

Návrh prototypu SW platformy – spôsob evidencie, zberu dát a komunikácie

Life Defender - Ochranca života



Táto publikácia vznikla vďaka podpore v rámci Operačného programu Integrovaná infraštruktúra pre projekt *Life Defender – Ochranca života*, ktorý je spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.



SFÉRA, a.s. • Karadžičova 2 • 811 08 Bratislava
tel.: +421 2 502 13 142

ISBN 978-80-89778-11-9
© SFÉRA, a.s., 2022



EURÓPSKA ÚNIA
Európsky fond regionálneho rozvoja
OP Integrovaná infraštruktúra 2014 – 2020



MINISTERSTVO
DOPRAVY A VÝSTAVBY
SLOVENSKEJ REPUBLIKY



MINISTERSTVO
ŠKOLSTVA, VEDY,
VÝSKUMU A ŠPORTU
SLOVENSKEJ REPUBLIKY

Táto publikácia je dielom kolektívu autorov:
Krbat'a Rastislav, Holiš Martin, Minárik Michal

Ostatní autori:
Rozinajová Viera, Lucká Mária, Bou Ezzeddine Anna, Burget Radim, Galajda Miroslav

OBSAH

1 ÚVOD	6
1.1 Členenie dokumentu	6
2 METODIKA RIADENIA PROJEKTU	8
2.1 Projektové riadenie.....	8
2.2 Projektový tím	9
2.3 Koordinačné stretnutia	10
3 EXPERIMENTÁLNY VÝVOJ PROTOTYPU SW PLATFORMY A CLOUDOVÉHO ÚLOŽISKA ...	11
3.1 Prototyp pre Domácu karanténu	11
3.1.1 <i>Popis používateľských scenárov</i>	11
3.1.1.1 <i>Scenár 1</i>	11
3.1.2 <i>Dátový model</i>	13
3.1.3 <i>Webové rozhranie</i>	14
3.1.3.1 <i>Hlavička</i>	14
3.1.3.2 <i>Obsah</i>	14
3.1.3.3 <i>Dashboard</i>	14
3.1.3.4 <i>Pacienti</i>	14
3.1.3.4.1 <i>Detail pacienta</i>	15
3.1.3.5 <i>Lekári</i>	16
3.1.3.5.1 <i>Detail lekára.....</i>	17
3.1.3.6 <i>Pozvánky</i>	17
3.1.3.7 <i>Zoznam zariadení</i>	17
3.1.3.8 <i>Ovládacie prvky</i>	18
3.1.3.9 <i>Rozmiestnenie portálu</i>	19
3.1.4 <i>Architektúra systému</i>	23
3.1.4.1 <i>Logická architektúra systému</i>	24
3.1.4.1.1 <i>Typy integrácií</i>	25
3.1.4.1.2 <i>Komponenty systému</i>	25
3.1.4.2 <i>Fyzická architektúra systému</i>	25
3.1.4.2.1 <i>Prostredia</i>	26
3.1.4.2.2 <i>Bezpečnosť.....</i>	26
3.1.4.2.3 <i>Autentifikácia</i>	26
3.1.4.2.4 <i>Autorizácia</i>	27
3.2 Prototyp pre Automatickú testovaciu bunku	27
3.2.1 <i>Popis používateľských scenárov</i>	27
3.2.1.1 <i>Scenár 1</i>	27
3.2.1.2 <i>Scenár 2</i>	29
3.2.2 <i>Webové rozhranie</i>	29
3.2.2.1 <i>Ovládacie prvky</i>	30
3.2.3 <i>Architektúra systému</i>	30
3.2.3.1 <i>Logická architektúra systému</i>	31
3.2.3.1.1 <i>Typy integrácií</i>	32
3.2.3.1.2 <i>Komponenty systému</i>	32
3.2.3.2 <i>Fyzická architektúra systému</i>	33
3.2.3.2.1 <i>Prostredia</i>	34
3.2.3.2.2 <i>Bezpečnosť.....</i>	34

3.2.3.2.3	Autentifikácia	34
3.2.3.2.4	Autorizácia	34
3.3	Prototyp pre Dezinfekčného robota.....	35
3.3.1	Popis používateľských scenárov	35
3.3.1.1	Scenár 1	35
3.3.1.2	Scenár 2	35
3.3.1.3	Scenár 3	36
3.3.2	Webové rozhranie	37
3.3.2.1	Ovládacie prvky	37
3.3.2.2	Rozmiestnenie portálu	41
3.3.3	Architektúra systému	42
3.3.3.1	Logická architektúra systému	42
3.3.3.2	Typy integrácií	43
3.3.4	Komponenty systému	44
3.3.5	Fyzická architektúra systému	44
3.3.5.1	Prostredia	45
3.3.5.2	Bezpečnosť	45
3.3.5.3	Autentifikácia	45
3.3.5.4	Autorizácia	45
4	EXPERIMENTÁLNY VÝVOJ PROTOTYPU MOBILNEJ APLIKÁCIE	46
4.1	Prototyp pre domácu karanténu	46
4.1.1	Zber a synchronizácia dát	46
4.1.2	Popis GUI	46
4.1.3	Architektúra	47
4.1.4	Technológie	48
4.1.5	Rozhrania	48
4.2	Prototyp pre Dezinfekčného robota.....	48
4.2.1	Popis GUI	48
4.2.2	Architektúra	49
4.2.3	Technológie	49
4.2.4	Rozhrania	49
5	EXPERIMENTÁLNY VÝVOJ PROTOTYPU MODULU POKROČILEJ ANALÝZY A VIZUALIZÁCIE DÁT	50
5.1	Prototyp modulu vizualizácie dát.....	50
5.1.1	Kibana	50
5.1.2	Zoznam grafických metód	52
5.1.2.1	Grafické zobrazenie spojitých premenných	52
5.1.2.2	Charakteristiky spojitých premenných	53
5.1.2.3	Histogram	53
5.1.2.4	Krabicový diagram	54
5.1.2.5	Paralelné súradnice	55
5.2	Prototyp dátového modelu na zbieranie, analýzu a vyhodnocovanie symptómov pomocou analytických nástrojov s využitím umelej inteligencie	55
5.2.1	Dátová analytika pre zdravotníctvo	55
5.2.1.1	Detekcia anomálií	56
5.2.1.2	Kategorizácia anomálií	57
5.2.1.3	Anomálie v jednorozmernom priestore	58

5.2.1.4 Anomálie vo viacrozmerom priestore	60
5.2.1.5 Metódy detekcie anomálií	60
5.2.1.6 Hodnotiace kritériá	61
5.2.1.7 Analýzy hlavných komponentov	61
5.2.1.8 K-najbližších susedov	62
5.2.1.9 Algoritmus K-means	62
5.2.1.10 Podporné vektory	65
5.2.1.11 DB scan	65
5.2.2 Posúdenie dynamiky predikcie zdravotného stavu analýzou a grafickým zobrazením diferencií prvého rádu	67
5.2.2.1 Stacionarita a diferencovanie	68
5.2.2.2 Anomálie v časových radoch	68
5.2.2.3 Metóda blokových maxím	68
5.2.2.4 Peaks-over-treshold	69
5.2.3 Konceptie detekcie anomálií a extrémnych hodnôt	69
5.2.3.1 Využitie predikčného intervalu spoľahlivosti	70
5.2.3.2 Využitie štatistického profilovania	71
5.2.3.3 Využitie prístupu založeného na zhlukovaní bez učiteľa	72
5.2.4 Zhrnutie	73
6 ZÁVER	74
7 ZDROJE	76
8 ZOZNAM OBRÁZKOV	77
9 ZOZNAM TABULIEK	78

1 ÚVOD

Cieľom dokumentu je popísať vývoj softvérovej časti projektu a jeho nadväznosť na hardvérovú časť, ktorého činnosti sa realizovali v termíne 1.6.2021 – 30.6.2022. Dokument je výstupom projektu Life Defender – Ochranca života, kód projektu v ITMS: 313011ASQ6 v rámci *míľníka 2 - Návrh prototypu SW platformy (spôsob evidencie, zberu dát a komunikácie)* pre aktivitu H1 - Riešenie SW platformy na integrovanie evidencie návštevníkov, zberov dát z existujúceho HW ako i prototypov nového HW do jednotného informačného systému Life Defender – Ochranca života - prototyp (EV80) a aktivitu H2 - Riešenie SW platformy na integrovanie evidencie návštevníkov, zberov dát z existujúceho HW, ako i prototypov nového HW do jednotného informačného systému Life Defender – Ochranca života - prototyp (EV80) – flexibilita 15 %.

Ako sme konštatovali vo výstupe projektu k jeho prvému míľniku, absencia účinných liečebných metód a vakcinačných prostriedkov mala za následok nutnosť izolácie a vyžadovala cielečné zastavenie mnohých spoločenských aj obchodných aktivít s cieľom minimalizovať šírenie vírusu. Pre správne rozhodovanie aktivovania preventívnych opatrení je kľúčovým prvkom zber dát a ich následná analýza. Na základe správne nastavenej metriky a vyhodnocovacích metód je možné rozhodovanie o efektívnych opatreniach zamedzujúcich šíreniu nákazy. V prípade analýzy dát o jednotlivých anamnézach a priebehu ochorenia konkrétnych pacientov v domácej liečbe sa dá vyvodiť efektívnosť liečby, účinnosť jednotlivých preparátov a predikovať zhoršenie priebehu ochorenia. Synergickým efektom je schopnosť predikovať nutnosť hospitalizácie. Analýza evidencie osôb zúčastňujúcich sa kultúrno spoločenských podujatí môže pomôcť pri identifikácii možných vektorov šírenia nákazy, a môže mať vplyv pri presnejšej definícii ohnísk. Prediktívne mechanizmy môžu vopred určiť väzby medzi zdanlivo nesúvisiacimi informáciami a definovať možné korelácie či hrozby.

Aktuálny stav zberu a vyhodnocovania dát súvisiacich s pandémiou je determinovaný roztrieštenosťou jednotlivých projektov a platforiem, ktoré medzi sebou dáta nezdieľajú alebo je zdieľanie zbytočné pre nekonzistentnú dátovú štruktúru. Víziou projektu je niektoré z absentovaných oblastí pokryť a podporiť úspešný boj s pandémiou, keďže efektívne opatrenia je možné zaviesť len na základe zozbierania kvalitných dát a ich správnej interpretácie a analýzy.

V aktuálnej pandemickej situácii, vidíme stále potrebu a využiteľnosť ucelených riešení, ktoré poskytuje projekt Life Defender – Ochranca života. Našou víziou a naším cieľom je vytvorenie uceleného a komplexného riešenia, ktoré poskytne v spolupráci s partnerom v projekte prepojenú HW a SW platformu. Toto riešenie poskytne možnosť efektívnej prípravy na ochranu pred možnými budúcimi variantmi tohto ochorenia. Zamerali sme sa na čo najefektívnejšie využitie zozbieraných dát a informácií z predchádzajúcej analytickej časti dokumentu a ich najefektívnejšiu implementáciu do jednotlivých SW modelov, ktoré sme vytvorili v tomto míľniku.

Všetky dáta v rámci projektu budú slúžiť nielen pre interné potreby projektu, ale budú poskytované v anonymizovanej podobe pre potreby vedeckého skúmania.

V jednotlivých kapitolách postupne predstavujeme aktuálny stav riešenia pracovných balíkov, pričom softvérová časť riešenia nadväzuje na hardvérovú časť, ktorú má v gescii partner v projekte a ktorú popíše vo vlastnom výstupnom dokumente.

Šablóna pre vkladanie tabuliek a jej popis:

1.1 Členenie dokumentu

Aktivity H1 a H2 sa rozdeľujú na základné pracovné balíky, ktoré zároveň vystupujú ako hlavné kapitoly dokumentu:

- Experimentálny vývoj prototypu SW platformy a cloudového úložiska (pracovný balík (ďalej iba PB) 1.1 a 2.1) - Návrh prototypu SW platformy (spôsob evidencie, zberu dát a komunikácie).
- Experimentálny vývoj prototypu mobilnej aplikácie (PB 1.2, PB 2.2) - Návrh prototypu mobilnej aplikácie.
- Experimentálny vývoj prototypu modulu pokročilej analýzy a vizualizácie dát (PB 1.3, PB 2.3) - Návrh prototypu modulu analýzy a vizualizácie dát.

Výstupmi každého pracovného balíka je v rámci mílnika č. 2 popis a opis vytvorených softvérových platforiem a prototypov aplikácií a pracovných postupov.

Výstupom jednotlivých pracovných balíkov budú tieto prototypy riešenia:

- Prototyp SW platformy pre zber a uchovávanie anonymizovaných dát z integrovaných riešení vytvorený v rámci PB 1.1 a PB 2.1.
- Prototyp mobilnej aplikácie vytvorený v rámci PB 1.2 a PB 2.2.
- Prototyp modulu vizualizácie dát a Prototyp dátového modelu na zbieranie, analýzu a vyhodnocovanie symptómov pomocou analytických nástrojov s využitím umelej inteligencie vytvorený v rámci PB 1.3 a PB 2.3.

2 METODIKA RIADENIA PROJEKTU

2.1 Projektové riadenie

Pre riadenie projektu vo fáze implementácie bola zvolená agilná metóda SCRUM, ktorá má nasledujúce výhody oproti klasickému riadeniu:

1. Výsledky je možné dodávať a prezentovať (prostredníctvom dema) v krátkych časových intervaloch.
2. Väčšia flexibilita.
3. Rýchle možnosti zlepšovania produktu a implementačného procesu na základe častejšej spätnej väzby od teamu alebo zákazníka, a to formou prezentácie funkčného dema a vyhodnocovania jednotlivých vývojových cyklov (sprintov) na tzv. retrospektívnej porade.
4. Lepšia možnosť reagovať na požiadavky pre zmenu (projekt si vyžadoval spoluprácu dvoch firiem pre implementáciu SW a HW vrstvy projektu).

Ako podklad pre plánovanie jednotlivých úloh projektu a jeho transformácia do tzv. epikov a úloh, ktoré riešil implementačný team bola použitá forma WBS (pozri nasledujúci obr.):

Life Defender - domáca karanténa													
Manažment	Technológie	Dátový model			Webový portál				Mobilná aplikácia				Testovanie
Projektový a produktový manažment	Technológie	Dátový model / API	Notifikácie	Profily/Roly	Stránky na webe	Registrácia	Zaevidovanie pacienta	Zber dát	Objednanie zariadenia	Aktivácia zariadenia	Späťvzatie a reset zariadenia	Android & iOS	Testovanie, nasadenie, dokumentácia
Analýzy a zadania	Preskúmanie technológií	API pre mobilnú aplikáciu	Pozvánka pre pacienta (prvotný mail s inštrukciou)	Profil pre pacienta (meno, priezvisko, mail, telefón, kontaktná adresa, ICE kontakt)	Dashboard	Registrácia do systému (lekár, pacient,poskytovateľ zariadenia)	Pozvanie pacienta	Vykonanie merania	Informácie o procese objednania zariadenia a aktivácii	Aktivácia zariadenia	Deaktivácia zariadenia	Prihlásenie	Nasadenie
Projektová koordinácia	Výber technológií	API pre web	Resent pozvánky	Profil pre lekára (kontakt, ordinačné hodiny, poznámkové okno)	Pacienti	Pridelenie a spravovanie oprávnení	Zaevidovanie pacienta	Synchronizácia dát	Objednanie zariadenia	Spáročovanie zariadenia s aplikáciou	Reset nastavení	Úvodná stránka	Testovanie
Projektový kontroling	Vzdelávanie v novej technológii	Evidencia zariadení mobil/tablet	Notifikácia s kritickými meraniami	Profil pre poskytovateľa	Lekári	Spravovanie používateľov (akceptovanie)	Poslanie pozvánky na inštaláciu aplikácie	Zber a vyhodnocovanie dát	zaevidovanie objednávky (zobrazenie na profile)	Prípraviť manuál ako spáročovať zariadenie		Denná kontrola	Dokumentácia
Komunikácia so zákazníkom		Evidencia meracieho zariadenia (check me)	Notifikácia pre ICE kontakt	Profil Administrátor	Pozvánky		Harmonogram meraní	Výsledky				Detail merania	
		Prepojiť DB s portálom a app	Prehľad notifikácií	Životný cyklus zariadenia - profil "poskytovateľ zariadenia"	Zariadenia			Manuálne zbieranie dát				Detail pacienta	
			Spracovanie notifikácií		Dotazníky							Detail lekára	
					Úlohy							Registrácia aplikácií do store-ov	

2.2 Projektový tím

Charakter a štruktúra projektu obsahovala nasledujúce projektové a výskumne roly:

1. Projektové roly:
 - a. Projektový manažér,
 - b. Asistent projektového manažéra,
 - c. Kľúčoví vedecko-výskumní pracovníci,

- d. Zahraničný výskumný pracovník,
 - e. Vedecko-výskumní pracovníci,
 - f. Technickí a iní pomocní pracovníci v rozsahu výskumných aktivít projektu.
2. Roly v rámci fázy implementácie a metodiky SCRUM:
- a. Scrum master,
 - b. Product owner – vyplývajúc z charakteru projektu bol product owner interný,
 - c. Implementačný team:
 - i. Analytici,
 - ii. Programátori:
 1. GUI,
 2. Integrácia,
 3. DB,
 - iii. Testeri.

2.3 Koordinačné stretnutia

Koordinačné stretnutia vyplynuli z použitej metodiky riadenia:

- Koordinačné stretnutia – implementačne:
 - Denne stretnutia implementačného teamu k stavu jednotlivých spracovávaných úloh – riešil sa stav spracovania úloh a prípadne komplikácie pri implementácii.
 - Týždenne koordinačné stretnutia medzi implementačnými teamami pre SW a HW časti projektu.
- Projektové:
 - Mesačný status stretnutia – hodnotenie stavu celého projektu.

3 EXPERIMENTÁLNY VÝVOJ PROTOTYPU SW PLATFORMY A CLOUDOVÉHO ÚLOŽISKA

3.1 Prototyp pre Domácu karanténu

Cieľom vývoja prototypu SW pre podporu domácej karantény bolo vytvorenie nástroja pre lekárov a pacientov, ktorý umožňuje vzdialené monitorovanie pacientov a vyhodnocovanie nazbieraných údajov z jednotného cloudového úložiska. Monitorovanie pacientov prebieha použitím technických prostriedkov alebo prostredníctvom dotazníkov, ktoré slúžia k subjektívnemu zhodnoteniu stavu pacienta. Vývoj prototypu pozostával z nasledujúcich súčastí:

- Prototyp databázy výsledkov testov,
- Prototyp analýzy miery a priebehu testovaných symptómov pri testovanej osobe,
- Prototyp analýz a reportov samostatných symptómov – pandemická mapa.

Technické riešenie SW prototypu:

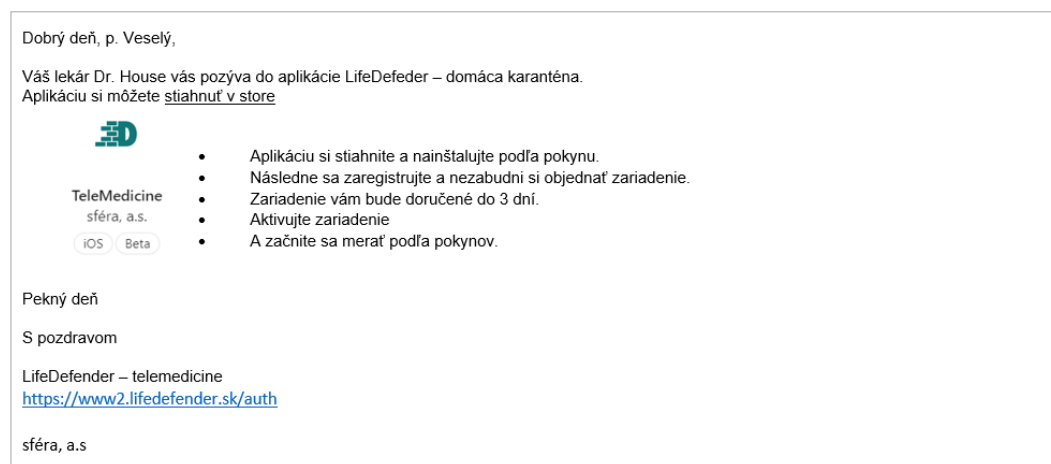
- Web portál - riešenie primárne pre lekára,
- Mobilná aplikácia - riešenie primárne pre pacienta.

3.1.1 Popis používateľských scenárov

V nasledujúcej kapitole je popísaný scenár využitia domácej karantény z pohľadu pacienta a jeho interakcie s monitorovacím zariadením a lekárom.

3.1.1.1 Scenár 1

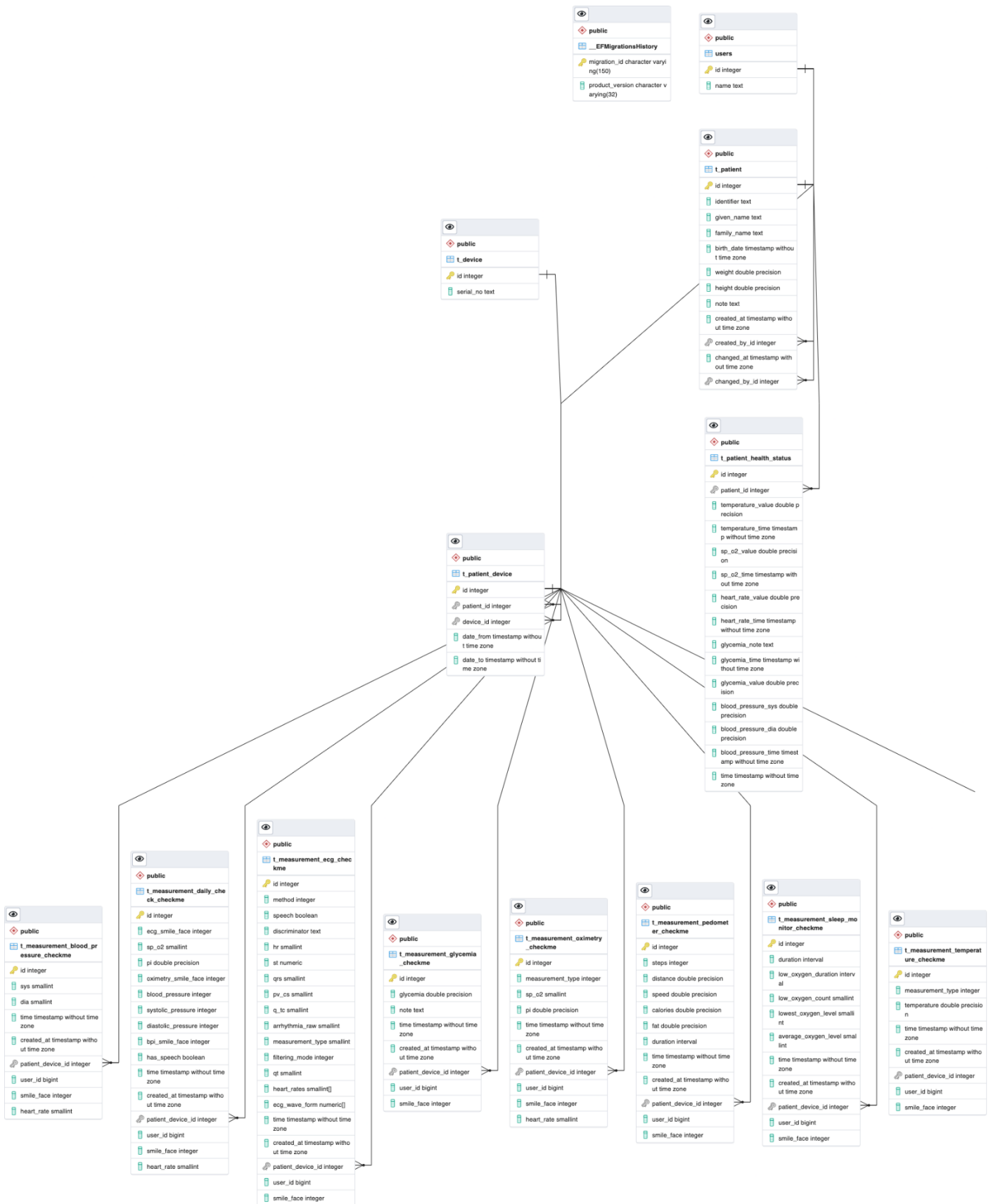
- Pacient zistí, že je pozitívny (kombináciou viacerých faktorov, AG test, PRC test, príznaky),
- Kontaktuje telefonicky/mailom lekára,
- Lekár zaeviduje pacienta na portáli Life Defender (meno, priezvisko, mail),
- Odošle pozvánku pacientovi:
 - Po odoslaní pozvánky sa pacient zobrazí v záložke *Pozvánky*,
 - Statusy:
 - Potvrdená – pacient pozvánku potvrdí v maili a spustil proces,
 - Nepotvrdená – pacient nevykonal žiadnu aktivitu. Lekár v prípade potreby vie opätovne zaslať pozvánku pacientovi.
- Pacient postupuje podľa inštrukcie v maili.



Obrázok 1 Ukážka pozvánky – mail

- Stiahnuť aplikáciu,
- Nainštalovať aplikáciu,
- Zaregistrovať sa,
- Objednať zariadenie,
- Aktivovať zariadenie,
- Meranie a zber dát,
- Pacient si stiahne aplikáciu (Google play/apple store),
- Zaregistruje sa:
 - Zadá: meno, priezvisko, lekára, mail a heslo,
 - Ak je mail rovnaký ako ten, ktorý zadal lekár, prebehne spárovanie – pozvánka dostane status POTVRDENÁ,
 - Ak je mail rozdielny ako ten, ktorý zadal lekár, párovanie neprebehne automaticky, ale zobrazí sa lekárovi v prehľade *Pozvánky* a manuálne akceptuje pacienta.
- Podľa inštrukcie si objedná zariadenie na meranie:
 - Zadá kontaktnú adresu:
 - na túto adresu bude zariadenie doručené,
 - adresa sa automaticky uloží ako kontaktná adresa na profil pacienta.
 - Odošle objednávku – objednávka je spracovaná D+2,
 - Inak je D rátané ako nasledujúci deň,
 - Objednávka sa uloží na profil pacienta.
- Poskytovať zariadenia zaeviduje/spracuje objednávku:
 - Zoznam objednávok vidí na prehľade – zoznam objednávok:
 - Stav/status:
 - Nová – objednávka je nová a bez aktivity,
 - Spracovávaná – priradená na pracovníka u poskytovateľa, ktorý priradí zariadenie na odoslanie,
 - Odoslaná – kuriér prevezme zariadenie/balík od poskytovateľa,
 - Odovzdaná – kuriér odovzdá zariadenie pacientovi.
- Priradí zariadenie objednávateľovi/pacientovi,
- Objedná kuriéra – doručenie podľa inštrukcie v objednávke,
- Pacient priradí zariadenie v aplikácii,
- Vykoná prvé meranie,
- Zobrazia sa namerané údaje v aplikácia,
- Synchronizácia a odoslanie meraní do web portálu,
- Dáta sa zobrazia v pohľade u lekára a sú priradené k pacientovi:
 - Ak sú dáta správne – sú zobrazené,
 - Ak sú niektoré namerané hodnoty neštandardné – zobrazia sa v kritických meraniach.
- Po ukončení karantény používateľ odovzdá zariadenie:
 - Objednaním kuriéra,
 - Odoslaním na adresu, ktorá je uvedená v aplikácii,
 - Odovzdaním u lekára (napr. pri ukončení PN).
- Zariadenie na meranie sa dostane k poskytovateľovi,
- Poskytovateľ zariadenie zresetuje (reset factory),
- Poskytovateľ zariadenie vydezinfikuje,
- Poskytovateľ zariadenie pripraví na opätovné použitie.

3.1.2 Dátový model



Obrázok 2 Dátový model

3.1.3 Webové rozhranie

Webový portál je umiestnený na linke: <https://www2.lifedefender.sk/auth>.

Webový portál je rozdelený vizuálne na 2 časti:

1. Hlavička (menu) – zoznam všetkých modulov.
2. Obsah (telo).

3.1.3.1 Hlavička

Hlavička je rozdelená na tri časti. V ľavej časti je umiestnené logo Life Defender. Následne sú umiestnené moduly, ktoré definujú obsah. V pravej časti sú umiestnené všeobecné funkčnosti.



3.1.3.2 Obsah

Obsah je tvorený z viacerých modulov a zobrazenie daných modulov je riadené oprávneniami (Tabuľka 1 – matica oprávnení).

Pre používanie je nutné, aby používateľ bol prihlásený.

3.1.3.3 Dashboard

Úvodná obrazovka, kam je používateľ presmerovaný priamo po prihlásení. Obsahuje zoznam pacientov, lekárov a pozvánky.

3.1.3.4 Pacienti

Zoznam pacientov je zobrazený podľa preddefinovaných kritérií a každý záznam v riadku je klikateľný. Sú definované 3 skupiny pacientov:

- 1) Všetci pacienti - zobrazujem všetkých pacientov bez ohľadu na hodnotu v zozbieraných dátach, zotriedenie je podľa posledného merania, teda najaktuálnejšie/posledné merania sú hore.
- 2) Kritické merania - zobrazuje zoznam pacientov, ktorých hodnoty boli vyhodnotené ako kritické.
- 3) Noví pacienti - pacienti, ktorí majú status noví.

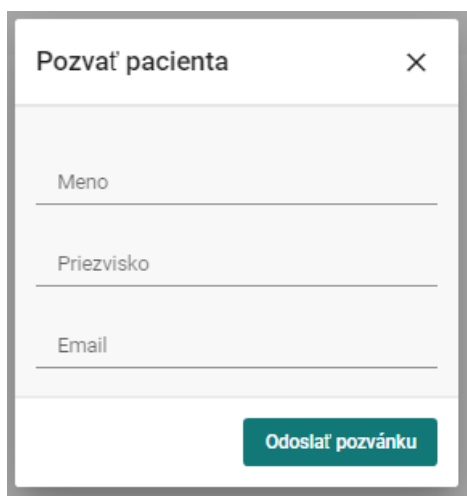
Každý lekár vidí primárne pacientov, ktorých má priradených. Avšak cez vyhľadávanie dokáže lekár vyhľadať akéhokoľvek pacienta a spravovať/editovať jeho dáta.

