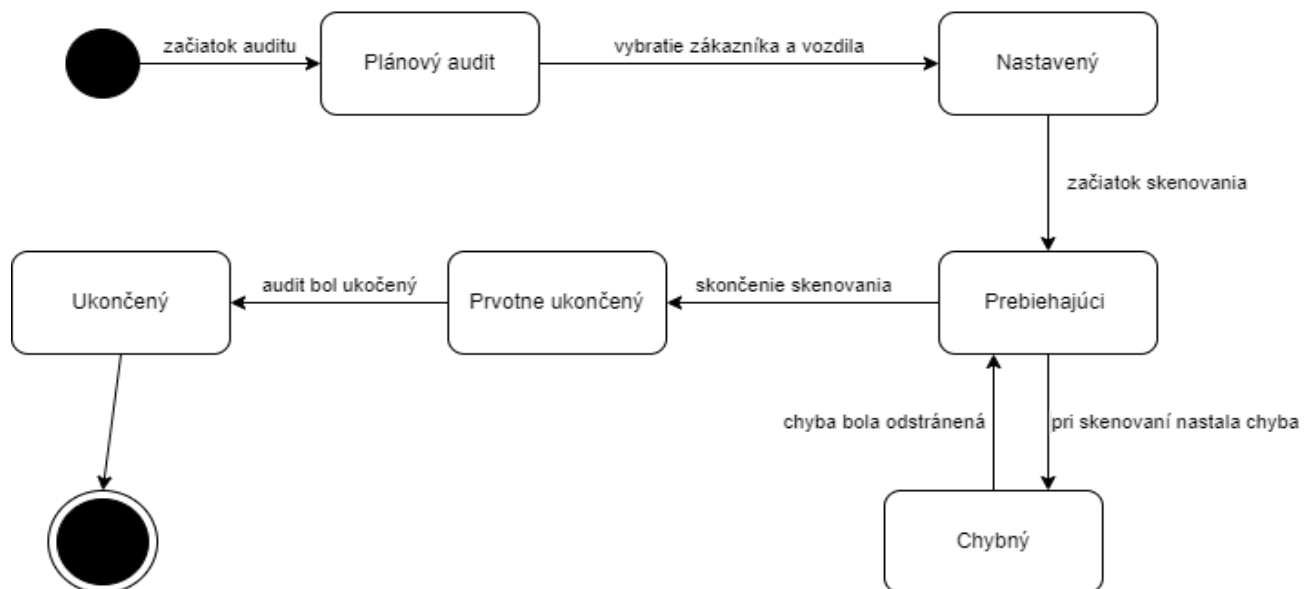
- Skenovanie vozidla bolo dokončené, plus buttony na next actions čo sa da robiť

Setting\_screen:

- Multifunkčne dotiahnem buttony

### 5.3. UML State diagram

State Diagram, Audit vozidla



State diagram zobrazuje entitu Audit vozidla.

# Plán implementácie projektu

Pripravené  
[25. november 2022 ]

## Ciele

Ciele projektu sú:

1. Dokončiť potrebnú dokumentáciu projektu, potrebnú pre definíciu a vývoj.
2. Vyvinúť úspšene skompilovaný zdrojový kód.
3. Otestovanie aplikácie na pracovisku.

## Projekt

### Projekt

Aplikácia na prácu auditu pri nakladaní vozidiel.

### Rozsah

Tento projekt bude pozostávať z prípravy dokumentácie, vývoja aplikácie a jej otestovania.

## Časová os

FÁZA	POPIS	KONCOVÝ DÁTUM
Prípravná fáza	Tím pripravím všetky dokumenty, plány a pod. na realizáciu projektu.	6.12.2022
Určenie úloh	Každý člen tímu dostane svoje vlastné úlohy, ktoré musia splniť.	2.12.2022

## Akčný plán

### CIEĽ 1

Dokončiť dokumentáciu projektu 6.12. 2022.

AKCIA	PRIRADENÝ	KONCOVÝ DÁTUM
Vytvorenie dokumentácie	Roman Božík (zodpovedná osoba), Jozef Kolek, Terézia Strišovská, Michal Uhrin	6.12.2022

**CIEĽ 2**  
Vyvinúť aplikáciu.

AKCIA	KONCOVÝ DÁTUM
Vytvorenie databázy pre prácu s dátami aplikáciami.	10.12.2022
Vytvorenie logických častí kódu aplikácie pre prácu s databázou	14.12.2022
Otestovanie správneho fungovania komunikácie aplikácie s databázou	16.12.2022
Vytvorenie časti aplikácie slúži na priebeh auditu	18.12.2022
Vytvorenie všetkých ostatných funkcií aplikácie ako výber jazyka a pod.	20.12.2022
Vytvorenie GUI aplikácie a pripojenie na zdrojový kód	21.12.2022
Otestovanie mobilnej aplikácie plného rozsahu	21.12.2022
Vytvorenie webového rozhrania pre prácu s databázou	21.12.2022
Otestovanie webového rozhrania	21.12.2022