FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

ZADÁNÍ DIPLOMOVÉ PRÁCE

Název:	Webová a mobilní aplikace pro systém SVAR
Student:	Bc. Tomáš Bučko
Vedoucí:	Ing. David Buchtela, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce zimního semestru 2021/22

Pokyny pro vypracování

Cílem práce je pro spolek Česká civilní ochrana obyvatelstva vytvořit webovou a mobilní aplikaci pro zadávání bezpečnostních stavů SVAR (Systém včasného varování) pro jednotlivé kraje ČR a to oprávněnými osobami zadávat stavy za příslušný kraj (či kraje). Dalším cílem je vytvoření databáze aktuálních stavů a z ní generování HTML kódu (tabulky stavů).

1. Proveďte analýzu požadavků na mobilní i webovou část aplikace. Mobilní část musí min. obsahovat autorizaci osoby (ověření její role) a umožnit jí změnit bezpečnostní stav v krajích. Webová část musí min. umožnit totéž, co mobilní aplikace (frontend) a navíc umožnit správu (pro admina systému) databáze a generovat HTML kód z aktuálně zadaných stavů umístitelný do libovolné webové stránky.
2. Na základě analýzy požadavků navrhněte konceptuální model celého systému.
3. Pomocí vhodných technologií implementujte prototyp mobilní i webové části systému.
4. Prototyp řádně otestujte a zdokumentujte postup uživatelského použití systému.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 27. května 2020

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Diplomová práca

Webová a mobilná aplikácia pre systém SVAR

Bc. Tomáš Bučko

Katedra softvérového inžinierstva
Vedúci práce: Ing. David Buchtela, Ph.D.

26. novembra 2020

Prehlásenie

Prehlasujem, že som predloženú prácu vypracoval(a) samostatne a že som uviedol(uviedla) všetky informačné zdroje v súlade s Metodickým pokynom o etickej príprave vysokoškolských záverečných prác.

Beriem na vedomie, že sa na moju prácu vzťahujú práva a povinnosti vyplývajúce zo zákona č. 121/2000 Sb., autorského zákona, v znení neskorších predpisov, a skutočnosť, že České vysoké učení technické v Praze má právo na uzavrenie licenčnej zmluvy o použití tejto práce ako školského diela podľa § 60 odst. 1 autorského zákona.

V Prahe 26. novembra 2020 . . . . . .

Odkaz na túto prácu

Bučko, Tomáš. *0.0.0.* Diplomová práca. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.

Abstrakt

Predmetom tejto záverečnej práce je implementácia systému pre zadávanie bezpečnostných stavov SVAR (Systému včasného varovania). Analýza súčasného stavu rozoberá aktuálne fungujúci proces a jeho nedostatky. Nasleduje analýza požiadaviek a definovanie prípadov použitia, ktoré sú základom pre konceptuálny model systému. V kapitole implementácia je popísaný implementovaný systém a jeho používateľské rozhranie, nasledované integračnými a funkčnými testami.

Klíčová slova JWT, Kotlin, MySQL, Node.js, REST, Vue.js

Abstract

The subject of this final thesis is to implement a system used for setting alert states of SVAR (Early warning system). The AS-IS analysis discusses the current process and its shortcomings. The following is a requirement analysis and the definition of use cases that are the basis for the conceptual model of the system. The implementation chapter describes the implemented system and its user interface, followed by integration and functional tests.

Keywords JWT, Kotlin, MySQL, Node.js, REST, Vue.js

Obsah

Zoznam obrázkov

Úvod

Systém včasného varovania je proces s cieľom znížiť dopad udalostí ohrozujúcich bezpečnosť, či zdravie ľudí. Poskytuje čas na prípravu na nepriaznivú udalosť a čas na minimalizáciu jej dopadu.