V pravej časti obrazovky je podporený export dát do .xlsx.

Meno	Posledné meranie	Saturácia	Ľep	Tlak	Teplota
Ján Hraško	20. 9. 2021	99	55	120/80	36.6
Ferko Mrkvička	20. 9. 2021	99	55	120/80	36.6
Miška Líška	20. 9. 2021	99	55	120/80	36.6
Majko Zajko	20. 9. 2021	99	55	120/80	36.6
Lacko Macko	20. 9. 2021	99	55	120/80	36.6
Majka Čajka	20. 9. 2021	99	55	120/80	36.6

Obrázok 3 Zoznam pacientov

V sub menu je podporený procesný tok pre pridanie nového pacienta. Po kliknutí na „+“ sa zobrazí pop-up formulár, kde lekár vyplní základné dáta (meno, priezvisko a mail) – pozri Obr. 5. Následne odošle pozvánku a pacient obdrží mail s inštrukciou (pozri Obr. 4).



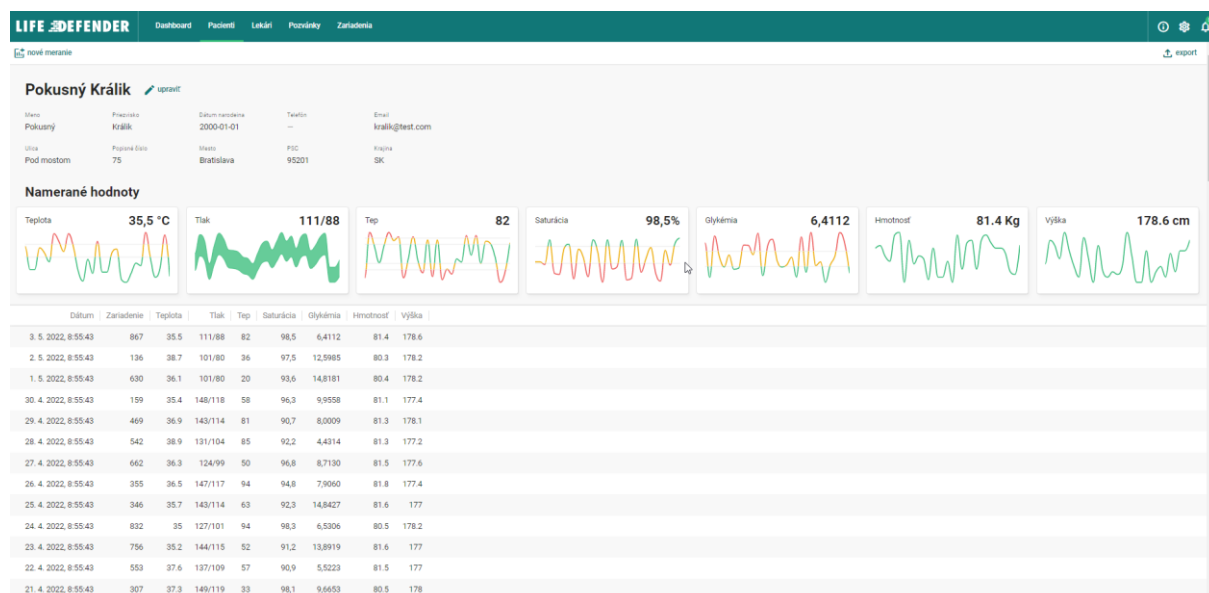
Obrázok 4 Pozvať nového pacienta do aplikácie

Po kliknutí na záznam (meno a priezvisko) sa zobrazí detail pacienta (pozri Obr. 5).

3.1.3.4.1 Detail pacienta

Detail pacienta je rozdelený na 3 základné časti:

1. Základné informácie o pacientovi.
2. Namerané hodnoty – grafické zobrazenie.
3. Namerané hodnoty – tabuľková forma.



Obrázok 5 Detail pacienta

Základné informácie je možné cez ceruzku editovať a následne zmeniť. Zmenené dáta sa následne prenesú do aplikácie do profilu pacienta. V hornej lište je podporený proces pre manuálne zadanie meraní. Lekár viem priamo do portálu zadať hodnoty, ktoré v danom čase namerá.

Zoznam hodnôt na manuálne zadanie:

1. Tlak
2. Teplota
3. Výška
4. Tep
5. Glykémia

3.1.3.5 Lekári

Zobrazuje zoznam všetkých lekárov, zobrazených podľa osobného čísla. Je podporené aj fulltextové vyhľadávanie a aj export do .xlsx.

Meno	Špecializácia	Osobné číslo	Zariadenie	Miesto	Poistovne
Ján Hraško	vseobecny lekar	123001	Nemocnica pre chorych	Bratislava	VŠZP dovera union
Ferko Mrkvička	vseobecny lekar	123002	Nemocnica pre chorych	Bratislava	VŠZP dovera union
Miška Liška	vseobecny lekar	123005	Nemocnica pre chorych	Bratislava	VŠZP dovera union
Majko Zajko	vseobecny lekar	123003	Nemocnica pre chorych	Bratislava	VŠZP dovera union
Lacko Macko	vseobecny lekar	123004	Nemocnica pre chorych	Bratislava	VŠZP dovera union
Majka Čajka	vseobecny lekar	123005	Nemocnica pre chorych	Bratislava	VŠZP dovera union

Obrázok 6 Zoznam lekárov

V sub menu je podporený procesný tok pre pridanie nového lekára. Po kliknutí na „+“ sa zobrazí pop-up formulár, kde lekár vyplní základné dáta (meno, priezvisko, špecializácia, osobné číslo, miesto pôsobenia (pozri Obr. 7)). Následne je pop-up odoslaný adminovi na schválenie.

Pridať lekára

Meno

Priezvisko

Špecializácia

Osobné číslo

Miesto

Poistovne

VŠZP

Dôvera

Union

Pridať

Obrázok 7 Pridanie nového lekára do portálu

3.1.3.5.1 Detail lekára

Po kliknutí na záznam (meno a priezvisko) sa zobrazí detail lekára (pozri Obr. 8). Detail lekára je rozdelený na 3 základné časti:

1. Základné informácie o lekárovi.
2. Zoznam zazmluvnených poisťovní.
3. Špecializácie.

LIFE DEFENDER Dashboard Pacienti **Lekári** Pozvánky Zariadenia

Dr. House [upraviť](#)

Meno	Priezvisko	Dátum narodenia	Telefón	Email
Dr.	House	—	+421900123432	house@sfera.sk
Ulica	Popisné číslo	Mesto	PSČ	Krajina
Nemocničná	432	Bratislava	90021	SK

Zmluvné poisťovne

- VŠZP
- Dôvera
- Union

Špecializácie

- abdominalna ultrasonografia u dospelých
- algeziológia
- andrológia (sexuológia v urológii)

Obrázok 8 Detail lekára

3.1.3.6 Pozvánky

Zobrazuje zoznam všetkých pozvánok, ktoré lekár odoslal. Zoznam je zotriedený od najstarších po najnovšie. V prípade ak pozvánka nebola akceptovaná, teda pacient si nestiahol aplikáciu, nenainštaloval, tak je možné opätovne poslať pozvánku pre pacienta. Lekár vidí len pozvánky pre pacientov, ktorých pozval on sám. Je podporené aj fulltextové vyhľadávanie, v tom prípade dokáže používateľ vyhľadať akéhokoľvek pacienta, ktorému bola odoslaná pozvánka (aj iným lekárom).

Je podporený ja export do .xlsx.

LIFE DEFENDER Dashboard Pacienti Lekári **Pozvánky** Zariadenia

+ pozvať pacienta [export](#)

Pozvánky 🔍

Meno	Dátum pozvania	Status
Ján Hraško	20. 9. 2021	nepotvrdená ➡
Ferko Mrkvička	17. 9. 2021	potvrdená ✓
Miška Liška	16. 9. 2021	nepotvrdená ➡
Majko Zajko	12. 9. 2021	potvrdená ✓
Lacko Macko	7. 9. 2021	potvrdená ✓
Majka Čajka	28. 8. 2021	nepotvrdená ➡

Obrázok 9 Zoznam pozvánok

3.1.3.7 Zoznam zariadení

Tento modul je prístupný len pre poskytovateľov služieb. V tomto module sú zobrazené všetky zariadenia, ktoré poskytovateľ má priradené a zaevidované.

Id	Typ	Status	Posledná synchronizácia
1	deviceType1	active	27. 4. 2022, 5:20:00
2	deviceType1	active	27. 4. 2022, 6:30:00
3	deviceType2	inactive	–
4	deviceType2	active	27. 4. 2022, 9:00:00
5	deviceType1	active	27. 4. 2022, 10:00:00

Obrázok 10 Zoznam zariadení

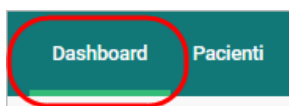
Cez tlačidlo „+“ pridat' zariadenie je podporené pridat' nové zariadenie, na pridanie je nutné vyplniť typ a ID (pozri Obr. 11).

Obrázok 11 Pridat' zariadenie

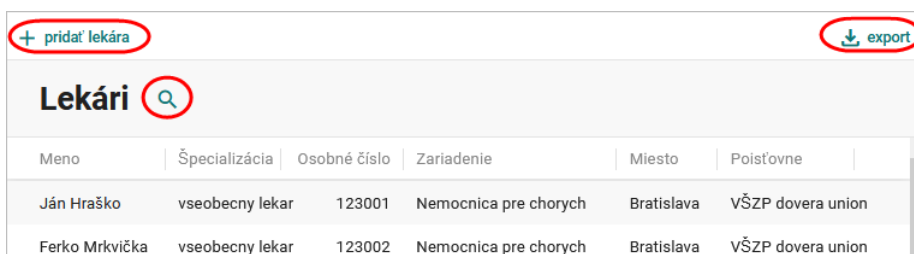
3.1.3.8 Ovládacie prvky

Konvencie ovládacích prvkov:

- **Aktuálna záložka** – názov aktuálne zobrazenej záložky formulára je uvedený bielym textom a zelenou čiarou.



- **Tlačidlá** – slúžia na vykonanie požadovanej funkčnosti. Pri umiestnení ukazovateľa myši na tlačidlo sa farba plochy tlačidla zmení z bledej na tmavú. Kliknutím na tlačidlo sa spustí činnosť.




- **Editovacie polia** - sú polia, do ktorých je možné vkladať údaje. Pri aktívnom zapisovaní údajov je pole označené zelenou čiarou.

Pozvať pacienta ✕

Meno

Priezvisko

-  **Zmena šírky stĺpca** – v prípade, že v hlavičke zoznamu nie je zobrazený celý názov stĺpca, používateľ si ho môže zobrazíť pohybom zvislej čiary oddeľujúcej názvy jednotlivých stĺpcov hlavičky. Podržaním ukazovateľa myši na zvislom oddeľovači stĺpcov sa zobrazí grafický znak, posunom ktorého je možné upraviť šírku stĺpca na požadovanú veľkosť.

Lekári 🔍

Meno	Špecializácia	Osobné číslo	Príariadenie	Miesto	Poist'ovne
Ján Hraško	všeobecný lekár	123001	Nemocnica pre chorých	Bratislava	VŠZP dôvera union
Ferko Mrkvička	všeobecný lekár	123002	Nemocnica pre chorých	Bratislava	VŠZP dôvera union

- **Zaškrtávacie pole** - umožňuje zapínať/vypínať voľby . Vlastnosti sa zapínajú alebo vypínajú kliknutím na zaškrtávacie pole .


Zmluvné poisťovne


VŠZP

Dôvera

Union

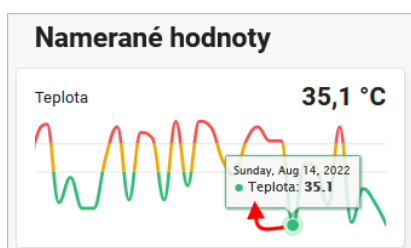
- **Tooltip** alebo **Pomocník v bubline** - priblížením kurzora myši nad tlačidlo alebo údajovú položku sa v bubline zobrazí vlastný informatívny, resp. upozorňujúci text.

nepotvrdená 

potvrdená 

Preposlať pozvánku

- V záložke **Pacienti** systém zobrazuje hodnoty v grafickom a v tabuľkovom zobrazení, kde priblížením ukazovateľa myši na bod grafu je možné sledovať zaznamenané údaje definovaných veličín v rôznych časových bodoch. Pohybom myši po ploche grafu sa zobrazia informácie s uvedením dátumu, názvami sledovaných veličín a hodnotami nameraných údajov. Príklad uvedený v grafe zobrazuje namerané hodnoty *Teploty*.



3.1.3.9 Rozmiestnenie portálu

Prvá stránka, ktorá sa otvorí používateľovi po prihlásení, je *Dashboard* určený na zobrazenie prehľadu určených údajov. Pri väčšom rozlíšení obrazovky sa portál dynamicky prispôsobí maximálnej šírke a výške prehliadača. Štandardná stránka portálu je členená na logické časti:

LIFE DEFENDER Dashboard Pacienti Lekári Pozvánky Zariadenia

Dashboard

Pacienti Lekári Pozvánky

+ pridat lekára export

Lekári 🔍

Meno	Špecializácia	Osobné číslo	Zariadenie	Miesto	Poisťovne
Ján Hraškovcebný lekár		123001	Nemocnica pre chorých	Bratislava	VŠZP dovera union
Ferko Mrkvica		123002	Nemocnica pre chorých	Bratislava	VŠZP dovera union
Miška Liškovec		123005	Nemocnica pre chorých	Bratislava	VŠZP dovera union
Majko Zajko		123003	Nemocnica pre chorých	Bratislava	VŠZP dovera union
Lacko Macko		123004	Nemocnica pre chorých	Bratislava	VŠZP dovera union

Dr. House [upraviť](#)

Meno	Priezvisko	Dátum narodenia	Telefón	Email
Dr.	House	—	+421900123432	house@sfera.sk
Ulica	Popísané číslo	Mesto	PSČ	Krajina
Nemocničná	432	Bratislava	90021	SK

Zmluvné poisťovne

VŠZP
 Dôvera
 Union

Špecializácie

- abdominálna ultrasonografia u dospelých
- algeziológia
- andrológia (sexuológia v urológii)

1 **Hlavička portálu** - horná lišta úvodného okna, ktorá obsahuje funkcionlitu:

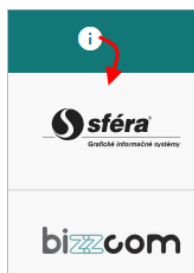
- Logo a názov systému **Life Defender**, umožňuje návrat na zobrazenie úvodného okna/dashboardu z ľubovoľnej stránky portálu alebo inej časti portálu.




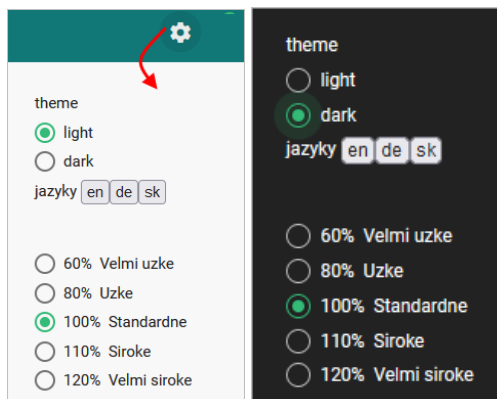
- **Navigačná lišta** poskytuje používateľovi základné funkcie jednotlivých stránok portálu v záložkách s príslušným názvom záložky. Bližší opis záložiek sa nachádza v kapitolách [Pacienti](#), [Lekári](#), [Pozvánky](#), [Zoznam zariadení](#).





- - kliknutím na ikonu, systém zobrazí informatívne okno s vlastníkom autorských práv spoločnosti sféra, a.s. a Bizzcom s.r.o.

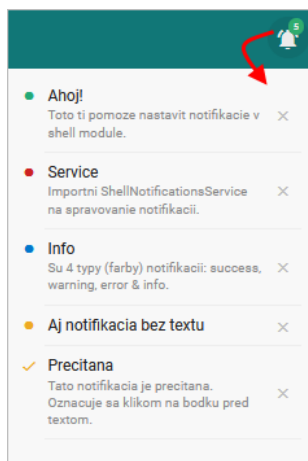


-  **Nastavenia** (možnosti) - po stlačení ikony systém zobrazí dialógové okno *Nastavenia*, v ktorom je možné zmeniť hodnoty položiek.



- Možnosti nastavenia podkladu okna - nastavenie inverzného zobrazenia farieb na tmavom podklade.
- Výber jazykovej mutácie - (EN – Anglicky, DE – Nemecky, SK - Slovensky). Štandardne je predvolená slovenská verzia (SK).
- Zobrazenie stránky (záložky) sa líši v závislosti od zväčšenia alebo zmenšenia celého obsahu okna.

-  - kliknutím na zvonček systém otvorí panel notifikácií. Číslo v zelenom políčku  zobrazuje počet nevybavených notifikácií. Už stiahnuté súbory je možné nájsť v adresári nastavenom vo webovom prehliadači.




2 **Obsahová časť** – je centrum portálu, kde sa zobrazujú požadované údaje jednotlivých stránok (záložiek) portálu, príslušné funkčnosti, informácie a údaje aktívnej stránky.

3 **Tabuľka/Zoznam** - kliknutím na danú záložku (príklad **Lekári**) systém zobrazí tabuľku (zoznam) s existujúcimi údajmi v režime prezerania, resp. pridania nového záznamu.

+ pridať lekára ↓ export						
Lekári <input type="text"/>						
Meno	Špecializácia	Osobné číslo	Zariadenie	Miesto	Poist'ovne	
Ján Hraško	vseobecny lekar	123001	Nemocnica pre chorych	Bratislava	VŠZP dovera union	
Ferko Mrkvička	vseobecny lekar	123002	Nemocnica pre chorych	Bratislava	VŠZP dovera union	
Miška Líška	vseobecny lekar	123005	Nemocnica pre chorych	Bratislava	VŠZP dovera union	
Majko Zajko	vseobecny lekar	123003	Nemocnica pre chorych	Bratislava	VŠZP dovera union	

K dispozícii je funkcionálnosť:

-  - stlačením tlačidla systém zobrazí formulár napr. **Pridať lekára** v režime pridania nového záznamu.

Pridať lekára ×

Meno

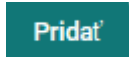
Priezvisko


Špecializácia


Osobné číslo

Pridať

V následne otvorenom formulári je potrebné vyplniť požadované hodnoty a potvrdiť tlačidlom



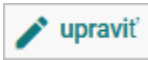
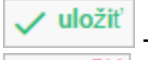
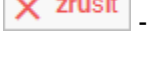
-  - obsah záznamov je možné exportovať do súboru formátu XLSX. Štandardný výstup zo systému je podporený na všetkých zmysluplných obrazovkách. Export rieši otázku exportu dát zo systému. Jeho snahou nie je riešiť formátovanie výstupov a pod., to zostáva na používateľovi a vhodnej aplikácii, štandardne MS Excel. Je potrebné upozorniť, že export zohľadňuje zobrazený zoznam záznamov, t. j. množstvo načítaných riadkov, nastavenie filtra, zoradenie podľa stĺpca aj poradie stĺpcov. Súčasťou exportovaných údajov sú názvy stĺpcov, termín vytvorenia exportu a nastavený filter.

-  - kliknutím na ikonu systém ponúkne vyhľadávacie (editovateľné pole), používateľ môže zadať hľadaný text alebo len jeho časť, následne systém ponúkne na výber napr. skupinu lekárov.

4 V tabuľke kliknutím na zvolený záznam systém zobrazí formulár v režime prezerania či modifikácie.

The image shows a user profile form for 'Dr. House'. The form includes fields for name, address, phone, and email. A red circle highlights the 'upraviť' (edit) button. A modal window is open, showing the same profile with a red circle around the address fields and a red box labeled 'editovatelné položky' (editable items).

Vo formulári má používateľ možnosť volať dostupné operácie umiestnené nad záznamami:

-  - tlačidlo slúži na modifikáciu údajov existujúcich záznamov v systéme.
-  - tlačidlom sa uloží nový záznam.
-  - tlačidlom sa zruší vytvorenie nového záznamu.

3.1.4 Architektúra systému

Informačný systém pre domácu karanténu sa skladá z dvoch kľúčových častí:

1. Cloudová služba,
2. Mobilná aplikácia.

Cloudová služba je budovaná ako platforma pre ukladanie nameraných údajov, ktoré môžu byť získané z rôznych zdrojov. Cloudová služba má rozhranie pre používateľov prostredníctvom webového rozhrania a rozhranie pre automatizovanú komunikáciu prostredníctvom Rest API – určené pre synchronizáciu s mobilnou aplikáciou.

Cloudová služba je vytvorená pomocou nasledovných technológií:

- Angular,
- Microsoft .NET Core,
- PostgreSQL,
- Docker.

Angular ako framework pre tvorbu webového používateľského rozhrania, bol použitý na vytvorenie samostatnej UI vrstvy. Táto UI vrstva komunikuje prostredníctvom Rest API služieb s biznis modelom, ktorý poskytuje operácie a dáta. Rest API služby sú vytvorené pomocou frameworku Microsoft .NET Core s využitím ORM (Object Relational Mapping) Entity Framework Core. Ako relačná databáza pre ukladanie perzistentných dát bola zvolená PostgreSQL databáza. Jednotlivé komponenty informačného systému sú publikované prostredníctvom technológie Docker.

Tabuľka 1 Zoznam endpointov systému

Názov	Endpoint
Používateľské rozhranie	https://app.lifedefender.sk
API endpointy	URL REST API: https://api2.lifedefender.sk <ul style="list-style-type: none"> • Skupiny endpointov jednotlivých logických celkov: <ul style="list-style-type: none"> ○ /authentication/.. (prihlasovanie do systému)

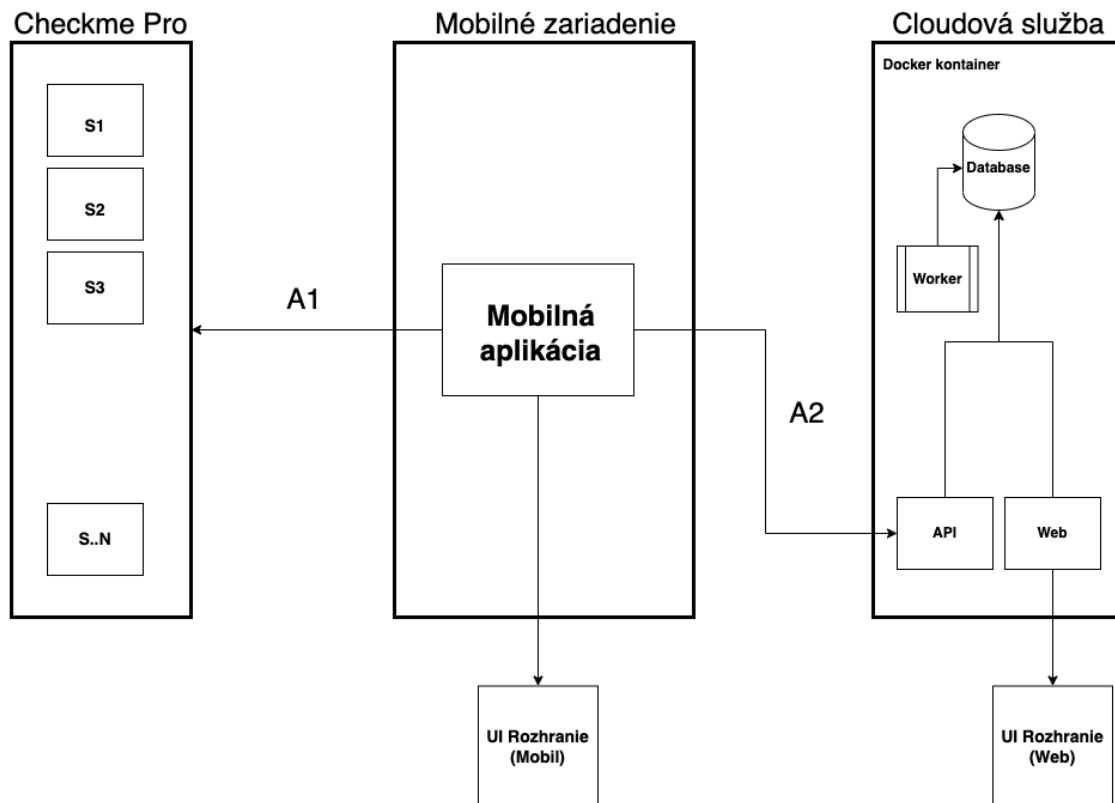
Názov	Endpoint
	<ul style="list-style-type: none"> ○ /patients/.. (správa pacientov) ○ /measurements/.. (vytváranie a prezeranie meraní) ○ /doctors/.. (správa lekárov) ○ /devices/.. (správa meracích zariadení (Checkme)) ○ /orders/.. (objednávanie zariadení Checkme) ○ /survey/.. (dotazníky stavu pacienta) ○ /installation/.. (aktivácia mobilnej aplikácie)

Mobilná aplikácia slúži ako nástroj pre získanie a odoslanie nameraných dát do cloudovej služby. Dáta získané z meracieho zariadenia ([Checkme Pro](#)) sa uložia do internej databázy aplikácie následne synchronizujú do cloudovej služby. Pre tvorbu mobilnej aplikácie bol zvolený framework [Xamarin](#), ktorého výhodou je, že umožňuje súčasnú tvorbu mobilných aplikácií pre obe platformy Android a iOS.

Všetky zvolené technológie a frameworky sú open source.

3.1.4.1 Logická architektúra systému

Na nasledujúcom obrázku je vidieť základnú logickú architektúru informačného systému pre domácu karanténu.



Obrázok 12 Logická architektúra systému

Mobilná aplikácia prostredníctvom technológie bluetooth (integrácia A1 na obrázku) synchronizuje dáta zo zariadenia Checkme Pro do svojej internej databázy, následne tieto získané dáta synchronizuje prostredníctvom Rest API služieb (integrácia A2 na obrázku) do cloudovej služby. Mobilná aplikácia poskytuje používateľské rozhranie pre ovládanie synchronizácie dát. Cloudová služba sprístupňuje namerané dáta prostredníctvom webového rozhrania. Dáta, v cloudovej službe sú ukladané do relačnej databázy. Cloudová služba obsahuje komponentu worker, ktorý reprezentuje sadu aplikačných služieb pre vykonávanie asynchrónnych operácií.

3.1.4.1.1 Typy integrácií

Systém obsahuje dve kľúčové integrácie, a to integráciu medzi mobilnou aplikáciou a cloudovou službou a medzi mobilnou aplikáciou a meracím zariadením Checkme Pro.

Tabuľka 2 Zoznam integrácií

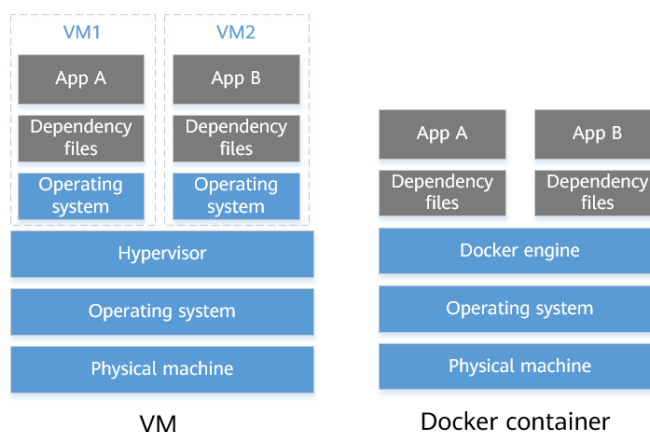
Integrácia	Systém	Popis
A1	Mobilná ap. -> Checkme Pro	Bluetooth slúži na synchronizáciu
A2	Mobilná ap. -> cloudová služba	HTTP Rest API slúži na synchronizáciu

3.1.4.1.2 Komponenty systému

- **Web** - Komponent, ktorý zabezpečuje používateľské rozhranie prostredníctvom webového rozhrania.
- **API** - Komponent, ktorý zabezpečuje REST API služby pre synchronizáciu dát medzi mobilnou aplikáciou a cloudovou službou, a medzi webovou aplikáciou a cloudovou službou.
- **Worker** - Komponent, ktorý zabezpečuje vykonávanie plánovaných úloh.
- **WEB:pgAdmin** - Komponent, ktorý zabezpečuje používateľské rozhranie pre prístup do databázy PostgreSQL.
- **DB:PostgreSQL** - Komponent, ktorý zabezpečuje perzistentné úložisko dát – relačná databáza.
- **Kibana** - Komponent, ktorý slúži na vizualizáciu dát v databáze Elasticsearch.
- **Elasticsearch** - Komponent, ktorý poskytuje úložisko pre dáta vo formáte a štruktúre vhodnej na dátové analýzy.

3.1.4.2 Fyzická architektúra systému

Cloudová služba pre domácu karanténu je vytvorená tak, aby sa dala prevádzkovať v docker kontajneroch (kontajnerizácia). V dnešnej dobe je to najmodernejší spôsob prevádzkovania informačných systémov. V rámci tohto projektu bol jeden z cieľov vyskúšať tento spôsob, aby sme využili jeho výhody voči štandardnej prevádzke vo virtuálnych serveroch (virtualizácia).



Obrázok 13 Docker kontajner vs Virtuálne servery (VM).

Na obrázku je vidieť porovnanie dvoch najpoužívanejších možností prevádzkovania informačných systémov, a to:

- VM - Virtualizácia,
- Docker container – Kontajnerizácia.

Virtualizácia – je technológia umožňujúca prevádzkovať viac nezávislých virtuálnych serverov na jednom fyzickom serveri. Prevádzka informačných systémov vo virtuálnom prostredí (voči prevádzke na fyzickom prostredí) je výhodná hlavne z bezpečnostného a ekonomického dôvodu.

Kontajnerizácia – je virtualizácia za použitia menšieho množstva systémových zdrojov. Aplikácie sú prevádzkované v dockeroch. Docker obsahuje minimálne systémové prostriedky potrebné pre prevádzku aplikácie. Prevádzka pomocou docker kontajnerov je výhodná hlavne z pohľadu náročnosti nasadenia aplikácií, závislosti aplikácií na operačnom systéme a aktualizácií aplikácií a prostredia.

Tabuľka 3 Zoznam kontajnerov

Názov kontajnera	Image	
ld-web	nginx:latest	Kontajner pre webový server
ld-api	mcr.microsoft.com/dotnet/aspnet:5.0	Kontajner pre API služby
ld-worker	mcr.microsoft.com/dotnet/runtime:6.0	Kontajner pre aplikačný server
ld-pgadmin	dpage/pgadmin4:6	Kontajner pre webové rozhrania databázového servera
ld-postgres	postgres:13.4	Kontajner pre databázový server
ld-kibana	kibana:7.14.0	Kontajner pre webový nástroj Kibana
ld-elasticsearch	elasticsearch:7.14.0	Kontajner pre databázu ElasticSearch

3.1.4.2.1 Prostredia

Cloudová služba pre domácu karanténu je prevádzkovaná v jednom vývojovom prostredí na infraštruktúre spoločnosti sféra, a.s.

Tabuľka 4 Zoznam prostredí

Názov	Doména	Infraštruktúra
Vývojové prostredie	lifedefender.sk	HW spoločnosti sféra, a.s.

3.1.4.2.2 Bezpečnosť

Bezpečnosť systému je riešená na úrovni infraštruktúry, ale aj na úrovni aplikačnej vrstvy. Infraštruktúra, do ktorej je systém inštalovaný, je budovaný s ohľadom na vysokú dostupnosť a s požiadavkou na minimalizáciu rizika straty dát. Všetky používateľské aj automatizované rozhrania systému sú zabezpečené voči neautorizovanému vstupu, a sú publikované ako zabezpečené prostredníctvom SSL protokolu.

3.1.4.2.3 Autentifikácia

Autentifikácia je proces identifikácie a overenia identity používateľa pri vstupe do informačného systému. V systéme pre domácu karanténu je autentifikácia používateľov zabezpečená pomocou autentifikačného modulu, ktorý používa štandardy OpenID Connect/OAuth 2.0. Voči tomuto autentifikačnému modulu sa budú overovať používatelia pri vstupe do všetkých rozhraní systému. Autentifikácia je jednofaktorová s použitím prihlasovacieho mena a hesla.

Pri vstupe používateľa do systému autentifikačný modul overí zadané meno a heslo používateľa. Po úspešnom overení autorizačný modul prideli oprávnenia používateľovi na základe jeho používateľských rolí, v opačnom prípade autentifikačný modul zabráni používateľovi vstup do systému. Ak je používateľ úspešne overený, môže pristupovať k jednotlivým častiam systému na základe pridelených používateľských rolí.

Počas autentifikácie používateľ systému získa autentifikačný token, ktorý má preddefinovanú životnosť. Po expirácii autentifikačného tokenu (dĺžka expirácie je konfigurovateľná, jej dĺžka závisí od bezpečnostných požiadaviek aplikácie) sa používateľ musí opätovne prihlásiť.

3.1.4.2.4 Autorizácia

Autorizácia je proces riadenia prístupu používateľa k funkčnostiam a dátam informačného systému. V systéme pre domácu karanténu bude proces riadenia prístupov založený na rolách RBAC (Role-based access control), čo znamená, že používateľ získa oprávnenia prostredníctvom používateľskej roly. Administrátor pre správu používateľov bude mať možnosť spravovať používateľské roly (katalóg rolí) a pridelovať ich používateľom.

3.2 Prototyp pre Automatickú testovaciu bunku

Cieľom vývoja SW prototypu pre automatickú testovaciu bunku bolo vyvinúť riešenie, ktoré ukladá a poskytuje dáta o otestovaných subjektoch do jednotného cloudového úložiska dát, a taktiež poskytnúť ovládacie prostredie pre dotykové obrazovky na automatickej testovacej bunke. Dôraz GUI pre automatickú testovaciu bunku bol kladený na jednoduchosť používateľského prostredia, čo najväčšiu možnú mieru automatizácie a urýchlenia procesov, a taktiež boli využité pokročilé SW prvky pre autentifikáciu testovanej osoby voči poskytnutému dokladu totožnosti. Vývoj prototypu SW pre automatickú testovaciu bunku pozostával z nasledujúcich súčastí:

- Preskúmanie možnosti implementácie overenia totožnosti testovaného subjektu oproti dokladu totožnosti cez technológiu spoločnosti Inovatics.
- Prototyp SW ovládacích prvkov pre automatickú testovaciu bunku.

Technické riešenie SW prototypu:

- Dotykové ovládanie automatickej testovacej bunky prostredníctvom webového rozhrania upraveného pre používanie na veľkom dotykovom paneli.

3.2.1 Popis používateľských scenárov

V nasledujúcej kapitole je opísaný prípad využitia automatickej testovacej bunky z pohľadu testovaného subjektu.

3.2.1.1 Scenár 1

1. Osoba, ktorá sa potrebuje otestovať na automatickej testovacej bunke musí byť zaregistrovaná (v SW Life Defender budú spracovávané osobné údaje subjektov a z toho dôvodu je uvažované s implementáciou životného cyklu pre správu osobných údajov na základe legislatívy GDPR), registráciu môže vykonať spôsobom:
 - a. Registrácia priamo na dotykovvej obrazovke testovacej bunky.
 - b. Z dôvodu, že testované osoby môžu byť aj nepnoleté, resp. bez dokladu totožnosti, je možné takéto osoby otestovať taktiež, ale musia byť zaregistrované samostatne pod profilom osoby, ktorú je možné overiť voči dokladu totožnosti.
2. Monitor 1 - po zaregistrovaní sa, testovaná osoba sa prihlási do systému pomocou dotykovvej obrazovky na testovacej bunke, na Monitore 1.
3. Monitor 1 - systém overí totožnosť testovanej osoby voči poskytnutému dokladu, ktorý je naskenovaný a fotka z dokladu je použitia pri porovnaní oproti reálnej fotke zhotovenej počas testu (pozri obrázky nižšie).

Overenie 2 Naskenovanie ID 3 Potvrdenie údajov 4 Fotka tváre

Položte váš identifikačný dokument
na scanovacie zariadenie

Predná strana



Naskenovať

Zadná strana



Naskenovať

Overenie 2 Naskenovanie ID 3 Potvrdenie údajov 4 Fotka tváre

Zarovnaite vasu tvár s rámom



Odfotiť

Pokračovať

4. Monitor 1 - po úspešnom overení je testovaná osoba usmernená pomocou pokynov na obrazovke, aby prešla k vedľajšiemu monitoru, kde je možné vykonať test.
5. Monitor 2 – testovaná osoba je pomocou jednoduchého wizaru navigovaná, akým spôsobom vykonať jednotlivé úkony potrebné k úspešnej realizácii testu.



6. Po úspešnom vykonaní testu bude výsledkoch zaslaný do 15 min. na emailovú adresu poskytnutú pri registrácii.

3.2.1.2 Scenár 2

1. Operátor automatickej testovacej bunky má prístup aj k samostatnému módu pre servis zariadenia.
2. V servisnom móde môže operátor zariadenia spustiť test (diagnostiku zariadenia), reštartovať alebo vypnúť celé zariadenie.

Servisný mód - samotestovacia jednotka

Stav prístroja	
Chyba	<input type="button" value="Spustiť test"/>
Režim stroja	
Automat	<input type="button" value="Zmeniť režim"/>
Počet OK kusov	
164	<input type="button" value="Resetovať"/>
Počet NOK kusov	
25	<input type="button" value="Vypnúť"/>
Počet neplatných testov	
43	

3.2.2 Webové rozhranie

- Prototyp SW pre automatickú testovaciu bunku:
 - Monitor 1 – slúži na registráciu a autentifikáciu testovaného subjektu:
 - <https://touch1.lifedefender.sk/home>
 - Monitor 2 - slúži ako návod pre vykonanie samotného testu:
 - <https://touch2.lifedefender.sk/home>
 - Samostatný servisný mód:
 - <https://touch1.lifedefender.sk/service>

3.2.2.1 Ovládacie prvky

Konvencie ovládacích prvkov:

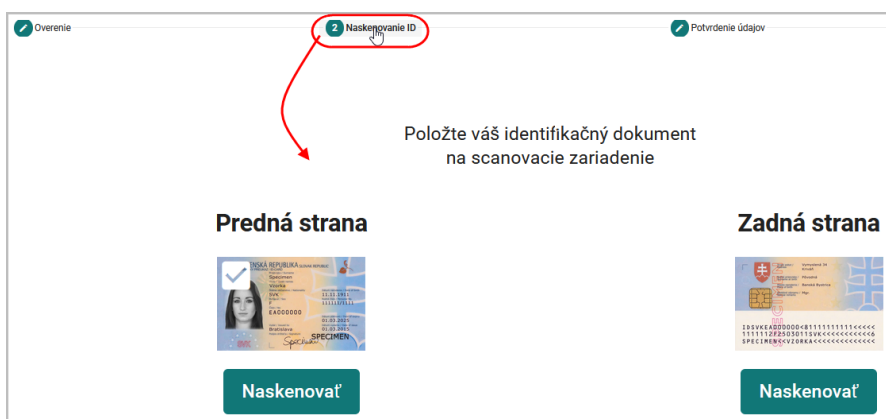
- **Tlačidlá** – slúžia na vykonanie požadovanej funkčnosti. Pri umiestnení ukazovateľa myši na tlačidlo sa farba plochy tlačidla zmení z bledej na tmavú. Kliknutím na tlačidlo sa spustí činnosť.



- **Editovacie polia** - sú polia, do ktorých je možné vkladať údaje. Pri aktívnom zapisovaní údajov je pole označené zelenou farbou.

- **Zaškrťavacie pole** - umožňuje zapínať/vypínať voľby . Vlastnosti sa zapínajú alebo vypínajú kliknutím na zaškrťavacie pole .

- **Vrátenie späť, opätovné vykonanie alebo zopakovanie akcie** - v aplikácii môžete veľa akcií vrátiť späť, opakovať.



3.2.3 Architektúra systému

Informačný systém pre automatickú testovaciu bunku sa skladá z dvoch kľúčových častí:

1. Cloudová služba.
2. Systém pre ovládanie testovacej bunky.

Cloudová služba je budovaná ako platforma pre ukladanie nameraných údajov, ktoré môžu byť získané z rôznych zdrojov. Cloudová služba má rozhranie pre používateľov prostredníctvom webového rozhrania a rozhranie pre automatizovanú komunikáciu prostredníctvom Rest API – určené pre synchronizáciu so systémom pre ovládanie testovacej bunky.

Cloudová služba je vytvorená pomocou nasledovných technológií:

- Angular,
- Microsoft .NET Core,

- PostgreSQL,
- Docker.

Angular ako framework pre tvorbu webového používateľské rozhrania, bol použitý na vytvorenie samostatnej UI vrstvy. Táto UI vrstva komunikuje prostredníctvom Rest API služieb s biznis modelom, ktorý poskytuje operácie a dáta. Rest API služby sú vytvorené pomocou frameworku Microsoft .NET Core s využitím ORM (Object Relational Mapping) Entity Framework Core. Ako relačná databáza pre ukladanie perzistentných dát bola zvolená PostgreSQL databáza. Jednotlivé komponenty informačného systému sú publikované prostredníctvom technológie Docker.

Tabuľka 5 Zoznam endpointov systému

Názov	Endpoint
Používateľské rozhranie	https://app.lifedefender.sk
API endpointy	https://api2.lifedefender.sk

Systém pre ovládanie testovacej bunky je vnorený systém prevádzkovaný priamo v automatickej testovacej bunke. Primárnou úlohou systému je používateľské rozhrania pre ovládanie automatickej testovacej bunky a synchronizácia nameraných dát do cloudovej služby.

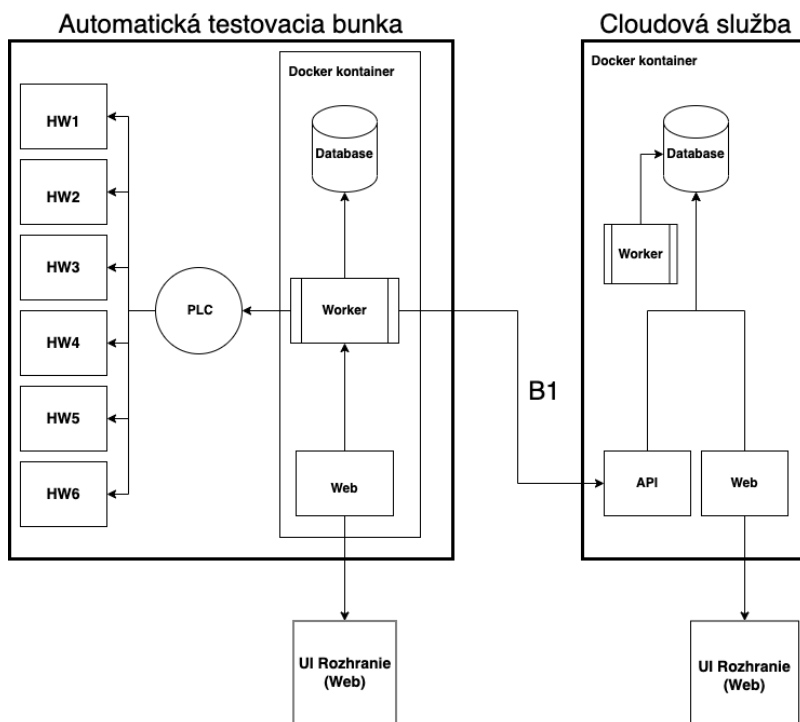
Cloudová služba je vytvorená pomocou nasledovných technológií:

- Angular,
- Microsoft .NET Core,
- PostgreSQL,
- Docker.

Všetky zvolené technológie a frameworky sú open source.

3.2.3.1 Logická architektúra systému

Na nasledujúcom obrázku je vidieť základnú logickú architektúru systému.



Obrázok 14 Logická architektúra systému

Cloudová služba sa skladá z nasledujúcich logických komponentov:

- Web,
- API,
- Worker,
- Databáza.

Cloudová služba poskytuje Rest API rozhranie pre synchronizáciu dát z automatickej testovacej bunky. Synchronizované dáta sa uložia do relačnej databázy. Dáta sú sprístupnené pre používateľov pomocou cloudovej služby prostredníctvom webového rozhrania. Cloudová služba obsahuje aplikačný server, ktorý vykonáva asynchrone úlohy, ako je notifikovanie používateľov alebo príprava dát pre export.

(Vnorený systém) Softvérová časť automatickej testovacej bunky sa skladá z nasledujúcich logických komponentov:

- Web,
- Worker,
- Databáza.

Vnorený systém pre ovládanie testovacej bunky je prevádzkovaný v docker kontajneroch na internom serveri testovacej bunky. Vnorený systém obsahuje tri základné komponenty: web, worker a databáza. Web sprístupňuje používateľské rozhranie na ovládanie automatizovanej testovacej bunky. Webové rozhranie je sprístupnené prostredníctvom dotykového displeja, na ovládanie nie je potrebná klávesnica. Webové rozhranie komunikuje prostredníctvom interného API rozhrania s aplikačným serverom (worker) a databázou. Worker predstavuje sadu aplikačných služieb, ktoré slúžia na ovládanie hardvérovej časti testovacej bunky prostredníctvom PLC rozhrania. Jedna zo služieb aplikačnej časti je služba DIS, ktorej úlohou je autentifikácia používateľa prostredníctvom získaných biometrických informácií. Ďalšou úlohou je synchronizácia spracovaných dát do cloudovej služby (napr. informácie o priebehu a výsledku testu). Databáza slúži ako dočasné úložisko pre dáta, ktoré ešte neboli zosynchronizované do cloudovej služby.

3.2.3.1.1 Typy integrácií

Primárna integrácia je medzi automatickou testovacou bunkou a cloudovou službou. Pomocou tejto integrácie sa synchronizujú nameraná dáta smerom do cloudovej služby.

Tabuľka 6 Zoznam integrácií

Integrácia	Systém	Popis
B1	Testovacia bunka -> cloudová služba	HTTP Rest API, slúži na synchronizáciu dát

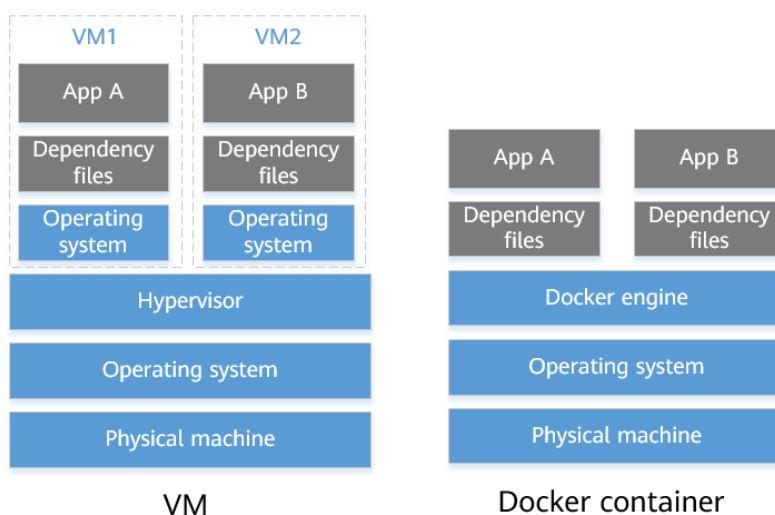
3.2.3.1.2 Komponenty systému

- **Web (UI rozhranie)** - Komponent, ktorý zabezpečuje používateľské rozhranie prostredníctvom webového rozhrania.
- **API** - Komponent, ktorý zabezpečuje REST API služby pre synchronizáciu dát medzi automatickou testovacou bunkou a cloudovou službou.
- **Worker** - Komponent, ktorý reprezentuje sadu aplikačných služieb,
- **Database** - Komponent, ktorý zabezpečuje perzistentné alebo dočasné úložisko pre namerané dáta – relačná databáza.
- **PLC** - Komponent, ktorý zabezpečuje komunikáciu s HW časťami testovacej bunky.
- **Digital Identity Service (DIS)** - Softvérový produkt slovenskej spoločnosti Innovatrics, líder trhu s biometrickým softvérom, slúži na overenie identity osoby automatizovane a na diaľku bez potreby obslužného personálu. Overenie identity funguje tak, že osoba na vyzvanie systému naskenuje doklad totožnosti a umiestni svoju tvár pred kameru. Systém extrahuje z dokladu

totožnosti všetky dostupné informácie a overí pravosť a fyzickú prítomnosť dokumentu pri skenovaní. Sken tváre zabezpečí kontrolu živosti osoby pred kamerou. Následne sú porovnané údaje dokumentu s výsledkami analýzy fotografie osoby. Porovnáva sa tvár, odhadovaný vek, pohlavie a ďalšie parametre. Výsledkom množstva kontrol je skóre autentickosti osoby, ktoré je spolu so všetkými zozbieranými dátami poskytnuté klientskému systému. Služba Digital Identity Service je lokálne hostovaná u klienta buď v cloude, alebo na serveri, ktorý je súčasťou lokálnej infraštruktúry. Jej rozhraním je sa REST API endpointov, ktoré klientský systém volá v rôznych krokoch overovania identity osoby. Služba neuchováva žiadne osobné údaje, perzistencia dát je kompletne v réžii klientskeho systému. V automatizovanej testovacej bunke je služba DIS nasadená ako súčasť lokálnej infraštruktúry danej bunky. Cieľom je umožniť autentifikáciu, a tým dočasný samostatný chod aj v prípade krátkych výpadkov internetového pripojenia. Konfiguračne je však možné smerovať systém automatizovanej bunky aj na cloudové nasadenie služby. DIS je použitá vo fáze overenia identity pacienta, kedy analyzuje doklad z čítačky dokladov a snímky kamery umiestnenej nad displejom testovacej bunky. Použitím tejto služby je docieľené, že potvrdenie o vykonaní testu bude vystavené naozaj na testovanú osobu.

3.2.3.2 Fyzická architektúra systému

Informačný systém pre automatickú testovaciu bunku je vytvorený tak, aby sa dal prevádzkovať v docker kontajneroch (kontajnerizácia). V dnešnej dobe je to najmodernejší spôsob prevádzkovania informačných systémov. V rámci tohto projektu bol jeden z cieľov vyskúšať tento spôsob, aby sme využili jeho výhody voči štandardnej prevádzke vo virtuálnych serveroch (virtualizácia).



Obrázok 15 Docker kontajner vs Virtuálne servery (VM)

Na obrázku je vidieť porovnanie dvoch najpoužívanejších možností prevádzkovania informačných systémov, a to:

1. VM - Virtualizácia,
2. Docker container – Kontajnerizácia.

Virtualizácia – je technológia umožňujúca prevádzkovať viac nezávislých virtuálnych serveroch na jednom fyzickom serveri. Prevádzka informačných systémov vo virtuálnom prostredí (voči prevádzke na fyzickom prostredí) je výhodná hlavne z bezpečnostného a ekonomického dôvodu.

Kontajnerizácia – je virtualizácia za použitia menšieho množstva systémových zdrojov. Aplikácie sú prevádzkované v dockeroch. Docker obsahuje minimálne systémové prostriedky potrebné pre prevádzku aplikácie. Prevádzka pomocou docker kontajnerov je výhodná hlavne z pohľadu náročnosti nasadenia aplikácií, závislosti aplikácií na operačnom systéme a aktualizácií aplikácií a prostredia.

Tabuľka 7 Zoznam kontajnerov

Názov kontajnera	Image	
ld-web	nginx:latest	Kontajner pre webový server
ld-api	mcr.microsoft.com/dotnet/aspnet:5.0	Kontajner pre API služby
ld-worker	mcr.microsoft.com/dotnet/runtime:6.0	Kontajner pre aplikačný server
ld-pgadmin	dpage/pgadmin4:6	Kontajner pre webové rozhrania databázového servera
ld-postgres	postgres:13.4	Kontajner pre databázový server
ld-dis	docker:5000/ld-dis:latest	Kontajner pre digitál identity service

3.2.3.2.1 Prostredia

Cloudová služba pre automatickú testovaciu bunku je prevádzkovaná v jednom vývojovom prostredí na infraštruktúre spoločnosti sféra, a.s.

Tabuľka 8 Zoznam prostredí

Názov	Doména	Infraštruktúra
Vývojové prostredie	lifedefender.sk	HW spoločnosti sféra, a.s.

3.2.3.2.2 Bezpečnosť

Bezpečnosť systému je riešená na úrovni infraštruktúry, ale aj na úrovni aplikačnej vrstvy. Infraštruktúra, do ktorej je systém inštalovaný, je budovaný s ohľadom na vysokú dostupnosť a s požiadavkou na minimalizáciu rizika straty dát. Všetky používateľské aj automatizované rozhrania systému sú zabezpečené voči neautorizovanému vstupu a sú publikované prostredníctvom SSL protokolu.

3.2.3.2.3 Autentifikácia

Autentifikácia je proces identifikácie a overenia identity používateľa pri vstupe do informačného systému. V systéme pre domácu karanténu je autentifikácia používateľov zabezpečená pomocou autentifikačného modulu, ktorý používa štandardy OpenID Connect/OAuth 2.0. Voči tomuto autentifikačnému modulu sa budú overovať používatelia pri vstupe do všetkých rozhraní systému. Autentifikácia je jednofaktorová s použitím prihlasovacieho mena a hesla.

Pri vstupe používateľa do systému autentifikačný overí zadané meno a heslo používateľa. Po úspešnom overení autorizačný modul prideli oprávnenia používateľovi na základe jeho používateľských rolí, v opačnom prípade autentifikačný modul zabráni používateľovi vstup do systému. Ak je používateľ úspešne overený, môže pristupovať k jednotlivým častiam systému na základe pridelených používateľských rolí.

Počas autentifikácie používateľ systému získa autentifikačný token, ktorý má preddefinovanú životnosť. Po expirácii autentifikačného tokenu (dĺžka expirácie je konfigurovateľná, jej dĺžka závisí od bezpečnostných požiadaviek aplikácie) sa používateľ musí opätovne prihlásiť.

Proces autentifikácie používateľa do automatickej testovacej bunky zabezpečuje softvér DIS od spoločnosti Innovatrics, ktorý vykoná overenie používateľa prostredníctvom biometrických údajov získaných kamerou testovacej bunky a informácií v občianskom preukaze.

3.2.3.2.4 Autorizácia

Autorizácia je proces riadenia prístupu používateľa k funkčnostiam a dátam informačného systému. V systéme pre domácu karanténu bude proces riadenia prístupov založený na rolách RBAC (Role-

based access control), čo znamená, že používateľ získa oprávnenia prostredníctvom používateľskej roly. Administrátor pre správu používateľov bude mať možnosť spravovať používateľské roly (katalóg rolí) a pridelovať ich používateľom.

3.3 Prototyp pre Dezinfekčného robota

Cieľom vývoja prototypu SW pre dezinfekčného robota bolo umožniť používateľom možnosti plánovania jednotlivých úkonov, ktoré dokáže dezinfekčný robot vykonávať (dezinfekcia, detekcia vírusov), a taktiež prehľadné zobrazenie histórie jednotlivých uskutočnených výkonov a ich výsledku.

Ako forma SW riešenia bolo zvolené najvhodnejšie riešenie prostredníctvom mobilnej aplikácie.

3.3.1 Popis používateľských scenárov

3.3.1.1 Scenár 1

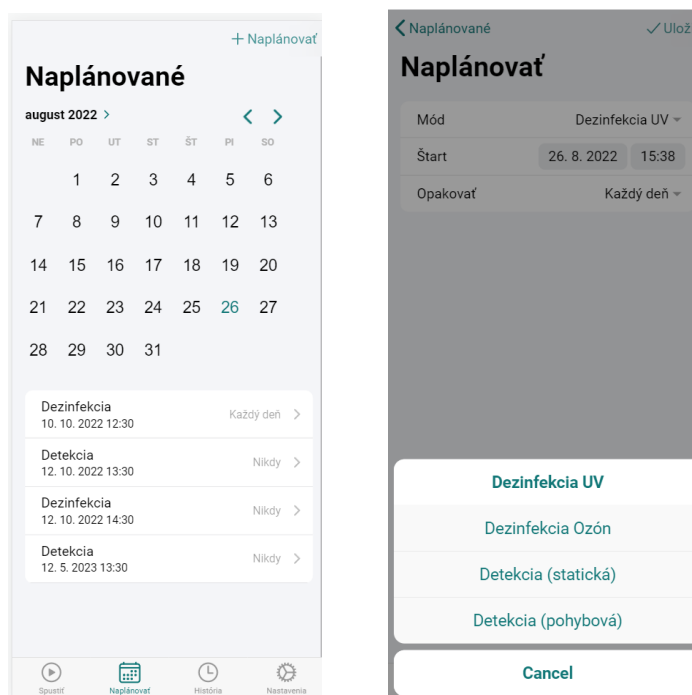
1. Po prihlásení do mobilnej aplikácie je možné spustiť automatickú dezinfekciu okamžite povelom z aplikácie z hlavnej stránky aplikácie.
2. Po výbere módu, ktorý ma byť vykonaný (Dezinfekcia UV, Dezinfekcia ozónom, Detekcia (statická), Detekcia (pohybová)):
 - a. Dezinfekcia prebieha po vopred nasnímaných trajektóriách a detekcia je možná buď statická, ktorá prebieha priamo na nabíjacej stanici, alebo po úprave konfigurácie je možný pohyb pri vykonávaní detekcie.



3.3.1.2 Scenár 2

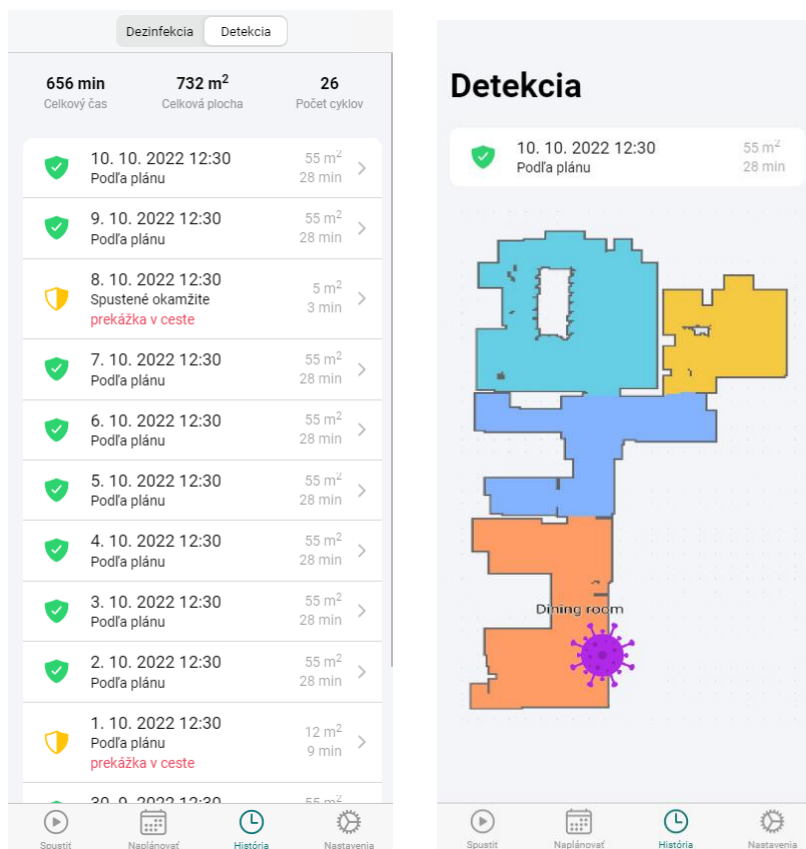
1. Po prihlásení do systému a výbere akcie *Plánovať* z úvodnej obrazovky, systém zobrazí kalendár plánov.
2. Prihlásený operátor si vyberie čas a dátum a mód použitia dezinfekčnej jednotky, a pridá nastavenie.