V reakcií na zhoršujúcu sa bezpečnosť vo svete, v Európe a v Českej republike bol od 10. marca 2018 sprevádzkovaný SYSTÉM VČASNÉHO VAROVANIA (SVAR). Tento systém slúži na informovanie a varovanie pred udalosťami s významným dopadom na bezpečnosť a ohrozenie života či zdravia občanov Českej republiky a ich činnosť. [1]

Účelom zavedenia systému včasného varovania pre potreby bezpečnosti Českej republiky je prenos informácií o vzniku a vývoji mimoriadných situácií na úrovni regionálnej, celoštátnej a medzinárodnej. Zároveň slúži na aktiváciu operačných plánov v rámci krajov, jednotlivých obcí a miest, a započatiu činností uvedených v operačných plánoch. [1]

Kapitola 1

Cieľ práce

Finálnym produktom tejto práce má byť systém pre podporu Systému včasného varovania, ktorý sa skladá z troch častí.

Prvou a hlavnou časťou je **mobilná aplikácia**, ktorá oprávnenej osobe umožní zmeniť bezpečnostný stav regiónu, za ktorý zodpovedá. Ďalšou časťou je **webová aplikácia**. Má umožňovať to, čo mobilná aplikácia, a navyše umožniť administrátorovi správu relevantných častí databáze. Treťou časťou je **databáza**, ktorá dokáže vygenerovať tabuľku aktuálnych bezpečnostných stavov jednotlivých regiónov. Tabuľku je následne možné priamo vložiť do ľubovoľnej webovej stránky.

Práca si kladie za cieľ analyzovať súčasný stav a požiadavky na nový systém, na základe analýzy tento systém navrhnúť a implementovať použitím vhodných technológií, a následne ho riadne otestovať. Použitie systému má byť zdokumentované v používateľskej príručke.

Kapitola 2

Analýza súčasného stavu (AS-IS)

2.1 Popis súčasného stavu

V súčasnosti je Systém včasného varovania na internete riešený statickou HTML tabuľkou obsahujúcou bezpečnostné stavy jednotlivých sektorov. Tabuľka je zverejnená na portáli Civilnej Ochrany a OBRAny (COOBRA). Zmena bezpečnostného stavu prebieha tak, že osoba zodpovedná za daný sektor kontaktuje administrátora webu, ktorý následne v zdrojovom kóde upraví bezpečnostný stav sektoru. Takéto riešenie má niekoľko zrejmých nedostatkov.

2.2 Nedostatky súčasného stavu

2.2.1 Reakčný čas

Prvým problémom je **reakčný čas**. V prípade, že dôjde k udalosti ohrozujúcej bezpečnosť alebo zdravie občanov, zodpovedná osoba musí najprv kontaktovať administrátora portálu. V závislosti na forme komunikácie administrátor obdrží požiadavku na zmenu bezpečnostného stavu v rozmedzí niekoľkých minút až hodín. Administrátor v danom okamihu nemusí mať po ruke telefón alebo počítač. Môže sa napr. nachádzať v inom meste alebo štáte. V takom prípade môže reakčný čas stúpnuť na niekoľko dní.