- Po pridaní nastavenia sú jednotlivé plány viditeľne prostredníctvom kalendára, kde je ich možné upraviť alebo odstrániť.



3.3.1.3 Scenár 3

- Po prihlasení do aplikácie je možné zobrazovať si jednotlivé vykonané akcie jednotlivých módov dezinfekcie alebo detekcie vírusu
- Aplikácia zobrazuje históriu pomocou údajov o vykonaných úkonoch v ich časovom trvaní a prekonanej ploche v textovom režime a taktiež pomocou mapy priestoru



3.3.2 Webové rozhranie

Prototyp SW pre dezinfekčného robota (pre korektné zobrazenie prototypu je odporúčane zobrazit stránku v prehliadači na mobilnom zariadení):

- <https://mobile1.lifedefender.sk/tabs/run>

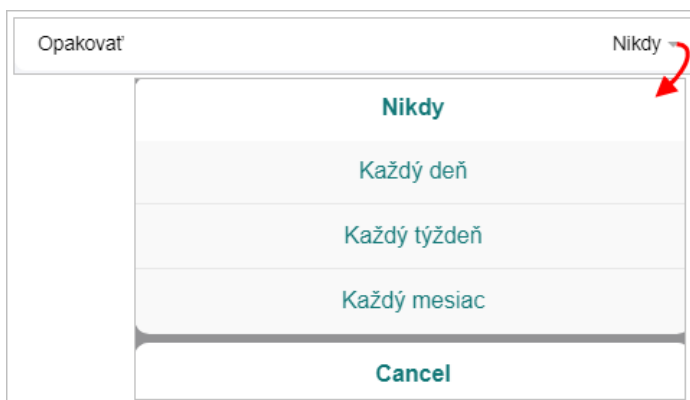
3.3.2.1 Ovládacie prvky

Konvencie ovládacích prvkov:


- **Tlačidlá/lkony** – slúžia na vykonanie požadovanej funkčnosti. Pri umiestnení ukazovateľa myši na tlačidlo sa farba plochy tlačidla zmení z bledej na tmavú. Kliknutím na tlačidlo sa spustí činnosť.

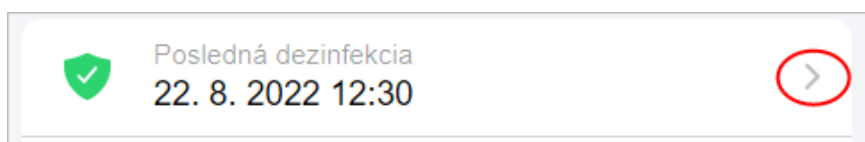




- **Rozbaľovacie polia** - umožňujú nastaviť hodnotu položky rozbalením zoznamu/ponuky a výberom z neho.




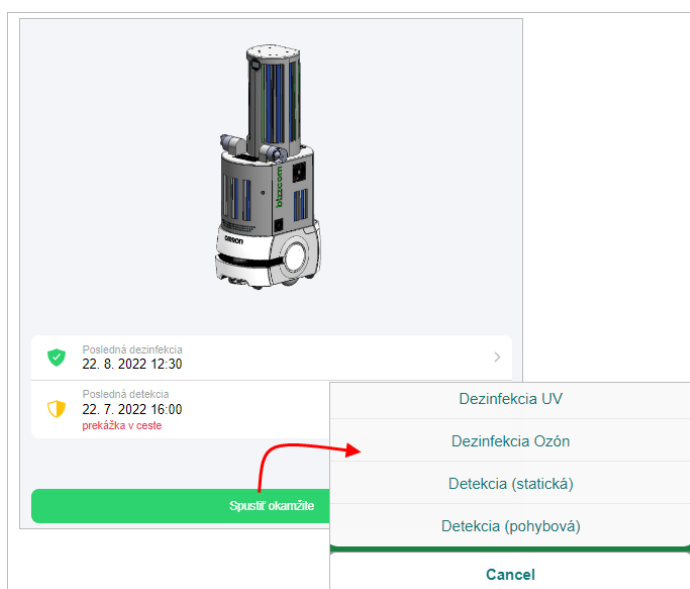
Pole tohto zoznamu je typ ovládacieho prvku, ktorý slúži na výber záznamov v tabuľkovej forme. Je tvorený zoznamom hodnôt v niekoľkoriadkovom poli, pričom platnou hodnotou je vysvietený riadok.


- Pohyb po jednej strane dopredu (na nasledujúcu stranu) 

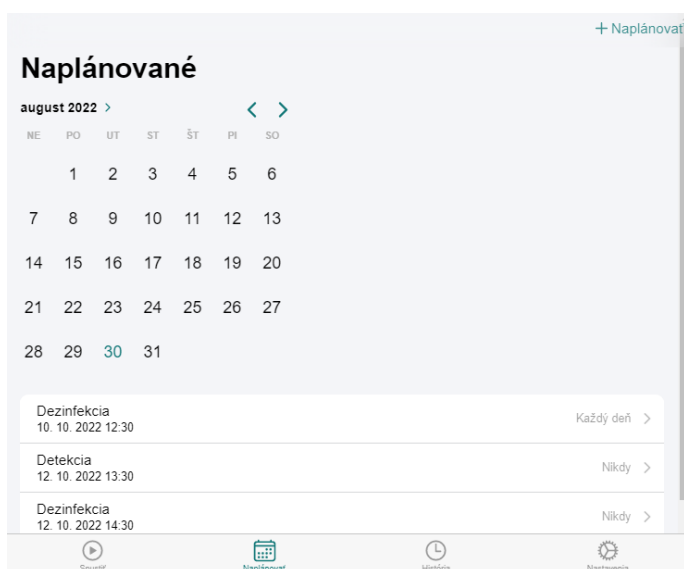


-   - označenie (podľa plánu) pre poslednú dezinfekciu/detekciu.

-  **Spustiť** - stlačením ikony mobilná aplikácia spustí automatickú dezinfekciu okamžite povelom z aplikácie z hlavnej stránky aplikácie:

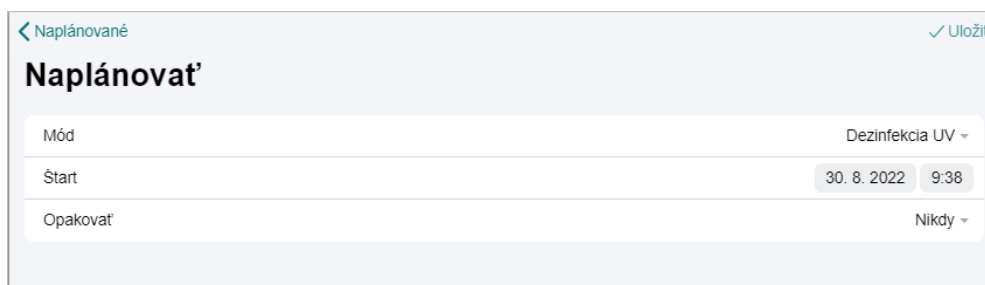



-  **Naplánovať** - stlačením ikony mobilná aplikácia zobrazí stránku **Naplánované/Kalendár plánov**:

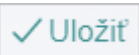



➤  - označenie v kalendári predošlý/nasledujúci mesiac.





➤  - stlačením tlačidla mobilná aplikácia zobrazí formulár **Naplánovať**:




➤  - tlačidlom mobilná aplikácia vráti používateľa späť na formulár **Naplánovať**.


➤  - tlačidlom mobilná aplikácia uloží nový záznam.


➤  - stlačením ikony mobilná aplikácia zobrazí históriu pomocou údajov o vykonaných úkonoch v ich časovom trvaní:

Dezinfekcia		Detekcia	
1256 min Celkový čas	8432 m² Celková plocha	256 Počet cyklov	
 10. 10. 2022 12:30 podľa plánu		56 m ² 13 min	>
 9. 10. 2022 12:30 podľa plánu		56 m ² 13 min	>
 8. 10. 2022 12:30 spustene okamžite prekážka v ceste		7 m ² 3 min	>
 7. 10. 2022 12:30 podľa plánu		56 m ² 13 min	>

-  **Nastavenia** - stlačením ikony mobilná aplikácia zobrazí dialógové okno **Nastavenia**:

Nastavenia





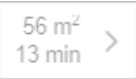
 Vypnúť

 Reštartovať

Reset

Notifikácie scan priestoru

Notifikácie dezinfekcia

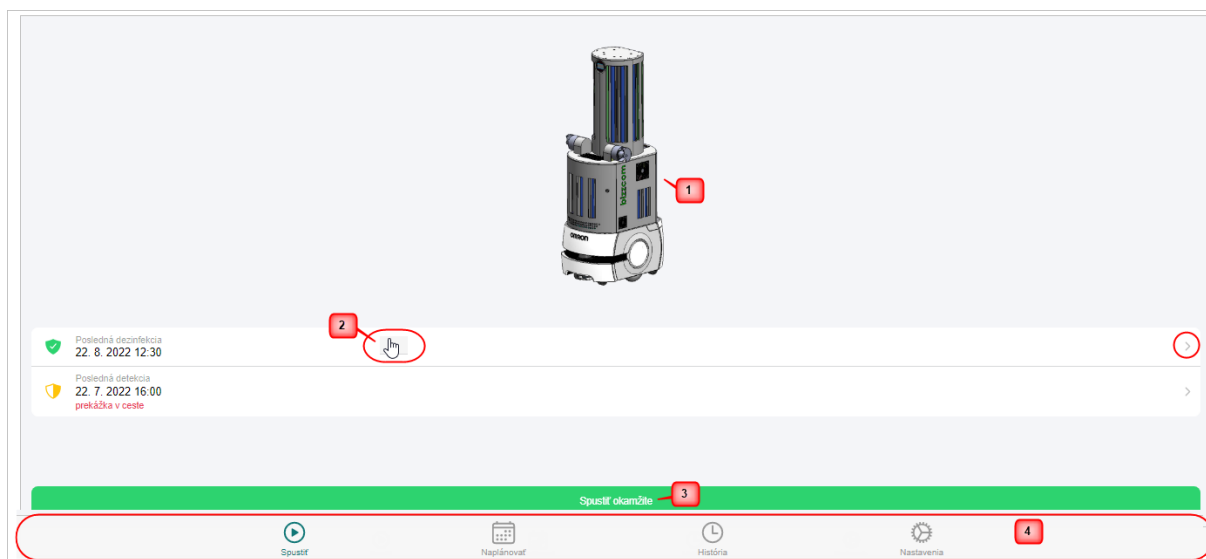
-  - stlačením ikony mobilná aplikácia vypne nastavenia.
-  - stlačením ikony mobilná aplikácia reštartne nastavenia.
-  /  - stlačením ikony mobilná aplikácia sprístupní aktivitu danej činnosti.
-  - stlačením operácie mobilná aplikácia zobrazí stránku **História** s údajmi na mapovom podklade.



3.3.2.2 Rozmiestnenie portálu

Prvá stránka, ktorá sa otvorí používateľovi po prihlásení, je úvodné okno mobilnej aplikácie, ktoré slúži na zobrazenie prehľadu určených údajov. Pri väčšom rozlíšení obrazovky sa portál dynamicky prispôsobí maximálnej šírke a výške prehliadača.

Štandardná stránka portálu je členená na logické časti:



- 1 - zobrazený prototyp SW pre dezinfekčného robota.
- 2 - kliknutím na riadok mobilná aplikácia spustí automatickú dezinfekciu jednotky.
- 3 - kliknutím na tlačidlo mobilná aplikácia ponúkne kontextové menu (výber módu).
- 4 - v spodnej navigačnej lište sú umiestnené operácie pre použitie dezinfekčnej jednotky.

3.3.3 Architektúra systému

Informačný systém pre dezinfekčného robota sa skladá z troch kľúčových častí:

1. Cloudová služba.
2. Interný systém pre ovládanie dezinfekčného robota.
3. Mobilná aplikácia pre ovládanie dezinfekčného robota.

Cloudová služba je budovaná ako platforma pre ukladanie nameraných údajov, ktoré môžu byť získané z rôznych zdrojov. Cloudová služba má rozhranie pre používateľov prostredníctvom webového rozhrania a rozhranie pre automatizovanú komunikáciu prostredníctvom Rest API – určené pre synchronizáciu so systémom pre ovládanie testovacej bunky.

Cloudová služba je vytvorená pomocou nasledovných technológií:

- Angular,
- Microsoft .NET Core,
- PostgreSQL,
- Docker.

Angular ako framework pre tvorbu webového používateľského rozhrania, bol použitý na vytvorenie samostatnej UI vrstvy. Táto UI vrstva komunikuje prostredníctvom Rest API služieb s biznis modelom, ktorý poskytuje operácie a dáta. Rest API služby sú vytvorené pomocou frameworku Microsoft .NET Core s využitím ORM (Object Relational Mapping) Entity Framework Core. Ako relačná databáza pre ukladanie perzistentných dát bola zvolená PostgreSQL databáza. Jednotlivé komponenty informačného systému sú publikované prostredníctvom technológie Docker.

Tabuľka 9 Zoznam endpointov systému

Názov	Endpoint
Používateľské rozhranie	https://app.lifedefender.sk
API endpointy	https://api2.lifedefender.sk

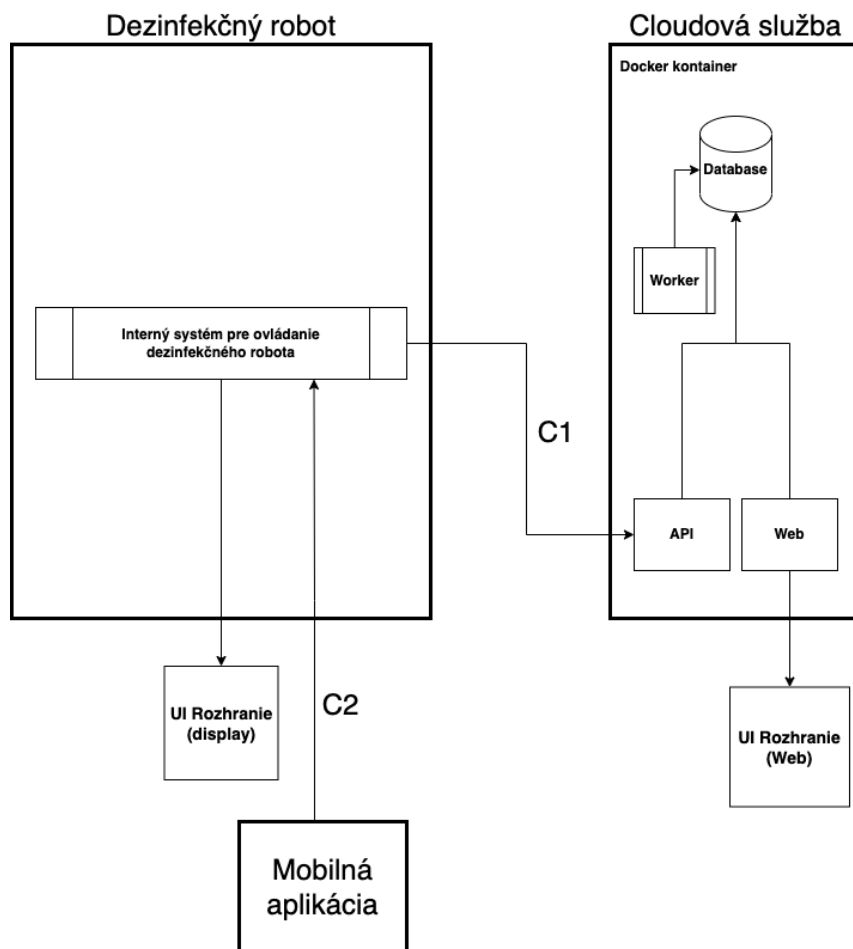
Systém pre ovládanie dezinfekčného robota bunky je vnorený systém prevádzkovaný priamo v automatickej testovacej bunke. Primárnou úlohou systému je ovládanie dezinfekčného robota a synchronizácia nameraných dát do cloudovej služby.

Mobilná aplikácia pre ovládanie dezinfekčného robota je mobilná aplikácia, pomocou ktorej je možné vzdialene ovládať dezinfekčného robota.

Všetky zvolené technológie a frameworky sú open source.

3.3.3.1 Logická architektúra systému

Na nasledujúcom obrázku je vidieť základnú logickú architektúru systému.



Obrázok 16 Logická architektúra systému

Cloudová služba sa skladá z nasledujúcich logických komponentov:

- Web,
- API,
- Worker,
- Databáza.

3.3.3.2 Typy integrácií

Primárna integrácia je medzi dezinfekčným robotom a cloudovou službou. Pomocou tejto integrácie sa synchronizujú nameraná dáta. Ďalšia integrácia je medzi dezinfekčným robotom a mobilnou aplikáciou.

Tabuľka 10 Zoznam integrácií

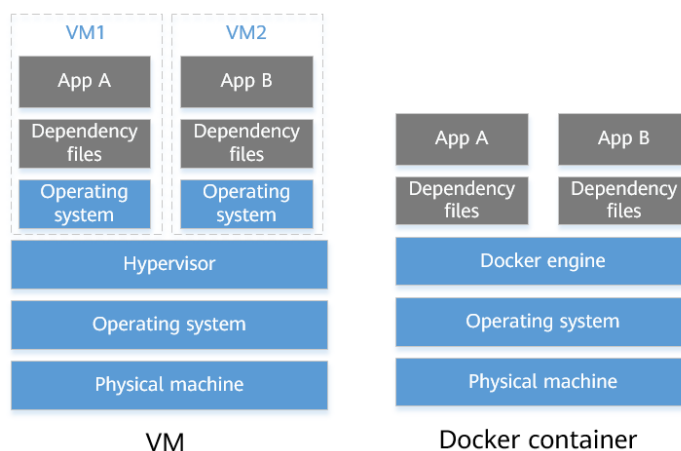
Integrácia	Systém	Popis
C1	Dezinfekčný robot -> cloudová služba	HTTP Rest API, slúži na synchronizáciu dát
C2	Mobilná aplikácia -> cloudová služba	Bluetooth, služba pre ovládanie robota

3.3.4 Komponenty systému

- **Web (UI rozhranie)** - Komponent, ktorý zabezpečuje používateľské rozhranie prostredníctvom webového rozhrania.
- **API** - Komponent, ktorý zabezpečuje REST API služby pre synchronizáciu dát medzi automatickou testovacou bunkou a cloudovou službou.
- **Worker** - Komponent, ktorý zabezpečuje vykonávanie plánovaných úloh.
- **Database** - Komponent, ktorý zabezpečuje perzistentné úložisko dát – relačná databáza.

3.3.5 Fyzická architektúra systému

Cloudová služba pre dezinfekčného robota je vytvorená tak, aby sa dala prevádzkovať v docker kontajneroch (kontajnerizácia). V dnešnej dobe je to najmodernejší spôsob prevádzkovania informačných systémov. V rámci tohto projektu bol jeden z cieľov vyskúšať tento spôsob, aby sme využili jeho výhody voči štandardnej prevádzke vo virtuálnych serveroch (virtualizácia).



Obrázok 17 Docker kontajner vs Virtuálne servery (VM)

Na obrázku je vidieť porovnanie dvoch najpoužívanejších možností prevádzkovania informačných systémov, a to:

1. VM - Virtualizácia.
2. Docker container – Kontajnerizácia.

Virtualizácia – je technológia umožňujúca prevádzkovať viac nezávislých virtuálnych serveroch na jednom fyzickom serveri. Prevádzka informačných systémov vo virtuálnom prostredí (voči prevádzke na fyzickom prostredí) je výhodná hlavne z bezpečnostného a ekonomického dôvodu.

Kontajnerizácia – je virtualizácia za použitia menšieho množstva systémových zdrojov. Aplikácie sú prevádzkované v dockeroch. Docker obsahuje minimálne systémové prostriedky potrebné pre prevádzku aplikácie. Prevádzka pomocou docker kontajnerov je výhodná hlavne z pohľadu náročnosti nasadenia aplikácií, závislosti aplikácií na operačnom systéme a aktualizácií aplikácií a prostredia.

Tabuľka 11 Zoznam kontajnerov

Názov kontajnera	Image	Popis
ld-web	nginx:latest	Kontajner pre webový server
ld-api	mcr.microsoft.com/dotnet/aspnet:5.0	Kontajner pre API
ld-worker	mcr.microsoft.com/dotnet/runtime:6.0	Kontajner pre aplikačný server
ld-pgadmin	dpage/pgadmin4:6	Kontajner pre webové rozhrania databázového servera

Názov kontajnera	Image	Popis
ld-postgres	postgres:13.4	Kontajner pre databázový server

3.3.5.1 Prostredia

Cloudová služba pre dezinfekčného robota je prevádzkovaná v jednom vývojovom prostredí na infraštruktúre spoločnosti sféra, a.s.

Tabuľka 12 Zoznam prostredí

Názov	Doména	Infraštruktúra
Vývojové prostredie	lifedefender.sk	HW spoločnosti sféra, a.s.

3.3.5.2 Bezpečnosť

Bezpečnosť systému (cloudová služba) je riešená na úrovni infraštruktúry, ale aj na úrovni aplikačnej vrstvy. Infraštruktúra, do ktorej je systém inštalovaný, je budovaný s ohľadom na vysokú dostupnosť a s požiadavkou na minimalizáciu rizika straty dát. Všetky používateľské aj automatizované rozhrania systému sú zabezpečené voči neautorizovanému vstupu, a sú publikované prostredníctvom SSL protokolu.

3.3.5.3 Autentifikácia

Autentifikácia je proces identifikácie a overenia identity používateľa pri vstupe do informačného systému. V systéme pre domácu karanténu je autentifikácia používateľov zabezpečená pomocou autentifikačného modulu, ktorý používa štandardy OpenID Connect/OAuth 2.0. Voči tomuto autentifikačnému modulu sa budú overovať používatelia pri vstupe do všetkých rozhraní systému. Autentifikácia je jednofaktorová s použitím prihlasovacieho mena a hesla.

Pri vstupe používateľa do systému autentifikačný overí zadané meno a heslo používateľa. Po úspešnom overení autorizačný modul pridelí oprávnenia používateľovi na základe jeho používateľských rolí, v opačnom prípade autentifikačný modul zabráni používateľovi vstup do systému. Ak je používateľ úspešne overený, môže pristupovať k jednotlivým častiam systému na základe pridelených používateľských rolí.

Počas autentifikácie používateľ systému získa autentifikačný token, ktorý má preddefinovanú životnosť. Po expirácii autentifikačného tokenu (dĺžka expirácie je konfigurovateľná, jej dĺžka závisí od bezpečnostných požiadaviek aplikácie) sa používateľ musí opätovne prihlásiť.

3.3.5.4 Autorizácia

Autorizácia je proces riadenia prístupu používateľa k funkčnostiam a dátam informačného systému. V systéme pre domácu karanténu bude proces riadenia prístupov založený na rolách RBAC (Role-based access control), čo znamená, že používateľ získa oprávnenia prostredníctvom používateľskej roly. Administrátor pre správu používateľov bude mať možnosť spravovať používateľské roly (katalóg rolí) a pridelať ich používateľom.

4 EXPERIMENTÁLNY VÝVOJ PROTOTYPU MOBILNEJ APLIKÁCIE

4.1 Prototyp pre domácu karanténu

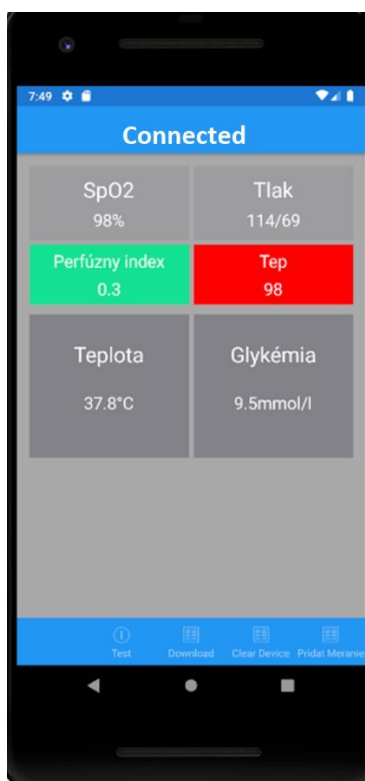
Cieľom vývoja prototypu mobilnej aplikácie pre domácu karanténu bolo vytvorenie SW riešenia, ktoré umožní používateľom monitorovanie svojho zdravotného stavu podľa veličín meraných HW zariadením (teplota, tlak). Mobilná aplikácia po spárovaní komunikáciu s HW zariadením a zaznamenáva namerané údaje

4.1.1 Zber a synchronizácia dát

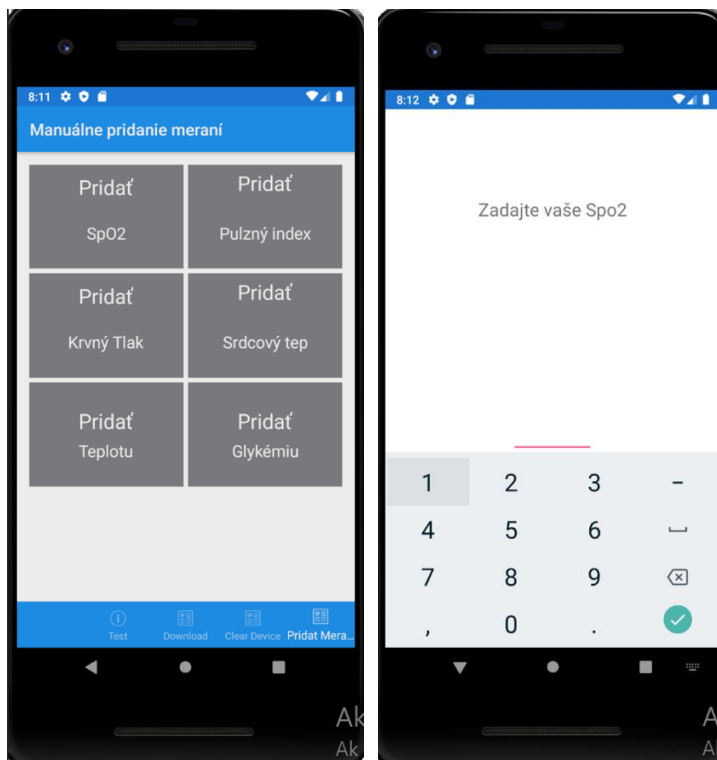
Pre zabezpečenie zberu dát je nutné, aby zariadenie bolo pripojené na smartfón (alebo iné smart zariadenie) cez bluetooth. Namerané hodnoty sú automaticky poslané do pripojeného zariadenia, a v pravidelných dávkach sú dáta zasielané aj do webového portálu.

4.1.2 Popis GUI

Pacient na prehľadnej obrazovke vidí posledné merania (pozri nasledujúci obrázok), v prípade ak hodnoty sú kritické, tak sú zobrazené v červenom poli.

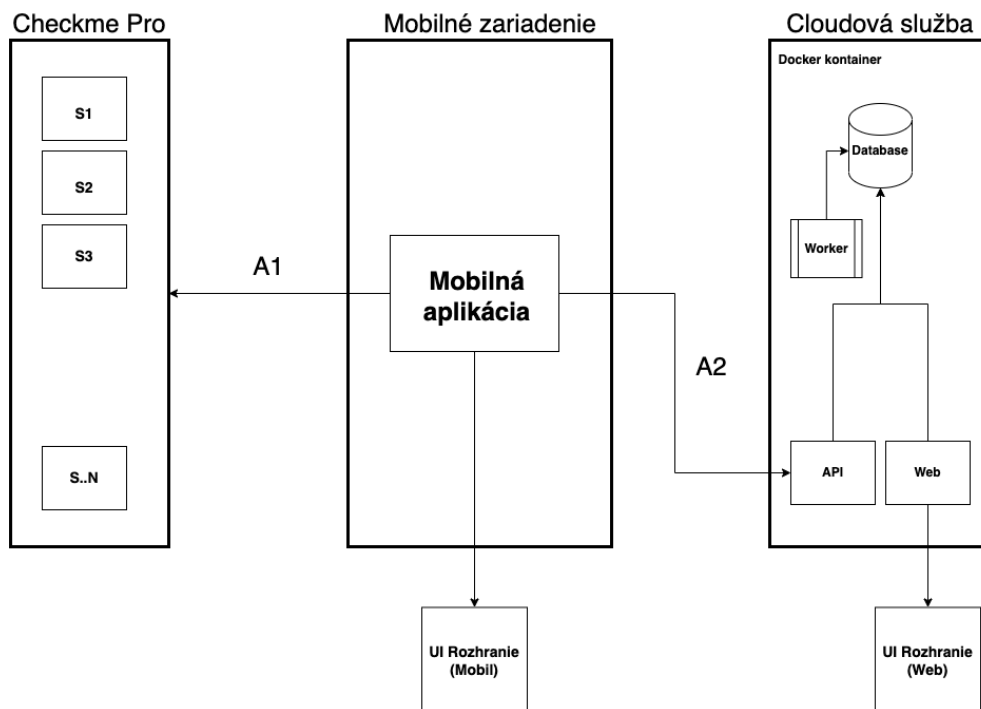


Merania je možné zadávať aj manuálne



4.1.3 Architektúra

Úlohou mobilnej aplikácie pre domácu karanténu je synchronizovanie nameraných dát medzi sádov meracích zariadení a cloudovou službou. V našom prípade bola integrácia vykonaná voči zariadeniu Checkme. Mobilná aplikácia je navrhnutá tak, aby umožňovala získavania nameraných dát aj z iných meracích zariadení – napr. z inteligentných hodienok.



Obrázok 18 Logická architektúra pre systém domácej karantény

4.1.4 Technológie

Na vývoj mobilnej aplikácie pre systém domácej karantény bol zvolený framework [Xamarin](#). Hlavný dôvod výberu tejto technológie bola možnosť vyvíjať mobilnú aplikáciu pre obe najpoužívanejšie platformy (iOS a Android) súčasne. Pri vývoji sa využíval nástroj Visual Studio 2019, a ako úložisko zdrojových súborov sa použil nástroj [Azure DevOps Server](#).

4.1.5 Rozhrania

Mobilná aplikácia obsahuje dve integrácie, a to integráciu smerom na cloudovú službu a smerom na meracie zariadenie Checkme Pro. V oboch prípadoch je iniciátorom komunikácie mobilná aplikácia.

Tabuľka 13 Zoznam integrácií

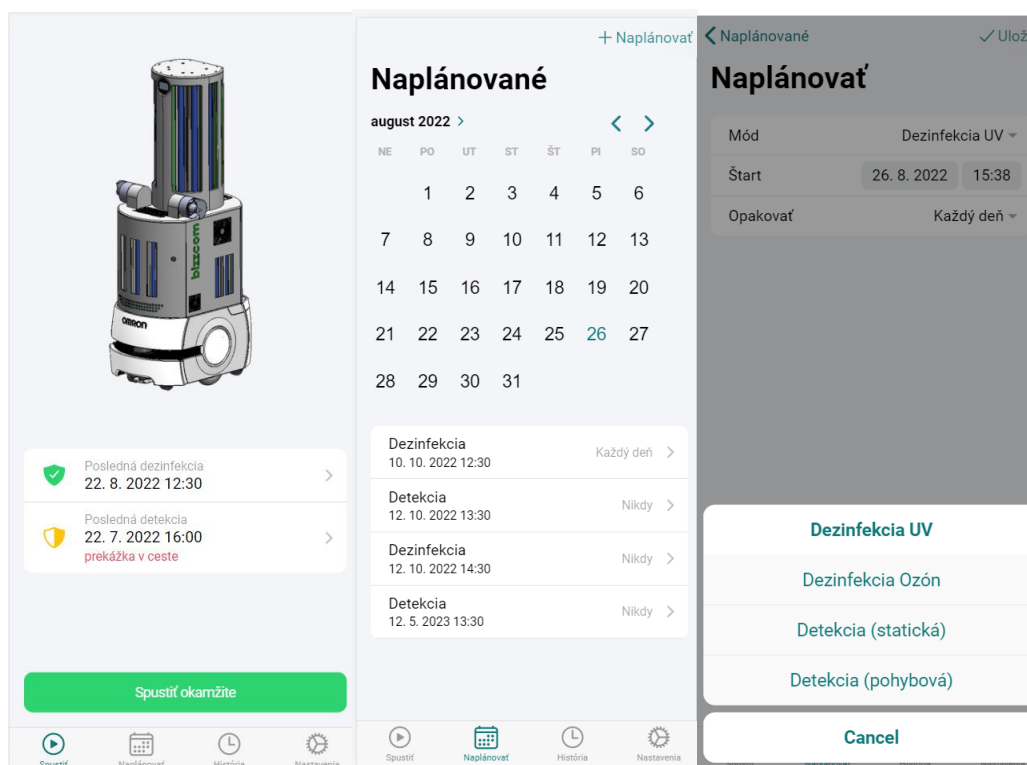
Integrácia	System	Popis
A1	Mobilná ap. -> Checkme Pro	Bluetooth, slúži na synchronizáciu
A2	Mobilná ap. -> cloudová služba	HTTP Rest API, slúži na synchronizáciu

Najzložitejším rozhraním bola práve integrácia A1 (mobilná aplikácia vs Checkme Pro). Na vytvorenie integrácie bolo potrebné dobre nastudovať možnosti meracieho zariadenia Checkme Pro.

4.2 Prototyp pre Dezinfekčného robota

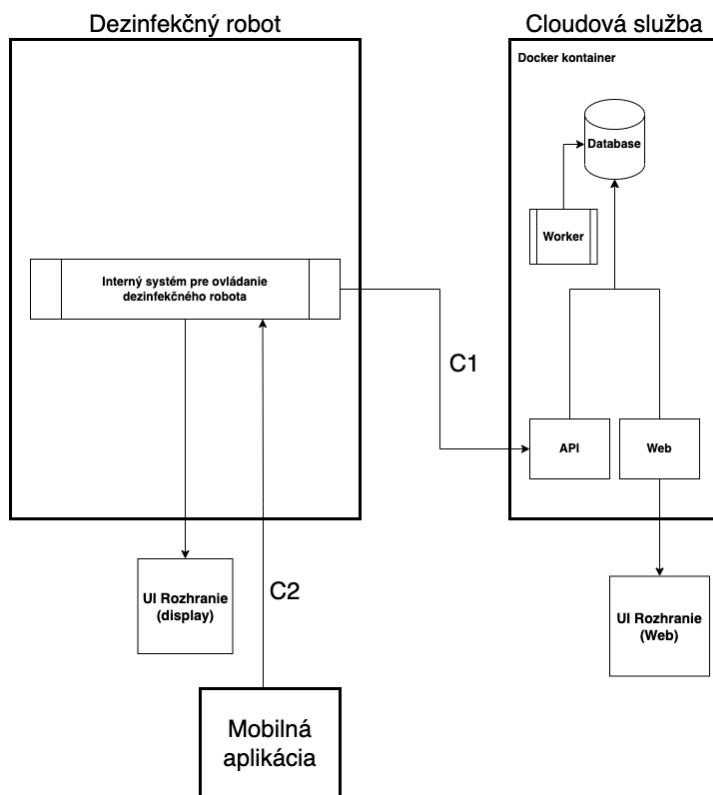
4.2.1 Popis GUI

Používateľské rozhranie aplikácie bolo navrhnuté tak, aby mal používateľ základný prehľad o stave jednotlivých módov robota už na základnej obrazovke a rýchlu možnosť prístupu k naplánovaniu jednotlivých úkonov v čase. Používateľské scenáre sú opísané v predchádzajúcich kapitolách (kap. [3.3.1 Popis používateľských scenárov](#)).



4.2.2 Architektúra

Úlohou mobilnej aplikácie pre dezinfekčný robot je diaľkové ovládanie tohto zariadenia. Na nasledujúcom obrázku je vidieť logickú architektúru systému pre automatickú dezinfekciu.



Obrázok 19 Logická architektúra pre systém automatickej dezinfekcie

Mobilná aplikácia komunikuje s interným systémom pre ovládanie dezinfekčného robota prostredníctvom technológie bluetooth.

4.2.3 Technológie

Na vývoj mobilnej aplikácie pre systém automatickej dezinfekcie bol zvolený framework [Xamarin](#). Hlavný dôvod výberu tejto technológie bola možnosť vyvíjať mobilnú aplikáciu pre obe najpoužívanejšie platformy (iOS a Android) súčasne. Pri vývoji sa využíval nástroj Visual Studio 2019, a ako úložisko zdrojových súborov sa použil nástroj [Azure DevOps Server](#).

4.2.4 Rozhrania

Mobilná aplikácia obsahuje integráciu smerom na dezinfekčný robot. Úlohou integrácie je diaľkové ovládanie robota. Dezinfekčný robot obsahuje integráciu aj smerom na cloudovú službu, ktorej úlohou je synchronizácia dát.

Tabuľka 14 Zoznam integrácií

Integrácia	Systém	Popis
C1	Dezinfekčný robot -> cloudová služba	HTTP Rest API, slúži na synchronizáciu dát
C2	Mobilná aplikácia -> cloudová služba	Bluetooth, služba pre ovládanie robota

5 EXPERIMENTÁLNY VÝVOJ PROTOTYPU MODULU POKROČILEJ ANALÝZY A VIZUALIZÁCIE DÁT

5.1 Prototyp modulu vizualizácie dát

5.1.1 Kibana

Kibana je vizuálny nástroj, ktorým je možné interaktívnym spôsobom prezerať dáta uložené v Elasticsearch. Pomocou Kibany je možné:

1. Vyhľadávať, monitorovať a chrániť dáta. Od zobrazenia štruktúrovaných dát cez analýzu logov až po nájdenie bezpečnostných zraniteľností v IT infraštruktúre.
2. Analyzovať dáta – nájsť skryté súvislosti a korelácie medzi dátami, vizualizovať zaujímavé nálezy v diagramoch, grafoch, mapách a tieto vhodne skombinovať na dashboard.

Kibana je určená pre administrátorov, dátových analytikov a biznis používateľov. Administrátor má za úlohu riadiť Elastic Stack od nasadenie Elasticsearch databázy, konfiguráciu Kibana portálu a zabezpečenie toku dát, ktoré prúdia do Elasticsearch.

Analytik má za úlohu dať surovým dátam zmysel, nájsť súvislosti, vizualizovať dáta na dashboarde do prezentatívnej formy pre biznis používateľa.

Biznis používateľ vie zobrazovať jednotlivé dashboards poprípade sa vŕtať v detailoch. Kibana dokáže pracovať so všetkými typmi dát. Štruktúrované dokumenty alebo neštruktúrovaný text, časové rady alebo číselné hodnoty, logy, priestorové geospatial dáta, metriky či udalosti v čase. Kibana pomáha objaviť neznáme vzory a vzťahy a vizualizovať výsledok.

Elasticsearch je distribuovaný analytický a vyhľadávací engine. Dokáže vyhľadávať a analyzovať takmer v reálnom čase nad rôznymi typmi dát, ako sú štruktúrované, neštruktúrovaný text, hodnoty alebo priestorové geospatial dáta. Elasticsearch tieto dáta efektívne ukladá a indexuje tak, aby sa dali rýchlo nájsť. Okrem jednoduchého vyhľadávania dokáže vykonávať zložitejšie agregácie objaviť tak rôzne trendy a vzory. Ako dáta pribúdajú, nasadenie Elasticsearch je možné škálovať. Vyhľadávanie a agregácie potom bežia distribuovane na základe požiadaviek aplikácie na nízku odozvu a počet súbežných dotazov.

Elasticsearch pridáva množinu JSON dokumentov, ktoré spĺňajú rovnakú schému, schopnosť rýchleho vyhľadávania a agregácie. Dokumenty sú interne uložené ako indexy a dátové toky, ktoré reprezentujú jeden spoločný typ dát.

Elasticsearch poskytuje nasledovné časté prípady použitia:

- Pridanie rýchleho full-text vyhľadávania na webstránku s možnosťou automatického dopĺňania textu,
- Ukladania a analýza logov, metrik a systémových udalostí,
- Použitie strojové učenie na automatické modelovanie chovania dát v reálnom čase,
- Automatizáciu biznis procesov použitím Elasticsearch ako úložiska,
- Manažovať, integrovať a analyzovať priestorové informácie použitím Elasticsearch ako GIS systém,
- Ukladať a spracovávať genetické dáta použitím Elasticsearch ako bio informačný výskumný nástroj.

Elasticsearch je distribuované úložisko dokumentov. Namiesto ukladania informácií po riadkoch a stĺpcoch, ako ukladajú tradičné relačné SQL databázy, Elasticsearch ukladá komplexné dátové štruktúry serializované ako JSON dokumenty. Ak je nasadených viacero Elasticsearch uzlov v klastri, dáta sú naprieč klastrom rovnomerne rozložené a sú okamžite prístupné z akéhokoľvek uzla.

Každý uložený dokument je indexovaný a plne vyhľadateľný skoro v reálnom čase. Elasticsearch využíva tzv. prevrätý index, ktorý umožňuje veľmi rýchle full-text vyhľadávanie. Slovo, ktoré sa nachádza v ľubovoľnom dokumente, obsahuje invertovaný index unikátne iba raz a zároveň identifikuje

všetky dokumenty, v ktorých sa dané slovo nachádza. Pod indexom si vieme predstaviť optimalizovanú množinu dokumentov pričom každý dokument obsahuje sadu položiek uložených vo forme kľúč-hodnota. Elasticsearch štandardne indexuje všetky dáta, tzn. všetky položky, kde každá je uložená vo vyhradenej optimalizovanej štruktúre. Napríklad textové položky sú uložené v invertovaných indexoch a priestorové dáta v BKD stromoch. Schopnosť použitia samostatnej štruktúry na jednotlivé položky robí Elasticsearch takým rýchlym. Elasticsearch umožňuje dokumenty indexovať aj bez schémy, to znamená nie je potrebné dopredu definovať názvy všetkých položiek v dokumente. Ak sa v novom dokumente objaví nová položka, Elasticsearch automaticky začne indexovať túto novú položku.

Schopnosť vyhľadávať znamená schopnosť filtrovať dokumenty na základe pravidiel a podmienok. Napríklad je možné nájsť všetky dokumenty za posledných x dní alebo dokument obsahujúci konkrétne slovo. Kibana poskytuje viacero spôsobov, ako skonštruovať filtre a dotazy. Tieto sa následne vykonávajú voči Elasticsearch inštancii.

Agregácia znamená možnosť získať napríklad sumárne hodnoty všetkých dokumentov, ktoré spĺňajú daný filter. Najčastejšou používanou agregáciou je počet (Count) a je často používaný v kombinácii s dátumovým histogramom na získanie počtu jednotiek v čase. Špeciálna agregácia „term“ ukáže najčastejšie výskyty hodnôt z množiny dokumentov, ktoré spĺňajú filter.

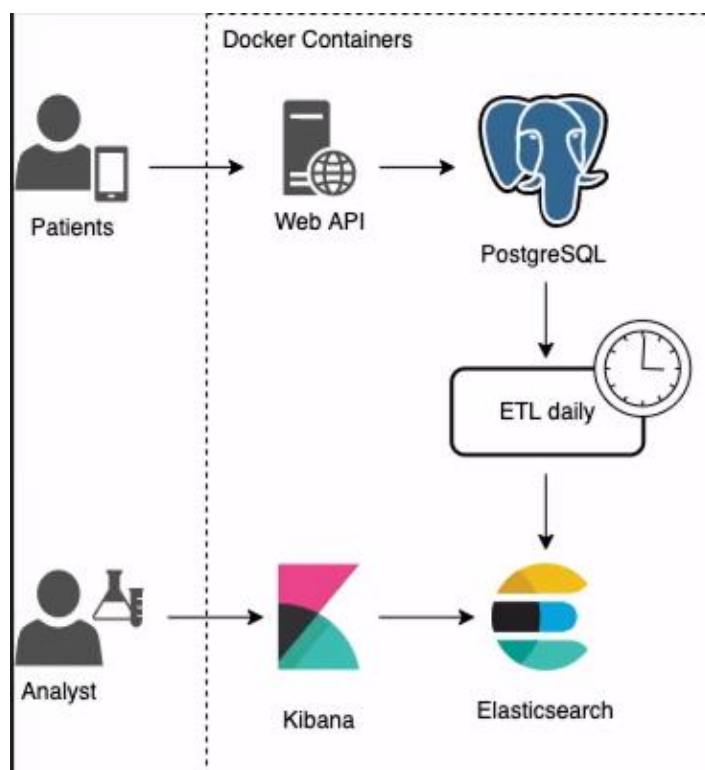
Kibana vyžaduje vytvoriť pohľad na dáta (data view) čím inštruuje Elasticsearch, ku ktorým dátam chceme prísť a či dáta obsahujú časový údaj. Pohľad na dáta môže v Elasticsearch referencovať na viacero dátových tokov alebo indexov.

Dátové pohľady sú vo väčšine prípadov vytvorené administrátorom v čase ukladania JSON dokumentov do Elasticsearch databázy. Kibana používa dátové pohľady na zobrazenie zoznamu veličín. V pohľade je možné prispôsobiť zobrazenie a formát pre každú veličinu. Napríklad veličinu pre dĺžku trvania je možné zobrazit' s jednotkou sekundy. Kibana podporuje formáty veličín pre reťazce, dátumy, čísla a geointy.

Kibana poskytuje niekoľko spôsobov, ako vytvárať vyhľadávacie dotazy, ktoré redukujú výslednú množinu dokumentov spĺňajúcich filter.

Polo-štruktúrované vyhľadávanie kombinuje vyhľadávanie voľného textu so štruktúrovaným vyhľadávaním hodnôt veličín použitím Kibana Query Language (KQL).

Štruktúrované filtre sú interaktívnou možnosťou, ako vytvoriť dotaz nad Elasticsearch a sú často používané napríklad v prípade vytvárania dashboardu zdieľanými medzi viacerými analytikmi. Jednotlivé štruktúrované filtre je možné kombinovať medzi sebou logickým operátorom AND.



Obrázok 20 Schéma plánovaného rozhrania

5.1.2 Zoznam grafických metód

Grafické metódy sú nástroje grafickej analýzy dát. Treba však brať do úvahy, že grafická analýza dát nie je kompletná analýza dát. Grafické výstupy slúžia na odhalenie niektorých informácií v množine dát. Vo všeobecnosti sa ľahšie interpretujú ako tabuľková forma dát alebo výstupy zo štatistického modelovania a vyjadrujú viac kvalitatívnych informácií. Vo všeobecnosti je vždy efektívnejšie zobrazit' dáta viacerými metódami, ktoré ponúkajú veľa rôznych pohľadov, aby sa zo súboru údajov získalo čo najviac informácií. Grafická analýza dát je proces, v ktorom sa paralelne sleduje viacero aspektov, tak ako by to malo byť v každom výskumnom procese. Hlavným cieľom je nájsť optimálnu skupinu grafických metód, ktoré čo najlepšie zhrnú grafickú interpretáciu množiny dát.

Grafická analýza údajov by mala byť kombinovaná a overovaná štatistickými metódami. Často sa využíva na kontrolu po aplikácii štatistických modelov. Slúži na odhaľovanie skrytých štruktúr v dátach. Grafická analýza a štatistické metódy by sa nemali vnímať ako konkurenčné metódy, ale ako metódy, ktoré sa navzájom dopĺňujú. Grafická analýza údajov je vhodná pre dáta, kde nemusia platiť štandardné štatistické predpoklady, ktoré sú potrebné na tvorbu modelov. Zvlášť je cenná pri analýze experimentálnych dát z výskumných projektov. Pomáha odhaľovať výskyt rôznych vzorov alebo iných faktorov, ktoré sa menia v čase.

5.1.2.1 Grafické zobrazenie spojitých premenných

Medzi typických zástupcov spojitých premenných patria časové rady. Spojitá premenná môže teoreticky nadobúdať ľubovoľnú hodnotu v rámci svojho rozsahu. (V praxi sa údaje pre spojitú premennú spravidla zaokrúhľujú na určitú úroveň presnosti merania). Na zobrazenie rozloženia údajov spojitých premenných bolo navrhnutých mnoho rôznych grafických metód a všetky majú rovnaký cieľ, a to zobrazit' dôležité charakteristiky dát. Niektoré grafické metódy vyzdvihujú jednu vlastnosť na úkor druhej, niektoré sú veľmi špecializované, niektoré vyžadujú pokročilé interpretačné schopnosti používateľa. Dôvodom vysokého počtu metód je veľké množstvo rôznych typov a druhov charakteristík vyskytujúcich sa v dátach. V tejto podkapitole sú uvedené histogramy, krabicové diagramy a iné.

5.1.2.2 Charakteristiky spojitých premenných

Asymetria - rozdelenie premennej je vychýlené doľava alebo doprava, napríklad rozdelenie peňažných príjmov.

Odľahlé (extrémne) hodnoty - sa vyskytujú, keď jedna alebo viac hodnôt sú vzdialené od zvyšku množiny dát.

Multimodalita - nastáva, keď rozdelenie spojitkej premennej má viac ako jeden vrchol.

Prázdne priestory - ide o oblasti množiny dát, ktoré neobsahujú žiadne dáta. Štandardne sa považujú chýbajúce hodnoty za neznáme hodnoty. V časovom rade môžu byť nejaké pozorovania neznáme.

Zaokrúhľovanie - vyskytuje sa len pri niektorých hodnotách (možno celých číslach), napríklad pri vekových rozdeleniach.

Hromadenie (častý výskyt) rovnakých hodnôt - v časových radoch môže byť v dominantnej miere prítomná číslica 0. Podľa uváženia je potom vhodné všetky nulové hodnoty vynechať z analýzy.

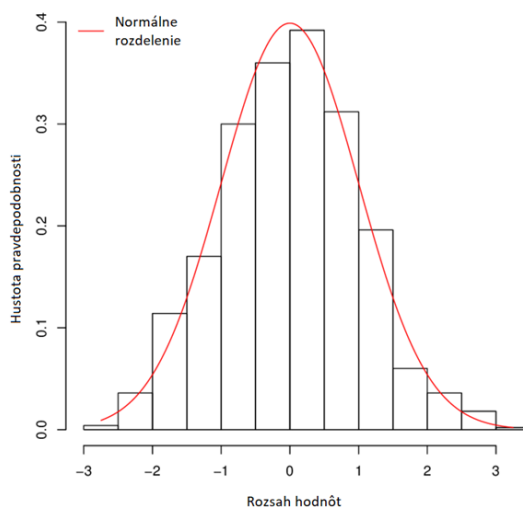
Absurdity - vyskytujú sa hodnoty, ktoré sa v praxi principiálne nevyskytujú napríklad záporný vek.

Chyby - hodnoty, ktoré z kontextu skúmaného procesu sa nemôžu vyskytovať, ale sú prítomné.

Grafická analýza je vhodná na zobrazenie prvkov, ktoré tvoria tvary pravdepodobnostného rozdelenia údajov. Môžu poskytnúť viac a iné druhy informácií ako súbor súhrnných štatistík. Je zrejmé, že najlepšie je používať oba prístupy. Pri jednej premennej je priemer zvyčajne najdôležitejšou štatistikou a možno žiadny štatistický test sa nepoužíva tak často ako t-test na testovanie priemerov. T-test sa môže použiť, ak základné údaje pochádzajú z normálneho rozdelenia. Pri malých súboroch údajov (a t-test je určený práve pre malé vzorky) môžu údaje z normálneho rozdelenia vyzeráť veľmi nenormálne, preto majú testy normality nízku silu a poskytujú malú podporu pre t-testy. T-test je však voči nenormálnosti obecné odolný. To by však nemalo brániť tomu, aby pred vykonaním analýz bolo vykonané testovanie, či dáta pochádzajú z normálneho rozdelenia. Takúto analýzu je možné vykonať aj graficky.

5.1.2.3 Histogram

Histogram (pozri nasledujúci obrázok) je grafická metóda, ktorá organizuje skupinu nameraných dát do používateľom zadáných rozsahov. Histogram, podobne ako stĺpcový graf, zhŕňa rad údajov do ľahko interpretovateľného vizuálneho zobrazenia tým, že berie mnoho dátových bodov a zoskupuje ich do logických rozsahov alebo intervalov. Histogram technicky zobrazuje pravdepodobnostné rozdelenie meraného procesu z danej množiny dát.



Obrázok 21 Príklad histogramu

5.1.2.4 Krabicový diagram

Krubicový diagram (pozri nasledujúci obrázok) alebo box-plot je jedna z grafických metód používaných v štatistike, ktorá umožňuje skúmať údaje pomocou kvartilov, ktoré patria do triedy kvantilov. Kvantily sú hodnoty, ktoré rozdeľujú usporiadaný súbor hodnôt na určitý počet rovnako obsadených častí. Vo všeobecnosti platí, že x_p , kde "p" je percentuálny kvantil. Najčastejšie používané kvantily sú kvartily. Sú to hodnoty, ktoré rozdeľujú súbor na štyri časti, z ktorých každá obsahuje 25 % jednotiek a označujú sa takto:

- x_{25} = dolný kvartil,
- x_{50} = druhý kvartil, nazývaný aj medián a bežne označovaný ako \bar{x} s priebehom,
- x_{75} = horný kvartil.

Počítačové programy sa používajú na výpočet kvartilov. Avšak aj bez vzorcov môžete pomerne ľahko určiť kvantily.

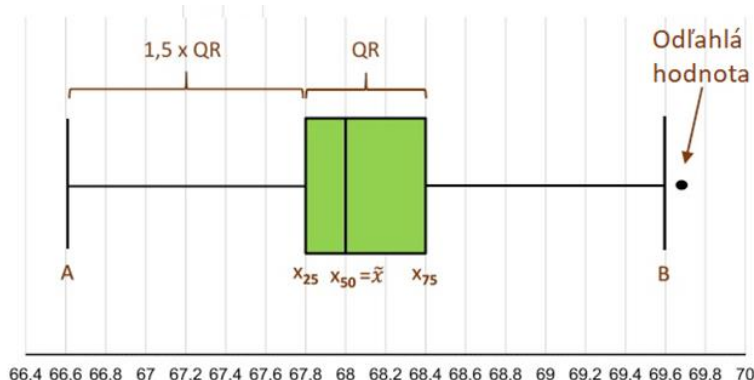
Krubicový diagram umožňuje:

- Identifikovať odľahlé hodnoty.
- Posúdiť symetrie na koncoch rozdelení.
- Porovnať odchýlky pre dva alebo viac súborov hodnôt a v prípade potreby posúdiť oprávnenosť procesu.

Krubicový diagram je nástroj, ktorý umožňuje identifikovať odľahlé hodnoty v množine hodnôt, ktoré pri klasickom spracovaní údajov skresľujú výsledné štatistiky charakteristík, ako je priemer alebo rozptyl. Odľahlé hodnoty nie sú vždy spôsobené nestabilitou prítomnou v meranom procese alebo výskytom chýb. Niekedy existujú objektívne technicko-technologické príčiny týchto odľahlých hodnôt. Napríklad pri výrobe odliatkov niektoré odliatky, ktoré sú kontrolované automatickou váhou so záznamníkom hodnôt môžu mať inú hmotnosť napríklad posledný odliaty kus, na ktorý neostalo dostatok materiálu.

Ak majú údaje dokonale normálne rozdelenie, čiara mediánu sa nachádza v strede krabicového diagramu a medián a priemer majú rovnakú hodnotu. V prípade rozdelení iných ako normálne rozdelenie sa často stáva, že medián a priemer sa významne odlišujú. Ak je čiara mediánu blízko jedného z kvartilov x_{25} alebo x_{75} , môže to naznačovať, že údaje môžu mať iné ako normálne rozdelenie. Asymetria v meranom procese často súvisí s technológiou postupu a výroby. Napríklad pri niektorých výrobe odliatkov budú niektoré odliatky vzhľadom na nedostatok vstupného materiálu.

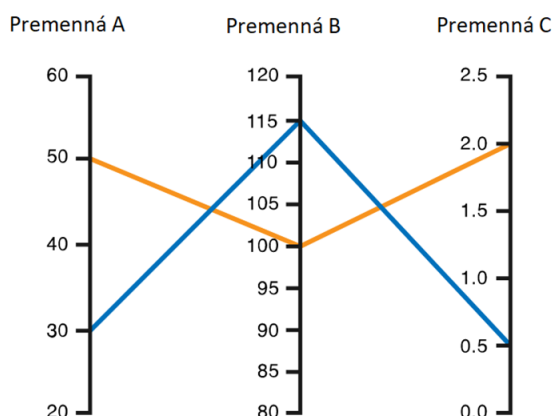
Pri spracovaní súboru množiny nameraných dát sa veľmi často počíta rozptyl. Zvyčajne sa tak získava štandardná odchýlka, ktorá je potrebná na ďalšie výpočty. Pomocou rozptylu je možné porovnať viacero procesov a rozhodnúť, ktorý z nich zodpovedá predpokladu o kvalite. Napríklad pri posudzovaní kvality je lepšie, keď má rozptyl menšiu hodnotu a priemer zodpovedá požadovanej hodnote kvality. Pomocou krabicového diagramu je možné graficky porovnať dva procesy a určiť, ktorý z nich má väčší rozptyl. Krabicový diagram je tak možné použiť na posúdenie kvality procesu.



Obrázok 22 Príklad krabicového diagramu

5.1.2.5 Paralelné súradnice

Tento typ vizualizácie sa používa na vykresľovanie viacrozmerných číselných spojitéch údajov (pozri nasledujúci obrázok). Paralelné súradnice ako typ súradnicového grafu sú ideálne na porovnanie mnohých premenných a sledovanie vzťahov medzi nimi. Napríklad pri porovnávaní produktov s mnohými premennými. V paralelnom súradnicovom grafe má každá premenná svoju vlastnú os a všetky osi sú umiestnené paralelne k sebe. Každá os môže mať inú mierku, pretože každá premenná pracuje s inou mernou jednotkou alebo môžu byť všetky osi normalizované, aby boli všetky mierky jednotné. Hodnoty sa vykresľujú ako séria čiar, ktoré sú spojené cez všetky osi. To znamená, že každá čiara je súborom bodov umiestnených na každej osi, ktoré boli spojené dohromady. Poradie, v akom sú osi usporiadané, môže ovplyvniť spôsob, akým používateľ chápe údaje. Jedným z dôvodov je, že vzťahy medzi susednými premennými sa vnímajú ľahšie, ako v prípade nesusedných premenných. Takže zmena poradia osí môže pomôcť pri objavovaní vzorov alebo korelácií medzi premennými. Nevýhodou paralelných súradnicových grafov je, že pri veľkom množstve údajov sa môžu stať neprehľadnými, a teda nečitateľnými.



Obrázok 23 Zobrazenie paralelných súradníc

5.2 Prototyp dátového modelu na zbieranie, analýzu a vyhodnocovanie symptómov pomocou analytických nástrojov s využitím umelej inteligencie

5.2.1 Dátová analytika pre zdravotníctvo

Algoritmy na detekciu anomálií pomáhajú pri mnohých aspektoch zdravotnej starostlivosti u ľudí, pričom medzi dve základné oblasti patrí diagnostika a monitorovanie pacientov.

Prakticky každá diagnostika je založená na údajoch (množine dát), ktoré poukazujú na abnormality v správaní pacienta alebo v jeho životných funkciách (ľahko pozorovateľné údaje charakterizujúce pacientov stav napríklad krvný tlak). V niektorých prípadoch je presná analýza kvantitatívnych údajov opisujúcich pacienta závažná a v takýchto prípadoch môže pomôcť v rámci celkovej účinnosti aplikácia algoritmov na detekciu anomálií.