2.2.2 Autorizácia osoby

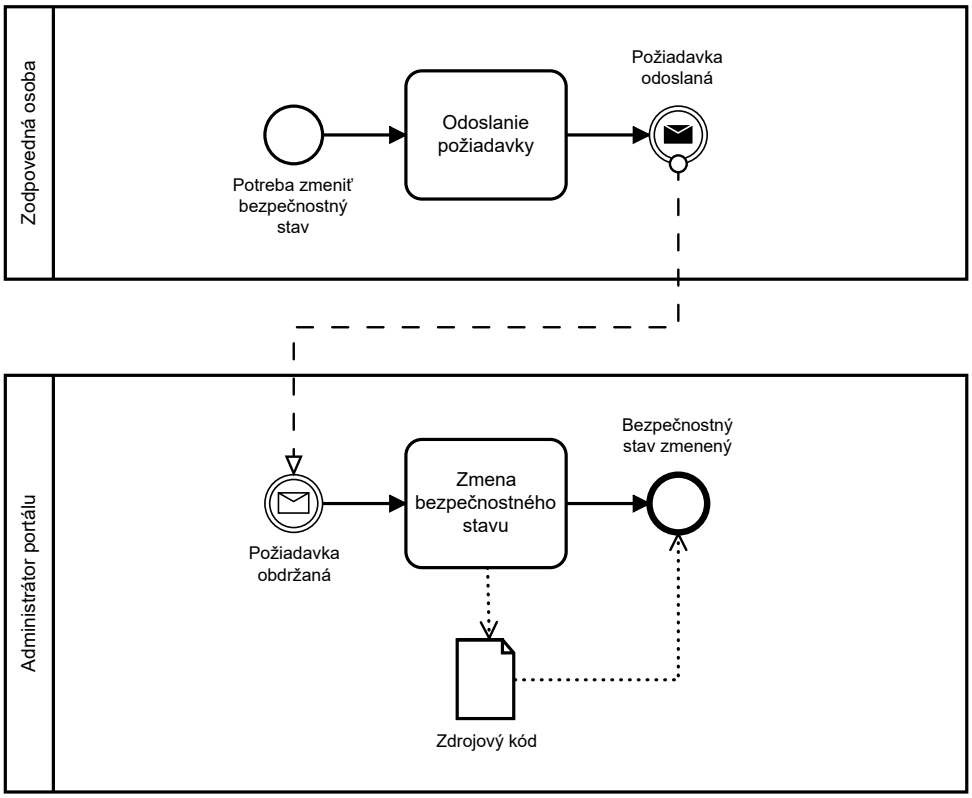
Ďalším problémom je **autorizácia osoby**. V prípade obdržania požiadavku na zmenu bezpečnostného stavu administrátor spolieha na to, že požiadavka skutočne pochádza od osoby, ktorá je oprávnená zadať bezpečnostný stav daného sektoru.

2.2.3 Single point of failure

Aj za ideálnych podmienok, kedy je reakčný čas minimalizovaný a autorizácia osoby zaručená, stále zostáva problém, že administrátor portálu predstavuje tzv. ***single point of failure***. Administrátor v celom procese reprezentuje časť, ktorá v prípade zlyhania zapríčiní zlyhanie celého procesu. Akonáhle administrátor z nejakého dôvodu nemôže upraviť danú časť v zdrojovom kóde, celý proces sa zastaví.

2.3 AS-IS proces

Súčasnú podobu procesu zadávania bezpečnostného stavu zachycuje nasledujúci diagram:



Obr. 2.1: Proces zadávania bezpečnostného stavu.

Kapitola 3

Analýza a návrh

3.1 Analýza požiadaviek

3.1.1 Funkčné požiadavky

Cieľom práce je vytvoriť mobilnú a webovú aplikáciu, ktorá bude komunikovať s databázou. Účelom mobilnej aplikácie je, aby zodpovedné osoby nemuseli meniť bezpečnostné stavy sektorov prostredníctvom tretej osoby (administrátora), ale aby to po prihlásení mohli vykonávať priamo. Účelom webovej aplikácie je, aby kopírovala funkcionalitu mobilnej aplikácie, a navyše umožňovala administrátorovi spravovať položky databázy. Databáza má okrem iného umožniť generovanie tabuľky sektorov a ich aktuálne priradených bezpečnostných stavov. Z toho plynie, že systém ako celok má umožňovať:

- Zaregistrovať sa
- Prihlásiť sa
- Priradiť osobu k sektoru
- Zmeniť bezpečnostný stav sektoru
- Vygenerovať tabuľku bezpečnostných stavov

3.1.2 Nefunkčné požiadavky

3.1.2.1 Požiadavky na bezpečnosť

Požiadavka správy položiek databázy administrátorom implikuje nutnosť rozlíšiť medzi bežným uživateľom a administrátorom, čo implikuje autorizáciu na základe rolí. Prístup k dátam implicitne predpokladá autentizáciu, takže v systéme bude implementovaný jeden zo štandardných spôsobov autentizácie používateľov.

3.1.2.2 Technologické požiadavky

Jediným vopred daným techologickým požiadavkom je implementácia databázy pomocou technológie *MySQL*.

3.1.2.3 Požiadavky na prevádzku

Česká republika má 14 krajov, čo implikuje maximálne 14 aktívnych používateľov a minimálne 1 administrátora. Predpokladaná vyťaženosť aplikáčného serveru bude tým pádom minimálna a load balancing nebude potrebný. Nebude nutná replikácia ani sharding databázy. Predpokladaná minimálna konfigurácia servera je dvojjadrový CPU, 4GB RAM a 120GB úložného priestoru na disku.

3.2 Prípady použitia

Analýza požiadaviek vedie na nasledujúce prípady použitia:

- UC01 – Zaregistrovať sa
- UC02 – Prihlásiť sa
- UC03 – Odhlásiť sa
- UC04 – Zobraziť prehľad regiónov
- UC05 – Zobraziť detail regiónov
- UC06 – Zmeniť bezpečnostný stav regiónu
- UC07 – Priradiť používateľa k regiónu
- UC08 – Vygenerovať tabuľku bezpečnostných stavov

3.2.1 UC01 – Zaregistrovať sa

Aktéri	• Návštevník • Systém
Cieľ	Pridať používateľa do systému a umožniť mu používať funkcionalitu vyžadujúcu prihlásenie.
Vstupné podmienky	• Používateľ nie je prihlásený do systému.
Ukončovacie podmienky	• Všetky povinné atributy sú vyplnené. • Zadané používateľské meno nie je obsadené. • Zadaný email nie je obsadený.
Konečný stav	Systém pridal používateľa do systému.
Spôsoby vyvolania	Otvorením registračného formulára.

Scenár	1. **Používateľ** vyplní povinné atributy:
	  • Meno a priezvisko
	  • Používateľské meno
	  • Email
	  • Heslo
	2. **Používateľ** potvrdí registráciu.
	3. **Systém** vykoná kontrolu splnenia ukončovacích podmienok.
	4. V prípade, že kontrola ukončovacích podmienok skončila neúspešne, **systém** zobrazí používateľovi relevantnú správu.
	  • *Návrat do kroku 1.*
	5. **Systém** pridá používateľa do systému a zobrazí mu relevantnú správu.
	  • *Koniec scenára.*

3.2.2 UC02 – Prihlásiť sa

Aktéri	• Návštevník • Systém
Cieľ	Identifikovať a autentizovať používateľa.
Vstupné podmienky	• Používateľ nie je prihlásený do systému.
Ukončovacie podmienky	• V systéme je zaregistrovaný používateľ so zadaným používateľským menom. • Používateľ zadal správne heslo.
Konečný stav	Systém autentizoval používateľa a sprístupnil mu funkcionalitu na základe jeho roly.
Spôsoby vyvolania	Otvorením prihlasovacieho formulára.
Scenár	1. **Používateľ** vyplní používateľské meno a heslo. 2. **Používateľ** potvrdí zadané údaje. 3. **Systém** vykoná kontrolu splnenia ukončovacích podmienok. 4. V prípade, že kontrola ukončovacích podmienok skončila neúspešne, **systém** zobrazí používateľovi relevantnú správu. • *Návrat do kroku 1.* 5. **Systém** prihlási používateľa do systému, autorizuje ho, a na základe jeho role mu zobrazí relevantné dáta. • *Koniec scenára.*

3.2.3 UC03 – Odhlásiť sa

Aktéri	• Administrátor • Bežný používateľ • Systém
Cieľ	Znemožniť používateľovi prístup k funkcionalite systému vyžadujúcej autentizáciu a autorizáciu.
Vstupné podmienky	• Používateľ je prihlásený do systému.
Ukončovacie podmienky	—
Konečný stav	Systém odhlásil používateľa zo systému.
Spôsoby vyvolania	Z ľubovoľného formulára vyžadujúceho autentizáciu.
Scenár	1. **Používateľ** vyvolá odhlásenie. 2. **Systém** odhlási používateľa zo systému. • *Koniec scenára.*

3.2.4 UC04 – Zobraziť prehľad regiónov

Aktéri	• Administrátor • Bežný používateľ • Systém
Cieľ	Zobraziť prehľad regiónov, s ktorými je ďalej možné vykonávať ďalšie aktivity.
Vstupné podmienky	• Používateľ je prihlásený do systému.
Ukončovacie podmienky	—
Konečný stav	Systém používateľovi zobrazil prehľad regiónov, ktorým má právo meniť bezpečnostný stav.
Spôsoby vyvolania	Automaticky po úspešnej autentizácií.
Scenár	1. V prípade, že používateľ nemôže meniť bezpečnostný stav žiadneho regiónu, **systém** používateľovi zobrazí relevantnú správu. • *Koniec scenára.* 2. V opačnom prípade **systém** používateľovi zobrazí prehľad regiónov, ktorým má právo meniť bezpečnostný stav. • *Koniec scenára.*