Diagnózy

- **Údaje EKG a EEG:** - údaje elektrokardiogramu (EKG/EKG) sú časové rady, ktoré merajú elektrickú aktivitu srdca. Arytmie sú nepravidelnosti v údajoch elektrokardiogramu, napríklad predčasná komorová kontrakcia (PVC) je pomerne bežná srdcová arytmia, ktorú možno zistiť v údajoch EKG porovnaním s normálnymi údajmi EKG. Automatická detekcia PVC z údajov EKG výrazne znižuje pracovné zaťaženie kardiológov a potenciálne zvyšuje mieru detekcie PVC. Arytmie sa v súčasnosti zisťujú vizuálnou kontrolou, ktorú vykonáva odborný lekár, ktorý nemusí byť vždy dostupný, keď má pacient problémy so srdcom. Túto úlohu by potenciálne mohli

efektívnejšie vykonávať algoritmy na zisťovanie anomálií. Automatizácia môže tiež pomôcť znížiť potenciálne ľudské chyby pracovne vyťaženej lekárskej špecialistov. Podobným problémom je aj identifikácia možných dôkazov epileptických záchvatov u pacientov skúmaním údajov elektroencefalogramu (EEG) na abnormálne odchýlky. Nedostatok kvalifikovaných neuroológov naznačuje, že by mohlo byť užitočné použitie algoritmov na detekciu anomálií. V takýchto prípadoch by algoritmy na detekciu anomálií mohli slúžiť aspoň na účely triedenia: určitý počet falošne pozitívnych pacientov môže lekár tolerovať, pokiaľ je počet falošne negatívnych prípadov takmer nulový.

- **Diagnóza rakovinových ochorení** - klasifikácia nádorov ako benígnych a malígnych na základe rádiografických snímok je už dlho známa ako úloha, ktorá je obzvlášť náročná kvôli relatívne malému počtu malígnych prípadov. Na tento problém sa používa strojové učenie a klasifikačné algoritmy. Algoritmy na detekciu anomálií sú v tejto aplikácii veľmi užitočné, pretože rôzne malígne prípady nemusia mať veľa spoločných črt okrem toho, že sa ich atribúty výrazne líšia od benígnych prípadov. Najťažšie prípady sú samozrejme zhubné prípady, ktorých hodnoty atribútov sú neodlíšiteľné od benígnych prípadov; identifikácia potenciálnych znakov, ktoré by mohli pomôcť pri ich rozlišovaní, by bola užitočné pred použitím algoritmov klasifikácie alebo detekcie anomálií.

Monitorovanie pacientov

U pacientov, ktorí sa liečia na niektoré závažné zdravotné poruchy je veľmi dôležité neustále sledovať priebeh a životné funkcie, aby sa zistil výskyt neočakávaných vedľajších účinkov liekov alebo chirurgického zákroku, ktoré si v niektorých prípadoch vyžadujú okamžitú pozornosť a ďalšiu neodkladnú liečbu. Bohužiaľ, vzhľadom na veľký počet pacientov v nemocniciach sa príznaky abnormality niekedy náhodne ignorujú alebo sa im nevenuje okamžitá pozornosť. Podobne starší a zdravotne postihnutí ľudia občas utrpia pád v mieste svojho bydliska, pričom sa im nedostane okamžitej starostlivosti. Použitie algoritmov na detekciu anomálií na upozornenie poskytovateľov starostlivosti na základe vhodných senzorov je preto nevyhnutná.

Rádiológia

V oblasti rádiológie sa často vyhľadávajú neobvyklé údaje na röntgenových snímkach z magnetickej rezonancie. Algoritmy na detekciu anomálií môžu potenciálne pomôcť pri včasnom vyhľadávaní počiatočných fáz nádorov, čo uľahčuje včasné odhalenie rakoviny.

Epidemiológia

Vírusy a baktérie mutujú rýchlym tempom a pochopenie tohto procesu je nevyhnutné pre aspoň dočasný úspech v hľadaní novej generácie liekov. Oblasťami, ako je genetika, proteomika a metabolomika môžu pomôcť algoritmy na detekciu anomálií, ktoré sú schopné vyhľadávať nezvyčajné mutácie signalizujúce špecifické ochorenia. Lekárskej vede tiež pomáha vykonávanie identifikácie jednotlivých bodov v čase, v ktorých na základe epidemiologických údajov je odhalené, že predtým úspešné lieky prestávajú byť pre pacientov efektívne, čo znamená vznik mutácie príslušného patogénu odolného voči používanej generácii liekov. Niektoré takéto údaje možno získať aj od jednotlivých pacientov, ktorých reakcia na liek má neobvyklý priebeh napríklad najprv sa javí, že sa zdravotný stav pacienta zlepšuje, a potom sa rýchlo zhorší. To si môže vyžadovať opatrenia ako karanténne postupy na zabránenie epidemického šírenia nového patogénu rezistentného na lieky.

5.2.1.1 Detekcia anomálií

Anomália je "odchýlka od normy" - v tejto časti sa tento pojem skúma podrobnejšie. Mnohé vedecké a technické oblasti sú založené na predpoklade, že v prírode existujú procesy alebo správanie, ktoré sa riadia určitými pravidlami, alebo všeobecnými princípmi, ktorých výsledkom je stav systému, prejavujúci sa v pozorovateľných údajoch. Na základe údajov musíme formulovať hypotézy o povahe základného procesu, ktoré možno overiť na základe pozorovania ďalších údajov. Tieto hypotézy opisujú normálne správanie systému, pričom sa implicitne predpokladá, že údaje použité na vytvorenie hypotéz sú pre systém v určitom zmysle typické. V procesoch sa však môžu vyskytnúť odchýlky od normy, preto môžu systémy existovať aj v abnormálnych stavoch, čo vedie k pozorovateľným hodnotám údajov, ktoré sa líšia od hodnôt pozorovaných v prípade, že k takýmto odchýlkam procesu/stavu nedochádza. Úlohou detekcie anomálií je odhaliť odchýlky (od normy) v pozorovaných hodnotách údajov pri monitorovaní a

meraní základného procesu. Základným problémom je, že neexistuje jednoduchá a jednoznačná definícia, ktorá by nám umožnila vyhodnotiť, ako veľmi sú si dva dátové body podobné, a teda ako veľmi sa jeden dátový bod líši od ostatných v súbore (množine) údajov.

Pri použití algoritmov na detekciu anomálií je potrebné zvážiť tri možné prípady:

1. Korektná detekcia - zistené anomálie v údajoch presne zodpovedajú anomáliám v procese.
2. Falošná pozitivita - proces je naďalej normálny, ale pozorujú sa neočakávané hodnoty údajov, napríklad v dôsledku vnútorného šumu systému.
3. Falošná negativita - proces sa stane abnormálnym, ale dôsledky sa nezaznamenajú formou abnormálnych údajov napríklad v dôsledku toho, že signál abnormality nie je dostatočne silný v porovnaní so šumom v systéme.

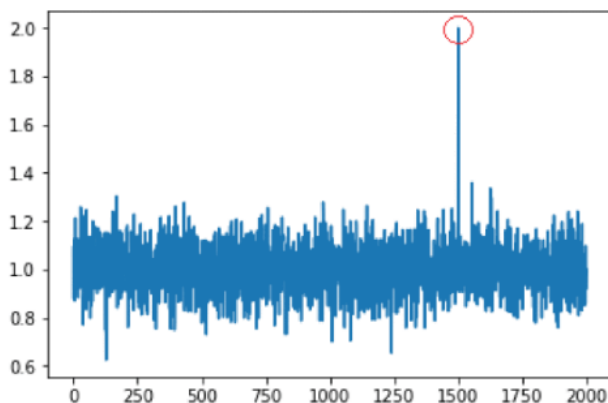
Väčšina reálnych systémov je taká, že dokonale korektná detekcia je nemožná. Úlohou dátovej analýzy je zvažovať aj túto skutočnosť a navrhnúť mechanizmy na minimalizáciu falošne pozitívnych aj falošne negatívnych výsledkov, prípadne povoliť viac jedného ako druhého z dôvodu úvah o asymetrických nákladoch s nimi spojených. Aby sme zohľadnili výskyt falošne pozitívnych a falošne negatívnych výsledkov, môžeme úlohu detekcie anomálií stanoviť ako úlohu odhadu pravdepodobnosti, že daný bod je anomáliou a nie ako úlohu klasifikácie, či daný bod je anomálny alebo nie. Ďalším často používaným hľadiskom je hodnotenie relatívnej anomálnosti rôznych bodov: môžeme povedať, že bod p je anomálnejší ako iný bod q , ale nemusíme si byť istí, či je niektorý z nich skutočnou anomáliou. Na riešenie tejto úlohy je možné použiť algoritmus na identifikáciu desiatich dátových bodov, zoradených podľa poradia, ako najpravdepodobnejších anomálnych prípadov. Týchto desať prípadov poskytuje východiskový bod pre ľudskú analýzu alebo zásah, hoci sa môže ukázať, že všetkých desať prípadov je prejavom prijateľného správania.

V kontexte kybernetickej bezpečnosti musia algoritmy na detekciu anomálií zohľadňovať skutočnosť, že procesy, ktoré sú predmetom záujmu, často nie sú deterministické ani úplne náhodné. Ťažkosti, ktoré sa vyskytujú v aplikáciách kybernetickej bezpečnosti možno totiž často pripísať ľudským aktérom s určitou slobodnou vôľou. Inými slovami, pozorovateľné správanie je výsledkom zámerného (nie náhodného) konania ľudí, ktorí nie sú predvídateľní, pretože ich zámery sú neznáme a ich plány sa môžu neočakávaným spôsobom zmeniť. V najlepšom prípade môže vykonanie niektorých pozorovaní v priebehu času odhaliť vzorec, od ktorého možno očakávať, že bude pokračovať, čo vedie k určitej predvídateľnosti procesu. Samotný ľudský zámer alebo plán, ktorý tento vzorec spôsobuje, sa môže náhle zmeniť, najmä v reakcii na mechanizmy kybernetickej obrany umiestnené na prekáženie predchádzajúcich plánov.

5.2.1.2 Kategorizácia anomálií

Bodové anomálie - jeden bod alebo hodnota meraného procesu je anomálna, ak sa významne líši od ostatných.

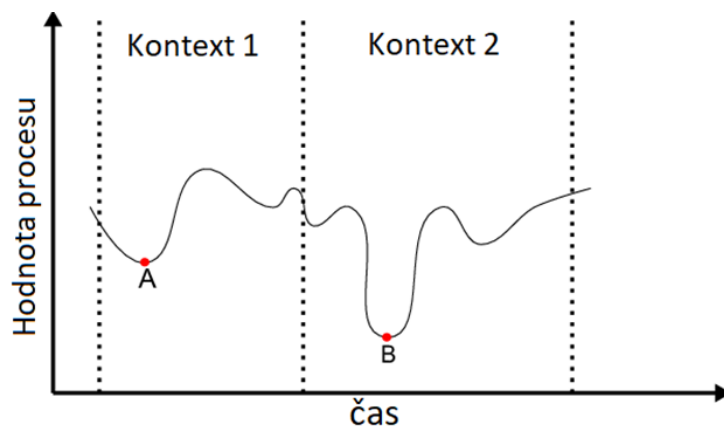
Obchodný prípad použitia - odhaľovanie podvodov s kreditnými kartami na základe podozrivu vysokej hodnoty jednorazového výberu finančnej hotovosti alebo abnormálne vysoká hodnota spotreby elektrickej energie v zaznamenanom časovom rade.



Obrázok 24 Príklad abnormálnej hodnoty v dátovom sete

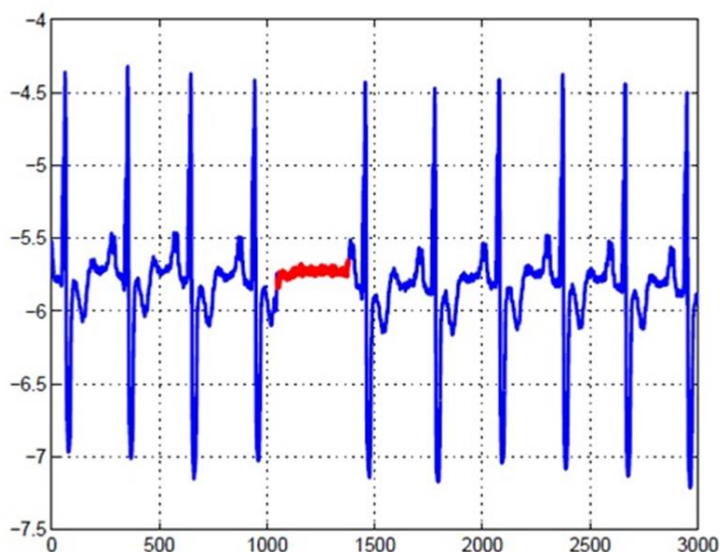
Kontextové anomálie - anomália je špecifická pre daný kontext. Tento typ anomálie je bežný najmä v časových radoch s prítomnou výraznou sezónnosťou alebo periodicitou.

Prípád použitia v podnikaní v praxi - výpadok výroby vo veľkom elektrárenskcom bloku alebo pokles spotreby v strede týždňa vplyvom štátneho sviatku.



Obrázok 25 Príklad kontextovej anomálie

Kolektívne (skupinové) anomálie - ak je súbor príbuzných prípadov údajov anomálny vzhľadom na celý súbor údajov, označuje sa ako kolektívna anomália. Jednotlivé inštancie údajov v kolektívnej anomálii nemusia byť anomáliami same osebe, ale ich spoločný výskyt ako kolekcia je anomálny.



Obrázok 26 Príklad skupinovej anomálie

5.2.1.3 Anomálie v jednorozmernom priestore

Kvantitatívne údaje sa často analyzujú výpočtom štatistík opisujúcich ich rozdelenie. Pod pojmom jednorozmerné kvantitatívne údaje rozumieme, že každý údajový bod je jedno číslo.

K najjednoduchšími možnými pravdepodobnostným rozdeleniam patria:

- Rovnomerné rozdelenie - keď sú údaje rovnomerne rozdelené v konečnom rozsahu, priemer a štandardná odchýlka len charakterizujú rozsah hodnôt. Ak je okolie ľubovoľného dátového bodu rovnako bohaté obsadené ako ktorýkoľvek iný bod, možno tvrdiť, že neexistujú žiadne anomálne dátové body, aj keď krajné dolné a horné hranice rozsahu sú relatívne ďaleko od priemeru. Jedným z možných náznakov anomálneho správania by mohlo byť, že malé okolie

obsahuje podstatne menej alebo viac dátových bodov, ako sa očakáva od rovnomerného rozdelenia.

- Normálne rozdelenie - keď sú údaje rozdelené normálne, hustota bodov sa podstatne znižuje, keď sa vzdalujeme od priemeru. Približne 0,1 % bodov je vzdialených od priemeru viac ako 3 (tri štandardné odchýlky) a len približne 5×10^{-8} % bodov je vzdialených od priemeru viac ako šesť štandardných odchýlok od priemeru. Preto sa často zvolí prahová hodnota (napríklad 3) a body, ktoré sú od priemeru vzdialené viac, sa vyhlásia za anomálne. Jeden z opačných pohľadov je, že existencia niektorých bodov vzdialených od priemeru je len dôsledkom toho, že premenná je normálne rozdelená. Môžeme teda tvrdiť, že súbor bodov (vzdialených od priemeru) je anomálny vtedy a len vtedy, ak je ich počet podstatne vyšší ako počet, ktorý by sa očakával, keby boli údaje normálne rozdelené, napríklad ak sa 2 % bodov údajov nachádzajú za hranicou 3.
- Unimodálne rozdelenia - mnohé unimodálne rozdelenia nie sú normálne, napríklad ak existuje striktná dolná hranica rozsahu hodnôt údajov. Medzi príklady patria log-normálne a gama rozdelenie. Podobne ako pri normálnom rozdelení, ak je známa povaha a vlastnosti rozdelenia, možno sa snažiť nájsť prahové hodnoty, za ktorými sa nachádza relatívne malý počet (napríklad 1 %) dátových bodov. opäť môžeme tvrdiť, že súbor bodov (v malej oblasti dátového priestoru) je anomálny, ak je ich počet väčší, ako predpovedá štatistika rozdelenia.
- Multimodálne rozdelenia - rozdelenia pre niektoré súbory údajov majú viacero módov, ktoré sa zistia až pri podrobnom skúmaní údajov. Heuristika, ako napríklad pravidlo 3, nie je pri takýchto rozdeleniach užitočná. Namiesto toho je užitočnejšie uvažovať o údajoch ako o súbore zhukov dátových bodov.

Zhlukovanie zohráva dôležitú úlohu pri detekcii anomálií. Body, ktoré nepatria do žiadneho zhuku, sa považujú za anomálne. Najmä body, ktoré sú vzdialené od susedných zhukov sú logickým výberom anomálií pre akýkoľvek algoritmus detekcie anomálií. Zostáva objasniť, kedy sa má súbor bodov považovať za zhuk a čo znamená, že bod sa považuje za dostatočne vzdialený od zhuku (alebo viacerých zhukov). Neformálna definícia zhuku je, že ide o súbor bodov, ktoré sú blízko navzájom, pričom úloha definovať pojem "blízko" zostáva otvorená a v praxi existuje viacero metodík na určenie tejto vzdialenosti.

1. Identifikácia zhukov na základe hustoty je veľmi populárna pri identifikácii anomálnych údajov. Ak je relatívny počet bodov (na jednotku vzdialenosti) v malej oblasti podstatne vyšší ako v celom súbore údajov, body v tejto oblasti možno považovať za zhuk. Stále sa však nejedná o striktnu matematickú definíciu, pretože výrazy "malá oblasť" a "podstatne vyšší" sú vo svojej podstate nejasné. Samotné rozdelenie hustoty sa dá analyzovať a (ak je unimodálne) môžeme určiť prah hustoty, po prekročení ktorého je hustota dostatočne vysoká na to, aby sa región považoval za zhuk. V praxi môže nastať situácia, že ten istý súbor údajov môže obsahovať jednu oblasť s vyššou hustotou a inú oblasť s nižšou hustotou, ktorá môže byť tiež považovaná za zhuk.
2. Iný pohľad je porovnávanie vnútrokupinových vzdialenosti s medziskupinovými vzdialenosťami a akceptovať, že body tvoria zhuk, ak sú podstatne bližšie k sebe navzájom (v priemere) ako k najbližším bodom mimo zhuku.

Nanešťastie existujú niektoré rozdelenia údajov, v ktorých táto definícia vedie k triviálnej identifikácii zhukov (napríklad umiestnenie takmer všetkých bodov do rovnakého zhuku).

Pri ktorejkoľvek z uvedených definícií si je potrebné uvedomiť, že niektoré body môžu ležať mimo zhukov. Niektoré zhukovacie algoritmy priradujú každý bod vždy do nejakého zhuku. Iné algoritmy predpokladajú, že počet zhukov je pevne stanovený (a vopred určený alebo poskytnutý používateľom algoritmu). Keď sa na detekciu anomálií používajú prístupy založené na zhukovaní, body vnútri zhukov s minimálnou veľkosťou sa zvyčajne nepovažujú za anomálie. Táto "minimálna veľkosť" je opäť externe špecifikovaný parameter, napríklad prahová hodnota založená na rozdelení veľkostí zhukov v súbore údajov.

Nie všetky body mimo zhukov sa musia považovať za rovnako anomálne. Jeden bod sa môže považovať za viac anomálny ako iný, ak je vzdialenejší od najbližšieho zhuku, pričom samotná vzdialenosť sa môže vyhodnocovať rôznymi spôsobmi (napríklad vzdialenosť k najbližšiemu bodu v zhuku alebo vzdialenosť k centroidu zhuku a podobne).

5.2.1.4 Anomálie vo viacrozmernom priestore

Z vyššie uvedeného vyplýva možnosť rozšíriť úvahu o výskyte anomálií vo viacrozmerných údajoch, pričom treba brať do úvahy, že zvyšujúca sa rozmernosť (dimenzionalita) spôsobuje ďalšie komplikácie. Základné problémy sa týkajú výberu miery vzdialenosti napríklad euklidovská miera vzdialenosti nemusí byť vhodná, pretože môže kombinovať čísla s rôznou dimenziou, ale ani voľba spoločného škálovacieho faktora nie je zrejmá pre všetky dimenzie. Napríklad pri pokuse vyhodnotiť, ktorý z dvoch bodov (mimo všetkých zhlukov) je najbližšie k zhluoku, môže o víťazovi rozhodnúť zvolená miera vzdialenosti. Často je potrebná normalizácia, napríklad lineárne škálovanie hodnôt údajov pozdĺž rôznych dimenzií tak, aby ležali v rovnakom rozsahu, ale vyvstáva otázka, aký druh normalizácie či škálovania dát je najlepší, pretože extrémne hodnoty môžu skresliť výsledky normalizačných prístupov.

Zaujímavá komplikácia nastáva, keď je jednou z relevantných dimenzií (alebo atribútov) čas, ktorému je potrebné priznať osobitný status.

Konkrétne údaje "časových radov" zahŕňajú tri atribúty: jeden je označenie (identifikujúce objekt), druhý označuje (absolútny) čas a tretí je číselná hodnota. Časové hodnoty pochádzajú z pevného a diskrétného rozsahu a zvyčajne sa predpokladá, že pre každú značku a každý prípustný časový bod existujú dátové body.

V tejto súvislosti vznikajú dva druhy problémov:

1. Ak existuje len jeden časový rad, t. j. všetky dátové body nesú informáciu o veľkosti meraného procesu, potom je relevantná otázka, či sa jedna časť časového radu podstatne líši od ostatných.
2. Môže existovať viac časových radov a je potrebné určiť, ktorý časový rad sa podstatne líši od väčšiny ostatných. V praxi sú to napríklad procesy, ktoré sa vyskytujú vo finančných aplikáciách, pričom často je potrebné zistiť, či jeden časový rad mení hodnoty v súlade s inými časovými radmi. Ak sa časový rad nemení takýmto spôsobom s inými časovými radmi, možno ho považovať za anomálny. Ak sa hodnoty údajov týkajú cien akcií/podielov, počet akcií v obehu dvoch spoločností sa môže líšiť, takže dve akcie môžu mať podobné zmeny v čase, aj keď sa ceny v danom okamihu môžu podstatne líšiť. Je potrebné navrhnúť algoritmy na určenie, či ide o takýto prípad, možno po lineárnej normalizácii všetkých cien akcií, aby boli v rovnakom rozsahu.

Pri vyhodnocovaní anomálií v správaní jednotlivcov alebo systémov možno vykonať tri druhy porovnaní:

1. Ako sa správanie jednotlivca v danom čase (alebo počas určitého časového obdobia) porovnáva s jeho vlastným správaním v minulosti? Vykazuje napríklad záznam o kreditnej karte človeka za jeden mesiac nákupy podstatne väčšieho rozsahu alebo nákupy inej kategórie ako v minulosti?
2. Ako sa správanie človeka porovnáva so správaním všetkých ostatných jednotlivcov, o ktorých sú k dispozícii údaje? Napríklad, stúpajú údaje o predaji automobilov istého predajcu aj vtedy, keď takmer všetky údaje o predaji automobilov v krajine klesajú?
3. Ako sa správanie jedného člena zhluoku alebo subpopulácie jednotlivcov porovnáva so správaním ostatných členov toho istého zhluoku?

5.2.1.5 Metódy detekcie anomálií

Detekcia anomálií zahŕňa metódy, na ktorých sú založené rôzne algoritmy a prístupy používané na v tejto oblasti napríklad v oblasti kybernetickej bezpečnosti. Základné prístupy možno charakterizovať ako prístupy založené na vzdialenosti, hustote a ranku.

- Na základe vzdialenosti - body, ktoré sú vzdialenejšie od ostatných, sa považujú za viac anomálne.
- Na základe hustoty - body, ktoré sa nachádzajú v oblastiach s relatívne nízkou hustotou, sa považujú za viac anomálne ako iné.
- Na základe ranku - kľúčovou ideou je použitie ranku namiesto vzdialenosti.

Pri každom z týchto prístupov môže byť charakter údajov s učiteľom, čiastočne s učiteľom alebo bez učiteľa. V prípade použitia algoritmov s učiteľom sú vopred označené anomálne hodnoty alebo údaje v

celej známej množine údajov a následná detekcia anomálií vychádza z tejto množiny s označenými anomáliami. V prípade použitia algoritmov bez učiteľa nie sú vopred označené anomálne hodnoty v množine údajov, takže detekcia si vyžaduje manuálne nastavenie použitého algoritmu odborným prístupom. V prípade, keď z celej množiny dostupných údajov má iba časť údajov vopred označené anomálne hodnoty, ide o detekciu s použitím algoritmov s čiastočným učiteľom. Napríklad môže byť k dispozícii niekoľko prípadov malvéru určitej novej kategórie a algoritmus s čiastočným učiteľom sa môže pokúsiť určiť, ktoré ďalšie prichádzajúce podozrivé prípady malvéru patria do rovnakej kategórie.

Praktické prístupy v oblasti detekcie anomálií často prebiehajú vo viacerých fázach, pričom v počiatočnej fáze sa neoznačeným údajom priradia predbežné značky. Algoritmus bez učiteľa na detekciu anomálií by mal spĺňať nasledujúce charakteristiky:

1. Normálne správanie musí byť dynamicky definované. Nie je potrebný žiadny predchádzajúci súbor trébovaných údajov alebo referenčný súbor údajov pre normálne správanie.
2. Odchýlky sa musia efektívne detegovať aj v prípade, že rozdelenie údajov nie je známe.
3. Algoritmus by sa mal dať prispôbiť rôznym charakteristikám oblasti, v ktorej je používaný; mal by byť použiteľný alebo modifikovateľný na detekciu odľahlých hodnôt v rôznych doménach bez toho, aby vyžadoval podstatné doménové znalosti.

5.2.1.6 Hodnotiace kritériá

Každý spôsob detekcie anomálií má iné vlastnosti a algoritmus, ktorý je úspešný pri riešení jedného problému, nemusí byť úspešný pri riešení iného problému. V skutočnosti je náročné charakterizovať optimálnu kvalitu detekcie a v tejto súvislosti možno položiť tri otázky:

1. Je možné navrhnúť kvantitatívne metriky, pri ktorých je možné jednoznačne povedať, ktorý z dvoch dátových bodov v danom súbore údajov je viac anomálny bez toho, aby bol potrebný zásah ľudskej intuície? Odpoveď na túto otázku je potrebná z dôvodu, aby sa minimalizoval počet falošne pozitívnych výsledkov generovaných algoritmi na detekciu anomálií, najmä v nekontrolovanom a čiastočne kontrolovanom kontexte.
2. Každý algoritmus detekcie anomálií odpovedá na uvedenú otázku procedurálnym spôsobom. Napríklad algoritmus k najbližším susedom vyžaduje zdať vstupný počet susedov. Dá sa táto implicitná voľba odôvodniť na matematickom alebo racionálnom základe?
3. V niektorých prípadoch sa algoritmy odvolávajú aj na snahu vypočítať výsledky v "rozumnom" čase, čo však vylučuje hľadanie optimálnych riešení. Môžeme v takýchto prípadoch povedať niečo o kvalite získaných riešení v porovnaní s optimálnymi riešeniami?

Teória informácií poskytuje možné odpovede na tieto otázky. Vysokoúrovňovým základom tohto prístupu je skutočnosť, že mnohé reálne procesy sa dajú stručne opísať z hľadiska ich podstaty. Z toho vyplýva, že variácia takéhoto stručného opisu procesu by sa mala považovať za anomáliu.

5.2.1.7 Analýzy hlavných komponentov

Analýza hlavných komponentov (anglicky PCA) nám umožňuje zhrnúť a vizualizovať informácie v súbore údajov, ktorý obsahuje osoby/pozorovania opísané viacerými vzájomne prepojenými kvantitatívnymi premennými. Každú premennú možno považovať za inú dimenziu. Ak máte v súbore údajov viac ako 3 premenné, mohlo by byť veľmi ťažké vizualizovať viacrozmerný hyperpriestor. Analýza hlavných komponentov sa používa na extrakciu dôležitých informácií z viacrozmernej tabuľky údajov a na vyjadrenie týchto informácií ako súboru niekoľkých nových premenných nazývaných hlavné komponenty. Tieto nové premenné zodpovedajú lineárnej kombinácii pôvodných premenných. Počet hlavných komponentov je menší alebo rovný počtu pôvodných premenných. Informácia v danom súbore údajov zodpovedá celkovej variabilite, ktorú obsahuje.

Cieľom analýzy je identifikovať smery (alebo hlavné komponenty), pozdĺž ktorých je variabilita údajov maximálna. Inými slovami, analýza hlavných komponentov redukuje dimenzionalitu viacrozmerných údajov na dva alebo tri hlavné komponenty, ktoré možno vizualizovať graficky s minimálnou stratou informácie.

Technicky povedané množstvo rozptylu, ktoré si zachováva každá hlavná zložka, sa meria pomocou tzv. vlastného čísla. Metóda analýzy hlavných komponentov je užitočná najmä vtedy, keď sú

premenné v rámci súboru údajov vysoko korelované. Korelácia naznačuje, že v údajoch existuje redundancia. Vzhľadom na túto nadbytočnosť možno analýzu použiť na redukciu pôvodných premenných na menší počet nových premenných, t. j. hlavných komponentov, ktoré vysvetľujú väčšinu rozptylu pôvodných premenných

5.2.1.8 *K-najbližších susedov*

Algoritmus *k*-najbližších susedov (anglicky *k*-nearest neighbors) jednoducho predpovedá novú vzorku pomocou *k*-najbližších vzoriek z množiny dát, na ktorej sa model trénoval. Na rozdiel od iných metód sa tento algoritmus nedá jednoducho interpretovať, ako napríklad model lineárnej regresie. Namiesto toho je jeho konštrukcia založená výlučne na jednotlivých vzorkách z množiny trénovacích údajov. Pri predikcii novej vzorky v rámci regresie algoritmus identifikuje vzorkovú skupinu najbližších susedov v predikčnom priestore. Predikovaná hodnota pre novú vzorku je potom priemerom odpovedí *k* susedov. Namiesto priemeru sa na predikciu novej vzorky môžu použiť aj iné súhrnné štatistiky napríklad medián. Základná metóda *k*-najbližších susedov, závisí od definície vzdialenosti medzi vzorkami. Najčastejšie používaná vzdialenosť je euklidovská vzdialenosť. Jej zobecnenie sa nazýva Minkovského vzdialenosť. Existuje mnoho ďalších metrických vzdialeností, napríklad Tanimotova, Hammingova a kosínusová, ktoré sú vhodnejšie pre špecifické typy prediktorov a v špecifických vedeckých kontextoch. Tanimotova vzdialenosť sa napríklad pravidelne používa v problémoch výpočtovej chémie, keď sa molekuly opisujú pomocou binárnych odtlačkov.

Keďže metóda *k*-najbližších susedov v podstate závisí od vzdialenosti medzi dátovými údajmi, rozpätie (variabilita) hodnôt premenných môže mať dramatický vplyv na vzdialenosti medzi jednotlivými pozorovaniami. Premenné, ktoré majú výrazne odlišnú variabilitu, budú generovať vzdialenosti, ktoré sú vážené smerom k premenným s najväčšou variabilitou. To znamená, že premenné s najväčšou variabilitou (rozpätím hodnôt) budú najviac prispievať k vzdialenosti medzi vzorkami. Aby sa predišlo tomuto potenciálnemu skresleniu a aby každá premenná prispela k výpočtu vzdialenosti rovnakou mierou, odporúča sa všetky premenné pred aplikáciou algoritmu *k*-najbližších susedov vycentrovať a preškálovať. Okrem problému škálovania môže byť použitie vzdialeností medzi vzorkami problematické, ak jedna alebo viac hodnôt premennej chýba, pretože potom nie je možné vypočítať vzdialenosť medzi vzorkami. V takom prípade existuje niekoľko možností riešenia. Po prvé, z analýzy možno vylúčiť dátové body zaťažené chýbajúcimi hodnotami alebo vynechať rovno celé premenné. Toto je najmenej žiaduca možnosť, avšak môže to byť jediná praktická voľba, ak sú chýbajúce hodnoty rozptýlené v množine dát. Ak vstupná premenná obsahuje dostatočné množstvo informácií aj napriek výskytu chýbajúcich údajov, alternatívnym prístupom je imputácia chýbajúcich údajov pomocou naivného odhadu, ako je priemer, medián či prístup použitia algoritmu *k*-najbližších susedov, ktorý používa len premenné s úplnými informáciami.

Metóda *k*-najbližších susedov môže mať slabú predikčnú výkonnosť, keď lokálna štruktúra vstupných premenných nie je objektívne relevantná pre tvorbu kvalitnej predikcie. Častou príčinou sú irelevantné alebo zašumené vstupné premenné, pretože tie môžu spôsobiť, že podobné vzory sa v predikčnom priestore od seba vzdalujú. Preto je odstránenie irelevantných, šumom zaťažených premenných kľúčovým krokom pred aplikáciou algoritmu *k*-najbližších susedov. Ďalším prístupom na zvýšenie predikčnej schopnosti je vyváženie príspevku jednotlivých susedov k predikcii novej vzorky na základe ich vzdialenosti od novej vzorky. V tomto variante sa trénovacie dáta, ktoré sú bližšie k novej vzorky, prispievajú k predpovedanej odpovedi viac, zatiaľ čo tie, ktoré sú vzdialenejšie, prispievajú k predpovedanej odpovedi menej.

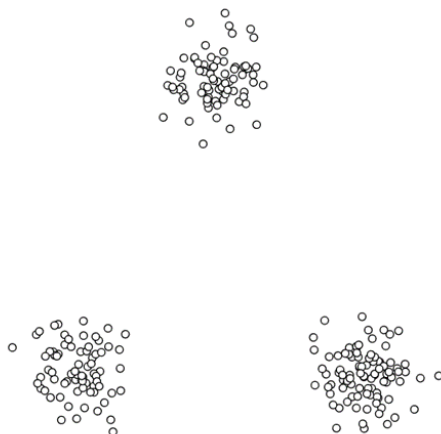
5.2.1.9 *Algoritmus K-means*

Algoritmus *k-means* je najpoužívanejší algoritmus nehierarchickej zhlukovej analýzy. Rieši problém zhlukovania, ktorý vytvára zhluky prvkov s podobnými vlastnosťami. Zhlukovanie sa podobá klasifikácii, ale na rozdiel od nej zaraďuje prvky do tried bez mien.

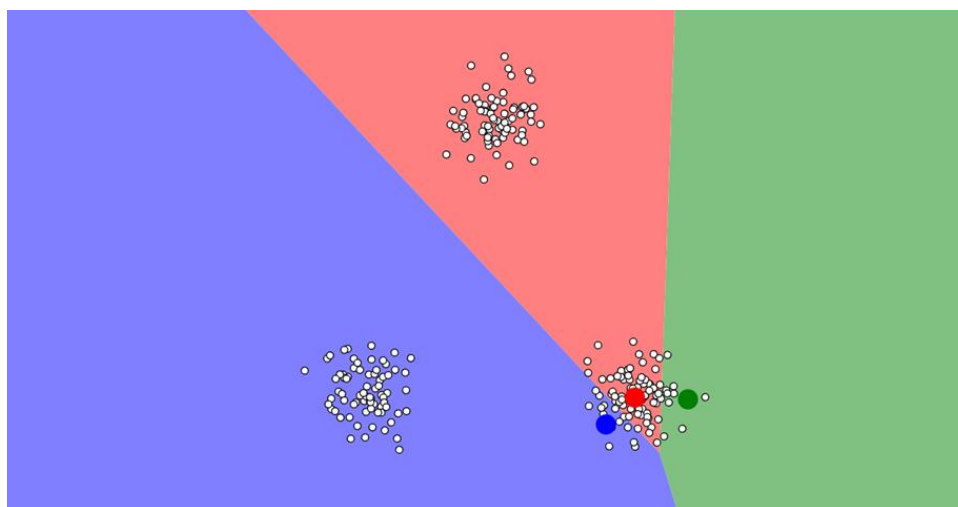
Algoritmus *k-means* (obr. X až Y) funguje v dvoch krokoch. Prvým je tzv. priraďovanie. Najprv sa do priestoru so zhlukmi pridá niekoľko centroidov, čo sú body definujúce jednotlivé zhluky, pričom sa predpokladá, že sa neskôr budú nachádzať v strede zhluku. Centroidy sú pridané na náhodné pozície a mal by ich byť rovnaký počet ako počet zhlukov. Centroidy sa pri krokoch algoritmu môžu pohybovať, ale dátové body musia zostať nehybné. Body sa zaraďia do zhluku, ktorého centroid je bodu najbližšie. Ďalším krokom je optimalizácia. V tomto kroku sa centroidy posunú do stredu oblasti, v ktorej sú umiestnené body ich kategórie. Následne sa im znova priradia najbližšie body. Poloha centroidu sa

znova prepočíta tak, aby tvoril ťažisko svojho zhluku. Tento proces sa opakuje, pokiaľ sa poloha centroidov neustáli. Po tomto procese sa centroidy z priestoru odstránia a vzniknú označené zhluky, ktorých body majú podobné vlastnosti.

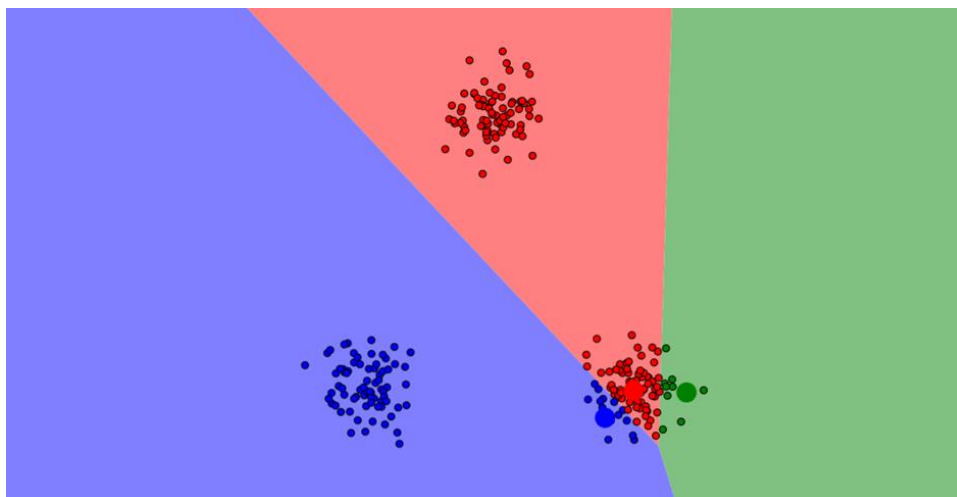
Grafická reprezentácia algoritmu K-means:



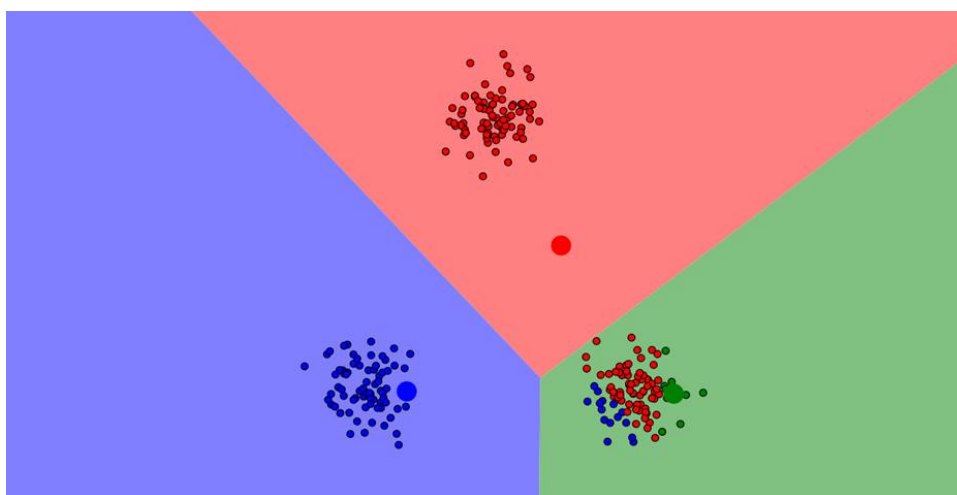
Obrázok 27 Krok 1



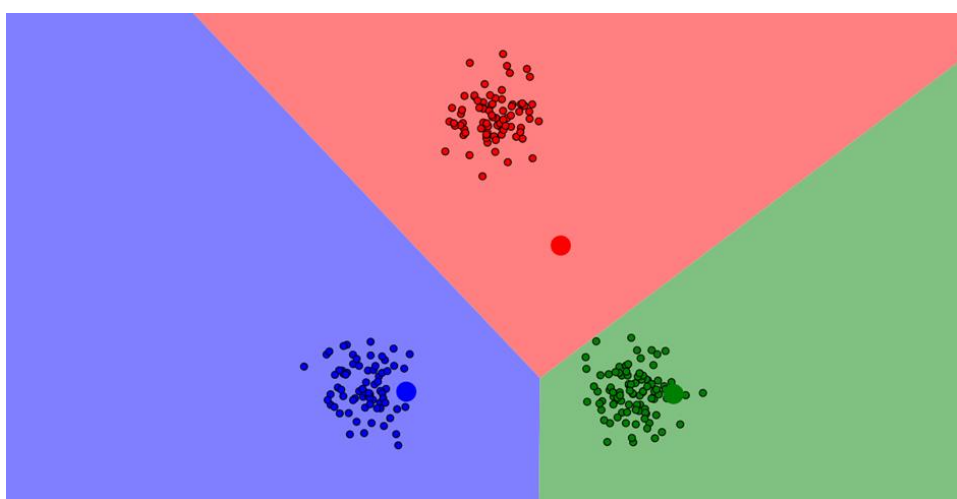
Obrázok 28 Krok 2



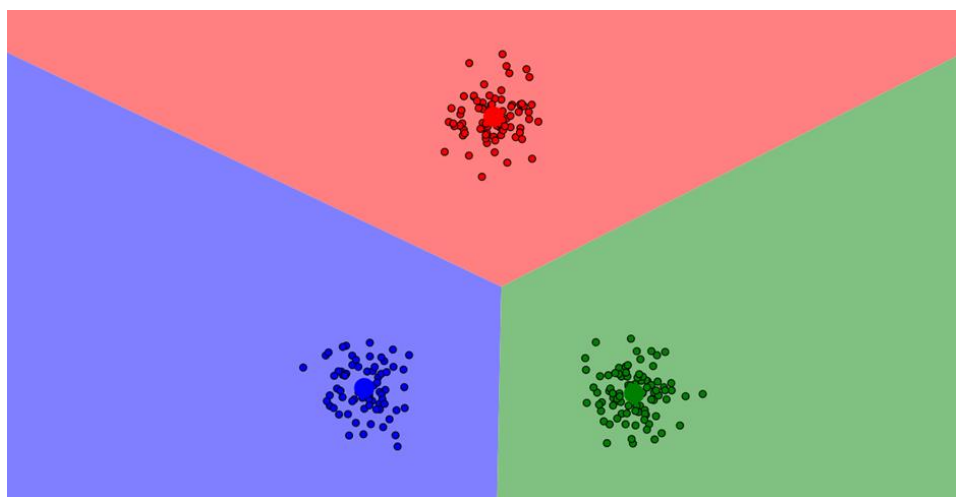
Obrázok 29 Krok 3



Obrázok 30 Krok 4



Obrázok 31 Krok 5



Obrázok 32 Krok 6

5.2.1.10 Podporné vektory

Podporné vektory (anglicky Support Vector Machines skratka SVM) sú triedou výkonných a veľmi flexibilných modelovacích algoritmov. Teória podporných vektorov bola pôvodne vyvinutá v kontexte klasifikačných modelov. Existuje niekoľko variantov regresie s podpornými vektormi, pričom táto podkapitola sa zameriava na jednu konkrétnu techniku nazývanú „ ϵ -insensitive“ regresia.

Pri klasickej lineárnej regresii je snaha nájsť také odhady parametrov, ktoré minimalizujú sumu štvorcových chýb. Jednou z nevýhod minimalizácie sumy štvorcových chýb je, že odhady parametrov môžu byť ovplyvnené výskytom len jedného odľahlého pozorovania, ktoré je vzdialené od celkového trendu v množine dát. Lineárna regresia je tak citlivá na prítomnosť odľahlých hodnôt.

Prítomnosť jednej odľahlej hodnoty môže výrazne ovplyvniť celkovú interpretovateľnosť výsledkov štatistického modelovania. Ak údaje obsahujú odľahlé pozorovania a anomálie, na nájdenie najlepších odhadov parametrov sa môže použiť alternatívna minimalizačná metrika, ktorá je menej citlivá napríklad Huberova funkcia.

Táto funkcia používa štvorcové rezíduá, keď sú "malé", a používa absolútne rezíduá, keď sú rezíduá veľké. Podporné vektory pri regresnom modelovaní používajú funkciu podobnú Huberovej funkcii s dôležitým rozdielom. Vzhľadom na prah nastavený používateľom algoritmu (označený ako ϵ), dátové body s rezíduami v rámci prahu neprispievajú k regresnej zhode, zatiaľ čo dátové body s absolútnym rozdielom väčším ako prah prispievajú lineárnou mierou. Tento prístup má niekoľko dôsledkov.

Po prvé, keďže sa nepoužívajú štvorcové rezíduá, veľké odľahlé hodnoty majú obmedzený vplyv na regresnú rovnicu. Po druhé vzorky, ktorým model dobre vyhovuje (t. j. rezíduá sú malé), nemajú žiadny vplyv na regresnú rovnicu. V skutočnosti, ak je prah nastavený na relatívne veľkú hodnotu, potom sú odľahlé hodnoty jedinými bodmi, ktoré definujú regresnú priamku. Je to trochu proti logickej intuícii: nepresne predpovedané body definujú priamku. Prax však ukázala, že tento prístup je pri definovaní modelu veľmi účinný.

5.2.1.11 DB scan

Zhlukovanie založené na hustote sa vzťahuje na metódy učenia bez učiteľa, ktoré identifikujú charakteristické skupiny/zhluky v údajoch na základe idey, že zhluk v dátovom priestore je súvislá oblasť s vysokou hustotou bodov, oddelená od iných takýchto zhlukov súvislými oblasťami s nízkou hustotou bodov.

Algoritmus, ktorý tvorí základ zhlukovania založeného na hustote sa nazýva DBSCAN (anglicky Density-Based Spatial Clustering of Applications with Noise). Dokáže objaviť zhluky rôznych tvarov a veľkostí vo veľkej dátovej množine, ktorá obsahuje rôzne šumy a odľahlé hodnoty.

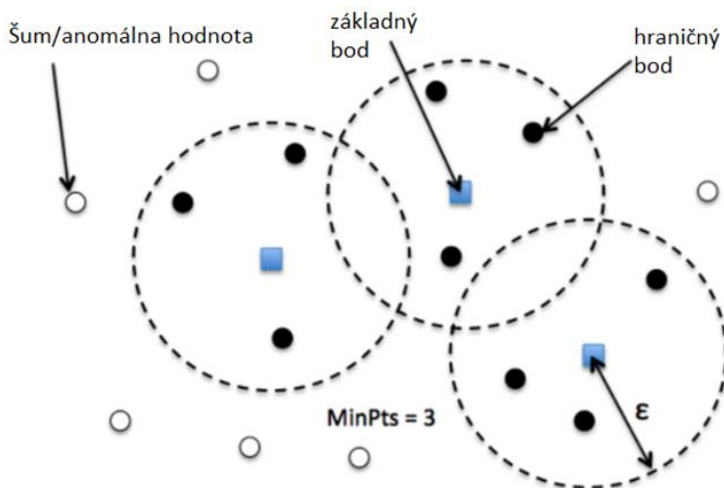
Algoritmus DBSCAN používa dva parametre:

- $minPts$ - minimálny počet bodov (prahová hodnota), ktoré sa majú zhlukovať, aby sa oblasť považovala za hustú,
- ϵ (ϵ) - miera vzdialenosti, ktorá sa použije na lokalizáciu bodov v okolí ktoréhokoľvek bodu.

Tieto parametre možno pochopiť po preskúmaní dvoch základných pojmov, a to „hustota dosiahnuteľnosti“ a „hustota spojitosti“. Dosiahnuteľnosť v zmysle hustoty stanovuje, že bod je dosiahnuteľný z iného bodu, ak od neho leží v určitej vzdialenosti. Táto vzdialenosť sa označuje ϵ .

Na druhej strane konektivita zahŕňa reťazový prístup založený na prechodnosti, ktorý určuje, či sa body nachádzajú v určitom zhluku. Napríklad body p a q by mohli byť prepojené, ak $p \rightarrow r \rightarrow s \rightarrow t \rightarrow q$, kde $a \rightarrow b$ znamená, že b je v susedstve a .

Po dokončení zhlukovania pomocou algoritmu DBSCAN existujú tri typy bodov:



Obrázok 33 Konceptuálna schéma algoritmu DBSCAN

- Základný bod (jadro) - je bod, ktorý má aspoň m bodov vo vzdialenosti n od seba.
- Hranica - je bod, ktorý má aspoň jeden jadrový bod vo vzdialenosti n .
- Šum - je bod, ktorý nie je ani jadrom, ani hranicou. A má menej ako m bodov vo vzdialenosti n od seba.

Algoritmické kroky pre zhlukovanie pomocou DBSCAN sú nasledovné:

1. Algoritmus postupuje tak, že ľubovoľne vyberie bod v súbore údajov (až kým sa nenavštívia všetky body).
2. Ak sa v okruhu nastavenej hodnoty ϵ od bodu nachádza aspoň minimálny počet bodov definovaných parametrom $minPts$, potom všetky tieto body sú považované za súčasť toho istého zhluku.
3. Zhluky sa potom rozšíria rekurzívnym opakovaním výpočtu susedstva pre každý susedný bod.

Odhad parametrov algoritmu DBSCAN

Pri aplikáciách nielen zhlukovacích, ale aj obecných modelov strojového učenia je vždy výzvou optimalizácia vstupných parametrov vlastných jednotlivým modelom. Každý parameter ovplyvňuje výsledok modelovania v rôznej miere.

Pre DBSCAN sú potrebné parametre ϵ a $minPts$:

- **$minPts$** - minimálnu hodnotu $minPts$ možno spravidla odvodiť z počtu dimenzií D v súbore údajov, keďže $minPts \geq D + 1$. Nízka hodnota $minPts = 1$ nemá zmysel, pretože potom už každý bod sám o sebe bude zhlukom. Pri $minPts \leq 2$ bude výsledok rovnaký ako pri hierarchickom zhlukovaní s metrikou jediného spojenia, pričom dendrogram bude orezaný na výške ϵ . Preto je potrebné zvoliť $minPts$ aspoň 3. Väčšie hodnoty sú však zvyčajne lepšie pre súbory údajov so šumom a prinesú významnejšie zhluky. Ako pravidlo možno použiť $minPts = 2 \cdot \dim$, ale v

prípade veľmi veľkých údajov, údajov so šumom alebo údajov, ktoré obsahujú veľa duplikátov, môže byť potrebné zvoliť väčšie hodnoty.

- ϵ - hodnota ϵ sa potom môže zvoliť pomocou grafického zobrazenia algoritmu k -najbližších susedov, v ktorom sa vykreslí vzdialenosť $k = \text{minPts}-1$ najbližšieho suseda zoradená od najväčšej po najmenšiu hodnotu. Dobré hodnoty ϵ sú tam, kde tento graf ukazuje "zlom": ak je ϵ zvolené príliš malé, veľká časť údajov nebude zhľukovaná, zatiaľ čo pri príliš vysokej hodnote ϵ sa zhľuky zlúčia a väčšina objektov bude v tom istom zhľuku. Vo všeobecnosti sú vhodnejšie malé hodnoty ϵ a spravidla by sa do tejto vzdialenosti mala nachádzať len malá časť bodov.
- **Funkcia vzdialenosti** - výber funkcie vzdialenosti je úzko spojený s výberom ϵ a má veľký vplyv na výsledky. Vo všeobecnosti je potrebné najprv určiť primeranú mieru podobnosti pre súbor údajov a až potom bude možné zvoliť parameter ϵ . Pre tento parameter neexistuje žiadny odhad, ale funkcie vzdialenosti sa musia zvoliť vhodne pre súbor údajov.

5.2.2 Posúdenie dynamiky predikcie zdravotného stavu analýzou a grafickým zobrazením diferencií prvého rádu

Analýza experimentálnych a výskumných údajov, ktoré boli pozorované v rôznych časových okamihoch vedie k novým a jedinečným výzvam v štatistickom modelovaní. Zjavná korelácia, ktorú zavádza výber vzoriek zo susedných bodov v čase, môže výrazne obmedziť použiteľnosť mnohých konvenčných štatistických metód, ktoré tradične závisia od predpokladu, že tieto susedné pozorovania sú nezávislé a identicky rozdelené. Systematický prístup, ktorým sa riešia na matematické a štatistické otázky, ktoré tieto časové korelácie vyvolávajú sa bežne označuje ako analýza časových radov. Vplyv analýzy časových radov na vedecké aplikácie možno čiastočne dokumentovať vytvorením skráteného zoznamu rôznych oblastí, v ktorých sa môžu vyskytnúť dôležité výzvy pre analýzu časových radov. Napríklad mnohé známe časové rady sa vyskytujú v oblasti ekonómie, kde sa spracovávajú denné burzové kurzy alebo mesačné údaje o nezamestnanosti. Sociológovia sledujú populačné rady, ako je napríklad pôrodnosť alebo počet žiakov v školách. Epidemiológovia môžu zaujímať počet prípadov chrípky pozorovaných za určité časové obdobie.

V medicíne by merania krvného tlaku sledované v priebehu času mohli byť užitočné pri hodnotení liekov používaných pri liečbe hypertenzie. Funkčná magnetická rezonancia zobrazujúca časové rady mozgových vln by sa mohla použiť na štúdium toho ako mozog reaguje na určité podnety za rôznych experimentálnych podmienok. Mnohé z najintenzívnejších a najsofistikovanejších aplikácií metód časových radov sa tiež týkajú problémov vo fyzikálnych a environmentálnych vedách. Táto skutočnosť vysvetľuje základný inžiniersky prístup, ktorý preniká do jazyka analýzy časových radov. Jedným z prvých zaznamenaných radov je mesačný počet slnečných škvŕn, ktorý skúmal Schuster (1906). Modernejšie výskumy sa môžu sústrediť na to, či je v meraniach globálnej teploty prítomné otepľovanie alebo či úroveň znečistenia môže ovplyvňovať dennú úmrtnosť v Los Angeles. Spoločné črty v charakteristike časového radu, známe ako výkonové spektrum sa používa na pomoc pri rozpoznávaní a preklade reči.

Geofyzikálne časové rady, napríklad tie, ktoré vznikajú pri ročných depozíciách rôzneho druhu, môžu poskytnúť dlhodobé charakteristiky pre analýzu teplôt a zrážok. Seizmické záznamy môžu pomôcť pri mapovaní zlomových líní alebo pri rozlišovaní zemetrasení a jadrových výbuchov. Uvedené časové rady sú len príkladmi experimentálnych databáz, ktoré možno použiť na ilustráciu procesu, na ktorý možno aplikovať klasickú štatistickú metodológiu v rámci korelovaných časových radov. Prvý krok pri každom skúmaní časových radov vždy zahŕňa dôkladné preskúmanie zaznamenaných údajov vynesenej v čase. Toto skúmanie často naznačuje metódu analýzy ako aj štatistiku, ktorá bude užitočná pri sumarizácii informácií v údajoch. Tiež je vhodné spomenúť, že existujú dva samostatné, ale nie nevyhnutne sa vylučujúce prístupy k analýze časových radov, ktoré sa bežne označujú ako prístup z pohľadu časovej domény a prístup z pohľadu frekvenčnej domény. Prístup v časovej doméne je vo všeobecnosti motivovaný predpokladom, že korelácia medzi susednými bodmi v čase sa najlepšie vysvetľuje v zmysle závislosti aktuálnej hodnoty od minulých hodnôt. Prístup v časovej doméne sa zameriava na modelovanie určitej budúcej hodnoty časového radu ako parametrickej funkcie súčasných a minulých hodnôt. V tomto scenári začíname s lineárnymi regresiami súčasnej hodnoty časového radu na jeho vlastných minulých hodnotách a na minulých hodnotách iných radov. Toto modelovanie vedie k tomu, že sa výsledky prístupu v časovej doméne používajú ako prognostický nástroj a sú obľúbené najmä v ekonomických vedách

5.2.2.1 Stacionarita a diferencovanie

Stacionarita - stacionárny časový rad je taký, ktorého štatistické vlastnosti ako priemer, rozptyl, autokorelácia atď. sú v čase konštantné. Väčšina metód štatistického prognózovania je založená na predpoklade, že časové rady môžu byť približne stacionárne (t. j. „stacionarizované“) pomocou rôznych matematických transformácií. Stacionarizovaná séria (časový rad) sa dá pomerne ľahko predpovedať, pretože sa predpovedajú štatistické charakteristiky, ktoré sa už vyskytli v minulosti. Predpovede pre stacionárne série môžu byť potom „netransformované“ obrátením akýchkoľvek matematických transformácií, ktoré boli predtým použité, na získanie predpovedí pre pôvodný rad. Preto nájdenie postupnosti transformácií potrebných na stacionarizáciu časového radu často poskytuje dôležité vodítko pri hľadaní vhodného predpovedného modelu. Stacionarizácia časových radov pomocou diferencovania (kde je to potrebné) je dôležitou súčasťou procesu prispôsobenia modelu ARIMA.

Ďalším dôvodom pre pokus o stacionarizáciu časového radu je možnosť získať zmyslupnú štatistiku vzorky, ako sú priemery, odchýlky a korelácie s inými premennými. Takáto štatistika je užitočnako deskripcia budúceho správania, iba ak je séria stacionárna. Ak sa napríklad časový rad v priebehu času neustále zvyšuje, priemer a rozptyl vzorky budú rásť s veľkosťou vzorky a v budúcich obdobiach budú vždy podhodnocovať priemer a rozptyl. A ak priemer a rozptyl radu nie je dobre definovaný, potom nie sú dobre definované ani jeho korelácie s inými premennými. Z tohto dôvodu treba byť opatrný pri pokuse extrapolovať regresné modely prispôsobené nestacionárnym údajom.

Väčšina obchodných a ekonomických časových radov nie sú stacionárne, ak sú vyjadrené v ich pôvodných merných jednotkách a dokonca aj po deflácií alebo sezónnom očistení budú zvyčajne stále vykazovať trendy, cykly, náhodné prechádzky a iné nestacionárne správanie. Ak má séria stabilný dlhodobý trend a má tendenciu vrátiť sa k trendovej línii po štrukturálnej zmene, môže byť možné stabilizovať ju odstránením trendu. Niekedy však ani eliminácia trendu nestačí na to, aby sa časový rad ustálil. V takom prípade môže byť potrebné transformovať ho na sériu rozdielov medzi jednotlivými obdobiami a/alebo sezónami. Ak priemer, rozptyl a autokorelácie pôvodných sérií nie sú konštantné v čase, dokonca ani po odstránení trendu, štatistiky zmien v časových radoch medzi obdobiami alebo medzi sezónami budú možno konštantné. O takejto sérii sa hovorí, že je rozdielovo stacionárna.

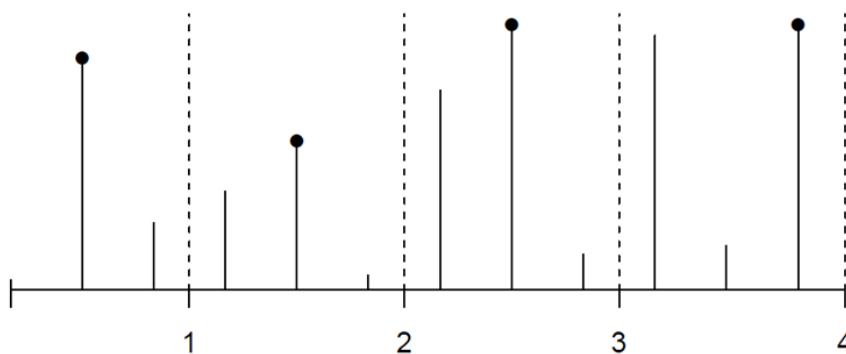
Prvá diferencia časového radu je séria zmien z jedného obdobia do druhého. Ak y_t označuje hodnotu časového radu y v perióde t , potom sa prvý rozdiel y v perióde t rovná $y_t - y_{t-1}$. Ak je prvá diferencia y stacionárna a tiež úplne náhodná (nie je prítomná autokorelácia), potom y ako časový rad je opísaný modelom náhodnej prechádzky, t. j. každá hodnota je o náhodný krok vzdialená od predchádzajúcej hodnoty. Ak je prvá diferencia y stacionárna, ale nie úplne náhodný, t. j. ak jej hodnota v období t autokoreluje s jej hodnotou v skorších obdobiach, potom môže byť vhodný sofistikovanejší model prognózy, ako je exponenciálne vyrovnávanie alebo ARIMA.

5.2.2.2 Anomálie v časových radoch

Klasická štatistika sa zjednodušene povedané, zameriava na priemerné správanie nejakého náhodného procesu, teda na javy realizujúce sa v okolí strednej hodnoty nejakého rozdelenia. Na druhú stranu, teória extrémnych hodnôt (ďalej len EVT, anglicky Extreme Value Theory) sa sústreďuje na extrémne, ojedinelé a vzácne javy, ktoré sa spravidla realizujú na chvostoch rozdelení. Z pohľadu štatistiky ide o tzv. odľahlé pozorovania. Ukazuje sa totiž, že pravdepodobnosť výskytu extrémnej udalosti je v skutočnosti vyššia, ako sa predpokladá na základe teoretického rozdelenia pravdepodobnosti, t. j., že rozdelenia majú ťažšie chvosty. Cieľom EVT je skúmanie správania týchto odľahlých pozorovaní. Záujem môže byť napríklad o stanovenie veľkosti extrémneho javu, ktorý nastane raz za zvolenú frekvenciu, prípadne naopak, stanovenie priemernej frekvencie výskytu extrémneho javu danej veľkosti a ďalšie. To znamená, že je potrebné nájsť vhodnú matematickú metódu, aby sme mohli vysvetliť okolnosti, ktoré sa vyskytujú s relatívne malou pravdepodobnosťou, no majú vysoký dopad. Najčastejšie sa na modelovanie týchto situácií využívajú dve metódy, a to metóda blokových maxím a metóda peaks-over-threshold (POT).

5.2.2.3 Metóda blokových maxím

Podstatou metódy blokových maxím je rozdelenie získaných pozorovaní do určitého počtu blokov, pričom v každom bloku sa zisťuje maximálne pozorovanie. Ideou potom je modelovanie samotných blokových pozorovaní. Nasledujúci obrázok ilustruje koncept metódy blokových maxím.

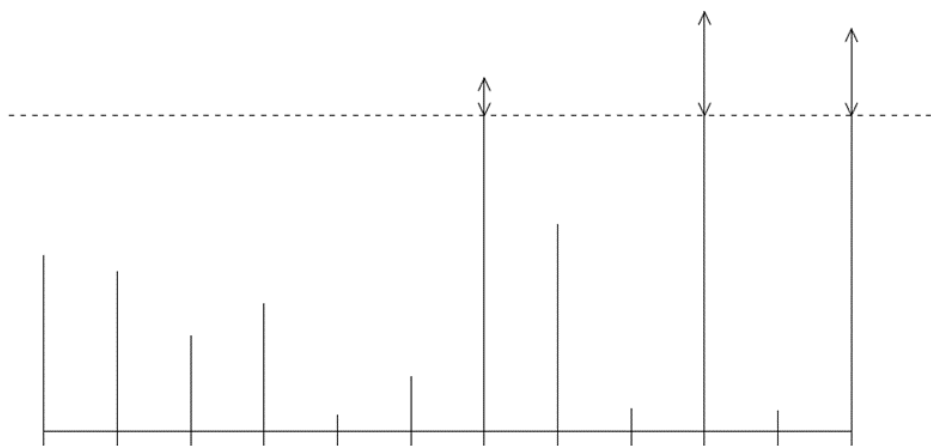


Obrázok 34 Príklad blokových maxím v dátovom sete

Pozorovania sú v tomto prípade rozdelené do štyroch blokov po troch pozorovaniach. Prístup k modelovaniu dát pomocou blokových maxím má však jednu zjavnú nevýhodu, v rámci bloku sa uvažuje len s jednou extrémnou hodnotou. Vo všeobecnosti môže nastať situácia, kedy druhá najvyššia hodnota daného bloku je vyššia ako maximum nejakého iného, čím v rámci analýzy sa strácajú cenné extrémne dáta.

5.2.2.4 Peaks-over-threshold

Spomenutý nedostatok metódy blokových maxím odstraňuje metóda peaks-over-threshold (POT). Je považovaná za modernejší spôsob analýzy extrémnych udalostí a je založená na prekročení určitej pevne zvolenej medze. Opäť sa nebudeme snažiť modelovať dané pozorovania. Namiesto toho sa budeme zaoberať prekročeniami pozorovaní (excesmi) vopred danej hranice, v prípade, že k tomuto prekročeniu dôjde. Tým sú dáta využité efektívnejšie ako v predchádzajúcom prípade. Na nasledujúcom obrázku je táto hranica vyobrazená prerušovanou čiarou.

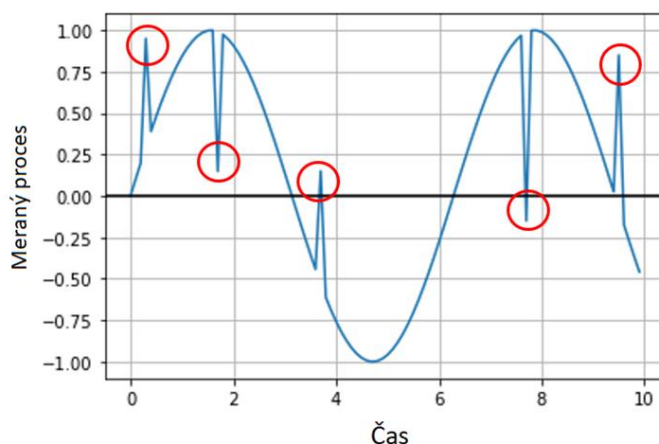


Obrázok 35 Príklad determinácie prahovej hodnoty v dátovom sete. Extrémne hodnoty sú všetky nad prerušovanou čiarou

5.2.3 Koncepcie detekcie anomálií a extrémnych hodnôt

V časových radoch možno anomáliu alebo odľahlý bod označiť ako dátový bod, ktorý nesleduje spoločný trend alebo sezónny či cyklický vzor celej množiny dát a výrazne sa odlišuje od ostatných hodnôt. Pod pojmom anomália alebo extrémna hodnota možno rozumieť štatistickú významnosť takéhoto správania, čo znamená, že štatistické charakteristiky anomálneho bodu nie sú v súlade so zvyškom časového radu.

Z nižšie uvedeného grafu časového radu je vidieť, že 5 dátových bodov, ktoré sa výrazne líšia od celkového priebehu je zvýraznených červeným kruhom. Teda týchto 5 anomálnych dátových bodov nesleduje celkový sínusový charakter časového radu, a preto ich možno označiť za anomálie časového radu.



Obrázok 36 Príklad anomálnych hodnôt odchyľujúcich sa od kontextu – štruktúrne narušený sínusový priebeh

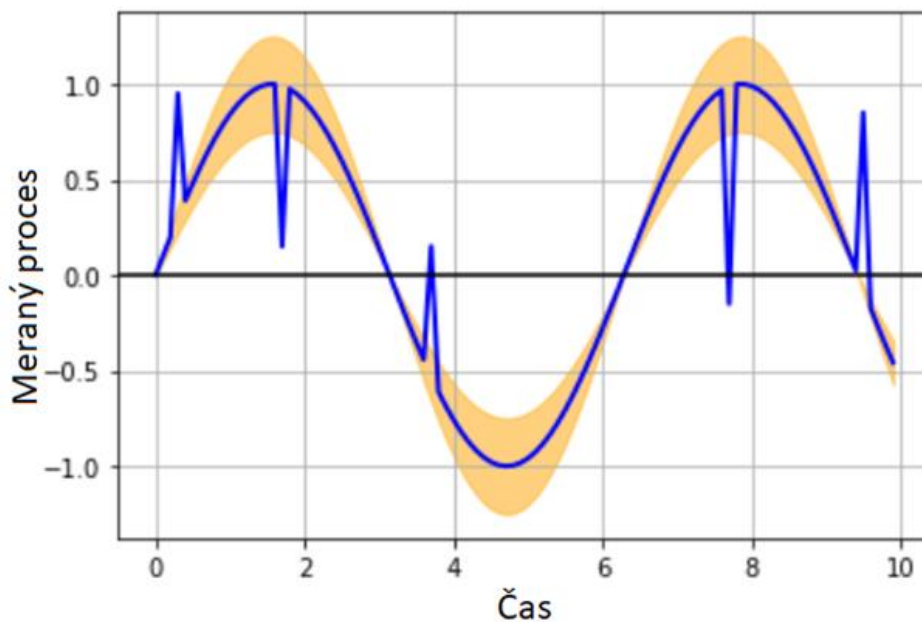
Detekcia a monitoring anomálií má v zdravotníctve výrazný vplyv na stratégiu a rozhodovacie procesy liečby, a preto je veľmi dôležité sledovať anomálie. Týka sa to samozrejme každého odvetvia a je naliehavo potrebné tieto anomálie neustále podrobne skúmať. Úloha dátovej analýzy sa preto stáva nielen kritickou, ale aj oblasťou, od ktorej sa očakáva tvorba nových prístupov na sledovanie, štúdium a analýzu anomálií v dátach a odvodenie zmysluplných informácií pre reálnu aplikáciu. Od predaja a marketingu, až po logistický reťazec a výrobu, každá fáza podnikania vyžaduje dostatok informácií, najmä o týchto anomáliách, aby mohla formovať svoje procesy a maximalizovať produktivitu a výsledky.

Z globálneho pohľadu je možné detekciu anomálií v časových radoch vykonať tromi hlavnými spôsobmi:

1. Prístupom, ktorý využíva predikčný interval spoľahlivosti.
2. Prístup štatistického profilovania.
3. Prístup založený na zhlukovaní bez učiteľa.

5.2.3.1 Využitie predikčného intervalu spoľahlivosti

Jedným zo spôsobov detekcie anomálií v časových radoch je vytvorenie prediktívneho modelu pomocou historických údajov s cieľom odhadnúť a získať predstavu o celkovom spoločnom trende, sezónnom alebo cyklickom vzore údajov časového radu a podobne. Pomocou prediktívneho modelu na predpovedanie budúcich hodnôt a na základe miery chyby (ktorú možno vypočítať pomocou MAPE - strednej absolútnej percentuálnej chyby) je možné stanoviť interval spoľahlivosti alebo pásmo spoľahlivosti pre predpovedané hodnoty a každý skutočný bod údajov, ktorý sa nachádza mimo tohto pásma spoľahlivosti, je potom považovaný za anomáliu (pozri nasledujúci obrázok).



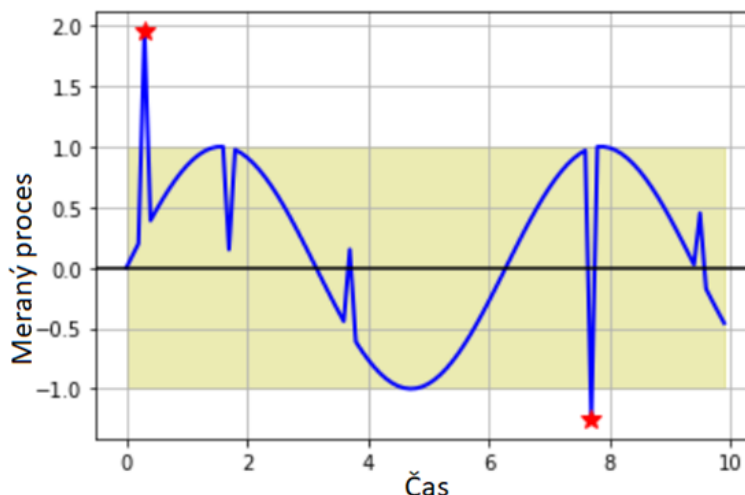
Obrázok 37 Príklad hodnôt vyskytujúcich sa mimo predikčného intervalu determinujúceho normálne a extrémne hodnoty

Na zostavenie predikčného modelu možno efektívne použiť aj populárne algoritmy modelovania časových radov, ako sú ARIMA, SARIMA, GARCH, VAR alebo akákoľvek regresia, prípadne algoritmus založený na strojovom učení a umelej inteligencii napríklad LSTM. Hlavnou výhodou tohto prístupu je nájdenie lokálnych odľahlých hodnôt. Nevýhodou je závislosť na účinnosti prediktívneho modelu.

5.2.3.2 Využitie štatistického profilovania

Tento prístup je pravdepodobne najviac využívaný najmä v oblasti ekonomiky a financií. Generovanie štatistického modelu alebo profilu daných údajov môže byť najrýchlejším a najužitočnejším prístupom, pretože táto metóda môže poskytnúť kontrolovanejšie a vysvetliteľnejšie výsledky. To možno vykonať výpočtom štatistických hodnôt, ako je priemer alebo medián kĺzavého priemeru historických údajov a použitím štandardnej odchýlky, aby sa dospelo k pásmu štatistických hodnôt, ktoré môžu definovať najhornejšiu hranicu a najspodnejšiu hranicu a všetko, čo sa nachádza mimo týchto pásiem, môže byť považované za anomáliu.

Tento prístup je veľmi praktický a môže byť vždy základným prístupom namiesto toho, aby sa používali rôzne komplexné a zložité metódy, ktoré si vyžadujú optimalizáciu parametrov a nemusia byť jednoducho interpretovateľné. Tento postup je veľmi účinný aj v prípade veľmi volatilných časových radov, pretože väčšina algoritmov predikčných modelov časových radov zlyháva, keď sú údaje veľmi volatilné. Hlavnou nevýhodou tohto prístupu je však detekcia lokálnych odľahlých hodnôt. Ako je vidno na nasledujúcom obrázku, z piatich zjavných bodov anomálie sa odhalili len 2 najvýznamnejšie body anomálie.

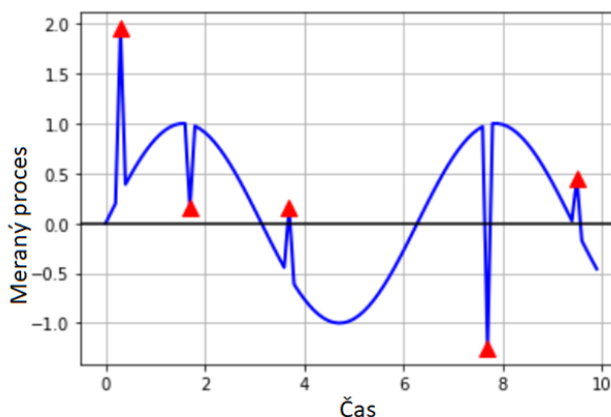


Obrázok 38 Príklad využitia štatistického profilovania

5.2.3.3 Využitie prístupu založeného na zhlukovaní bez učiteľa

Prístupy bez učiteľa sú mimoriadne užitočné na detekciu anomálií, pretože nevyžadujú žiadne označené údaje, ktoré by hovorili o tom, že určitý dátový bod je anomáliou. Zhlukovacie algoritmy môžu byť teda veľmi užitočné na detekciu anomálií v časových radoch. Jedným z častých úskalí zhlukovacích algoritmov na detekciu anomálií je definovanie počtu zhlukov, ktoré väčšina zhlukovacích algoritmov vyžaduje ako vstup zadaný používateľom algoritmu. Hoci existuje mnoho techník odhadu počtu zhlukov, pri časových radoch údajov nie je možné dynamicky odhadovať počet zhlukov pre každý rad. Vtedy sa prirodzenou voľbou stáva priestorové zhlukovanie aplikácií so šumom na základe hustoty (DBSCAN).

DBSCAN nevyžaduje žiadny vopred definovaný počet zhlukov a má len dva parametre (minimálny počet bodov v zhluku a epsilon, vzdialenosť medzi zhlukmi), takže sa veľmi ľahko ladí a je veľmi rýchly vo výkone. DBSCAN sa stáva najočividnejšou voľbou na detekciu anomálií kvôli týmto výhodám a nezoskupuje všetky dátové body do zhluku ako bežné techniky zhlukovania, ako napríklad *K*-Means. DBSCAN nezoskupuje anomálie alebo odľahlé dátové body do žiadneho zhluku, a preto sa veľmi ľahko používa. DBSCAN tiež pomáha mapovať "nový normál", čo sa väčšine ostatných prístupov nemusí podariť. DBSCAN má však aj niektoré nevýhody. Niektoré anomálne dátové body, ak sa opakujú mnohokrát v riedkom intervale, nemusia byť podľa DBSCAN zmapované ako anomálie. Takže v takomto prípade pomáha metóda DBSCAN založená na klízavom okne efektívnejšie mapovať tieto lokálne anomálie



Obrázok 39 Príklad narušenia kontextu meraného procesu extrémnymi hodnotami

5.2.4 Zhrnutie

V tejto kapitole boli popísané viaceré metódy pre analýzu zdravotníckych dát. V súčasnosti dátová veda poskytuje pokročilé algoritmy na analýzu, detekciu a grafické zobrazenie dát zo širokého spektra vedeckých oblastí. V zdravotníckych dátach dominuje záujem najmä o analýzu zhlukov a detekciu rôznych foriem analýzy. S narastajúcou presnosťou algoritmov narastá aj ich komplexnosť a náročnosť na výpočty. Napríklad tréning neurónových sietí môže trvať aj dni, pričom výsledok môže byť nepresvedčivý. Okrem toho komplexné algoritmy potrebujú byť prednastavené pomocou svojich vstupných parametrov užívateľom. Rovnako je dôležitá aj interpretácia výstupov, ktorá nie tak jednoduchá ako klasické štatistické modely. Moderné algoritmy môžu pri komplexných dátach, akými zdravotnícke dáta nepochybne sú, naplňať charakteristiky tzv. čiernej skrinky, kedy poznáme vstup a dostávame kvalitný výstup, nevieme však interpretovať alebo vysvetliť ako algoritmus dosiahol nami očakávanú kvalitu.

Nasadenie algoritmov si vyžaduje dlhodobý dohľad, prevádzkovú skúsenosť a pravidelné vyhodnocovanie výsledkov.

6 ZÁVER

V predkladanom dokumente sme predostreli výskum a vývoj, ktorý bol realizovaný v rámci tém 2. mílnika projektu Life Defender - Ochranca života.

V časti *Experimentálny vývoj prototypu sw platformy a cloudového úložiska* sme popisovali vytvorené prototypy pre zariadenia, ktoré budú výstupom projektu. Súčasťou každej časti je podrobný popis architektúry riešenia, dátového modelu a webového rozhrania.

Pre oblasť *Domácej karantény* bolo cieľom vývoja prototypu vytvorenie nástroja pre lekárov a pacientov, ktorý umožní vzdialené monitorovanie pacientov a vyhodnocovanie nazbieraných údajov z jednotného cloudového úložiska. Monitorovanie pacientov bude prebiehať použitím technických prostriedkov alebo prostredníctvom dotazníkov, ktoré slúžia k subjektívnemu zhodnoteniu stavu pacienta. Vývoj prototypu pozostával z viacerých súčastí ako Prototyp databázy výsledkov testov, Prototyp analýzy miery a priebehu testovaných symptómov pri testovanej osobe či Prototyp analýz a reportov samostatných symptómov – pandemická mapa. Technické riešenie SW prototypu spočívalo vo webovom portálovom riešení (primárne) pre lekára a v mobilnej aplikácii, ktorá má slúžiť hlavne pacientovi.

Pre oblasť *Automatickej testovacej bunky* bolo cieľom vývoja prototypu vyvinúť riešenie, ktoré ukladá a poskytuje dáta o otestovaných subjektoch do jednotného cloudového úložiska dát, a taktiež poskytnúť ovládacie prostredie pre dotykové obrazovky na automatickej testovacej bunke. Dôraz GUI pre automatickú testovaciu bunku bol kladený na jednoduchosť používateľského prostredia, čo najväčšiu možnú mieru automatizácie a urýchlenia procesov, a taktiež boli využité pokročilé SW prvky pre autentifikáciu testovanej osoby voči poskytnutému dokladu totožnosti. Vývoj prototypu SW pre automatickú testovaciu bunku pozostával zo súčastí ako - Preskúvanie možnosti implementácie overenia totožnosti testovaného subjektu oproti dokladu totožnosti cez technológiu spoločnosti Inovatics či vývoj Prototypu SW ovládacích prvkov pre automatickú testovaciu bunku. Technické riešenie SW prototypu pozostávalo z vytvorenia dotykového ovládania automatickej testovacej bunky prostredníctvom webového rozhrania upraveného pre používanie na veľkom dotykovom paneli.

Pre oblasť *Dezinfekčného robota* sme v rámci mílnika č. 2 vyvinuli riešenie, ktoré ukladá a poskytuje dáta o stave ovzdušia (hustota zamorenia konkrétnych častíc) v ľubovoľnom priestore, ako sú kancelárie, obchodné priestory, chodby, letiskové haly, továrne a podobne. Z webovej stránky aplikácie je možné na diaľku dané zariadenie ovládať a nariadiť mu niektoré z možných úkonov (Dezinfekcia UV, Dezinfekcia ozónom, Detekcia (statická), Detekcia (pohybová)). Informačný systém pre dezinfekčného robota sa skladá z troch kľúčových častí: cloudová služba; interný systém pre ovládanie dezinfekčného robota; mobilná aplikácia pre ovládanie dezinfekčného robota. Systém pre ovládanie dezinfekčného robota bunky je vnorený systém prevádzkovaný priamo v automatickej testovacej bunke. Primárnou úlohou systému je ovládanie dezinfekčného robota a synchronizácia nameraných dát do cloudovej služby. Mobilná aplikácia pre ovládanie dezinfekčného robota je mobilná aplikácia, pomocou ktorej je možné vzdialene ovládať dezinfekčného robota. V rámci danej kapitoly sa zaoberáme aj otázkou bezpečnosti, autorizácie a autentifikácie.

V časti *Experimentálny vývoj prototypu mobilnej aplikácie* sme popisovali vytvorené prototypy pre zariadenia, ktoré budú výstupom projektu. Súčasťou každej časti je podrobný popis architektúry riešenia, dátového modelu a webového rozhrania.

Úlohou mobilnej aplikácie pre domácu karanténu bolo synchronizovanie nameraných dát medzi sadou meracích zariadení a cloudovou službou. V našom prípade bola integrácia vykonaná voči zariadeniu Checkme. Mobilná aplikácia je navrhnutá tak, aby umožňovala získavanie nameraných dát aj z iných meracích zariadení, napr. z inteligentných hodín. Úlohou mobilnej aplikácie pre dezinfekčný robot je diaľkové ovládanie tohto zariadenia. Mobilná aplikácia obsahuje integráciu smerom na dezinfekčného robota. Úlohou integrácie je diaľkové ovládanie robota. Dezinfekčný robot obsahuje integráciu aj smerom na cloudovú službu, ktorej úlohou je synchronizácia dát.

V časti *Experimentálny vývoj prototypu modulu pokročilej analýzy a vizualizácie dát* sme popisovali výskum v oblasti Prototypu modulu vizualizácie dát a Prototypu dátového modelu na zbieranie, analýzu a vyhodnocovanie symptómov pomocou analytických nástrojov s využitím umelej inteligencie.

Pre oblasť *Prototypu modulu vizualizácie dát* sme navrhli a popísali možnosť použitia vizuálneho nástroja Kibana, ktorým je možné interaktívnym spôsobom prezerať dáta uložené v Elasticsearch. Kibana je určená pre administrátorov, dátových analytikov a biznis používateľov. Dokáže pracovať so

všetkými typmi dát. Štruktúrované dokumenty alebo neštruktúrovaný text, časové rady alebo číselné hodnoty, logy, priestorové geospatial dáta, metriky, či udalosti v čase. Kibana pomáha objaviť neznáme vzory a vzťahy a vizualizovať výsledok. Elasticsearch je distribuovaný analytický a vyhľadávací engine. Dokáže vyhľadávať a analyzovať takmer v reálnom čase nad rôznymi typmi dát, ako sú štruktúrované, neštruktúrované texty, hodnoty alebo priestorové geospatial dáta. Okrem jednoduchého vyhľadávania dokáže vykonávať zložitejšie agregácie objaviť tak rôzne trendy a vzory. Kibana vyžaduje vytvoriť pohľad na dáta (data view), čím inštruuje Elasticsearch, ku ktorým dátam chceme prísť a či dáta obsahujú časový údaj. Pohľad na dáta môže v Elasticsearch referencovať na viacero dátových tokov alebo indexy. V tejto kapitole sa venujeme aj zoznamu grafických metód, zobrazeniu spojitých premenných a ich charakteristikám.

Pre oblasť *Prototypu dátového modelu na zbieranie, analýzu a vyhodnocovanie symptómov* pomocou analytických nástrojov s využitím umelej inteligencie sme popísali viaceré metódy pre analýzu zdravotníckych dát. V súčasnosti dátová veda poskytuje pokročilé algoritmy na analýzu, detekciu a grafické zobrazenie dát zo širokého spektra vedeckých oblastí. V zdravotníckych dátach dominuje záujem najmä o analýzu zhlukov a detekciu rôznych foriem analýzy. S narastajúcou presnosťou algoritmov narastá aj ich komplexnosť a náročnosť na výpočty. Rovnako je dôležitá aj interpretácia výstupov, ktorá nie tak jednoduchá ako klasické štatistické modely. Nasadenie algoritmov si vyžaduje dlhodobý dohľad, prevádzkovú skúsenosť a pravidelné vyhodnocovanie výsledkov.

7 ZDROJE

Zdroje ku Kapitole 5:

- <https://smnd.sk/mcibula/alg/k-means.html>
- <https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html>
- https://datavizcatalogue.com/methods/parallel_coordinates.html
- <https://towardsdatascience.com/effective-approaches-for-time-series-anomaly-detection-9485b40077f1>
- KUHN, Max a Kjell JOHNSON. Applied Predictive Modeling [online]. New York, NY: Springer New York, 2013 [cit. 2022-06-08]. ISBN 978-1-4614-6848-6. Dostupné z: doi:10.1007/978-1-4614-6849-3

8 ZOZNAM OBRÁZKOV

Obrázok 1	Ukážka pozvánky – mail	11
Obrázok 2	Dátový model	13
Obrázok 3	Zoznam pacientov	14
Obrázok 4	Pozvať nového pacienta do aplikácie	15
Obrázok 5	Detail pacienta	15
Obrázok 6	Zoznam lekárov	16
Obrázok 7	Pridanie nového lekára do portálu	16
Obrázok 8	Detail lekára	17
Obrázok 9	Zoznam pozvánok	17
Obrázok 10	Zoznam zariadení	18
Obrázok 11	Pridať zariadenie	18
Obrázok 12	Logická architektúra systému	24
Obrázok 13	Docker kontajner vs Virtuálne servery (VM)	25
Obrázok 14	Logická architektúra systému	31
Obrázok 15	Docker kontajner vs Virtuálne servery (VM)	33
Obrázok 16	Logická architektúra systému	43
Obrázok 17	Docker kontajner vs Virtuálne servery (VM)	44
Obrázok 18	Logická architektúra pre systém domácej karantény	47
Obrázok 19	Logická architektúra pre systém automatickej dezinfekcie	49
Obrázok 20	Schéma plánovaného rozhrania	52
Obrázok 21	Príklad histogramu	53
Obrázok 22	Príklad krabicového diagramu	54
Obrázok 23	Zobrazenie paralelných súradníc	55
Obrázok 24	Príklad abnormálnej hodnoty v dátovom sete	57
Obrázok 25	Príklad kontextovej anomálie	58
Obrázok 26	Príklad skupinovej anomálie	58
Obrázok 27	Krok 1	63
Obrázok 28	Krok 2	63
Obrázok 29	Krok 3	64
Obrázok 30	Krok 4	64
Obrázok 31	Krok 5	64
Obrázok 32	Krok 6	65
Obrázok 33	Konceptuálna schéma algoritmu DBSCAN	66
Obrázok 34	Príklad blokových maxím v dátovom sete	69
Obrázok 35	Príklad determinácie prahovej hodnoty v dátovom sete. Extrémne hodnoty sú všetky nad prerušovanou čiarou	69
Obrázok 36	Príklad anomálnych hodnôt odchyľujúcich sa od kontextu – štrukturálne narušený sínusový priebeh	70
Obrázok 37	Príklad hodnôt vyskytujúcich sa mimo predikčného intervalu determinujúceho normálne a extrémne hodnoty	71
Obrázok 38	Príklad využitia štatistického profilovania	72
Obrázok 39	Príklad narušenia kontextu meraného procesu extrémnymi hodnotami	72

9 ZOZNAM TABULIEK

Tabuľka 1	Zoznam endpointov systému	23
Tabuľka 2	Zoznam integrácií.....	25
Tabuľka 3	Zoznam kontajnerov	26
Tabuľka 4	Zoznam prostredí.....	26
Tabuľka 5	Zoznam endpointov systému	31
Tabuľka 6	Zoznam integrácií.....	32
Tabuľka 7	Zoznam kontajnerov	33
Tabuľka 8	Zoznam prostredí.....	34
Tabuľka 9	Zoznam endpointov systému	42
Tabuľka 10	Zoznam integrácií.....	43
Tabuľka 11	Zoznam kontajnerov	44
Tabuľka 12	Zoznam prostredí.....	45
Tabuľka 13	Zoznam integrácií.....	48
Tabuľka 14	Zoznam integrácií.....	49