3.2.5 UC05 – Zobraziť detail regiónu

Aktéri	• Administrátor • Bežný používateľ • Systém
Cieľ	Zobraziť používateľovi atributy vybraného regiónu.
Vstupné podmienky	• Používateľ je prihlásený do systému. • Existuje región, ktorého detail je používateľ oprávnený zobraziť.
Ukončovacie podmienky	—
Konečný stav	Systém používateľovi zobrazil detail vybraného regiónu.
Spôsoby vyvolania	Z prehľadu regiónov.
Scenár	1. **Systém** používateľovi zobrazí detail vybraného regiónu. • *Koniec scenára.*

3.2.6 UC06 – Zmeniť bezpečnostný stav regiónu

Aktéri	• Administrátor • Bežný používateľ • Systém
Cieľ	Zmeniť bezpečnostný stav regiónu.
Vstupné podmienky	• Používateľ je prihlásený do systému. • Používateľ je oprávnený zmeniť bezpečnostný stav vybraného regiónu.
Ukončovacie podmienky	—
Konečný stav	Systém zmenil bezpečnostný stav vybraného regiónu.
Spôsoby vyvolania	Z detailu regiónu.
Scenár	1. **Používateľ** zvolí a potvrdí želaný bezpečnostný stav. 2. **Systém** vykoná kontrolu ukončovacích podmienok. 3. V prípade, že kontrola ukončovacích podmienok skončila neúspešne, **systém** zobrazí používateľovi relevantnú správu. • *Koniec scenára.* 4. **Systém** zmení bezpečnostný stav vybraného regiónu a zobrazí používateľovi relevantnú správu. • *Koniec scenára.*

3.2.7 UC07 – Priradiť používateľa k regiónu

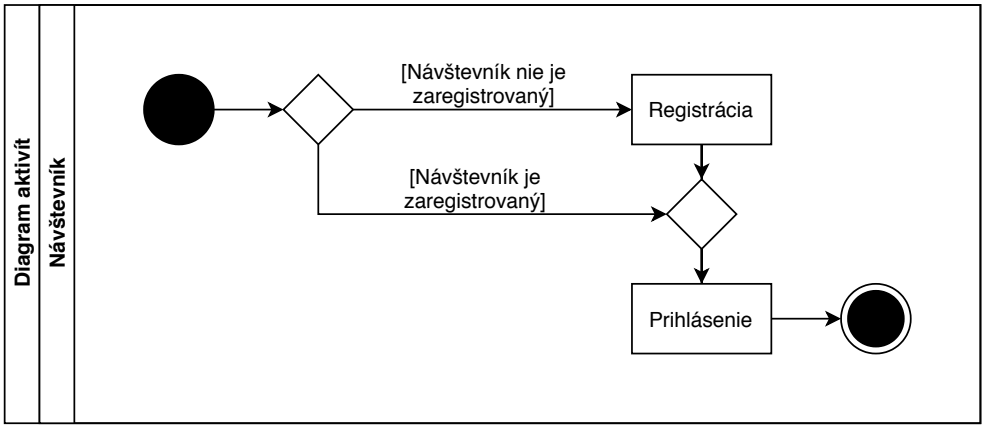
Aktéri	• Administrátor • Systém
Cieľ	Umožniť konkrétnemu používateľovi systému meniť bezpečnostný stav vybranému regiónu.
Vstupné podmienky	• Používateľ je prihlásený do systému.
Ukončovacie podmienky	—
Konečný stav	Systém priradil vybraného používateľa k vybranému regiónu.
Spôsoby vyvolania	Z prehľadu regiónov.
Scenár	1. Prihlásený **používateľ** zvolí región a používateľa, ktorého chce k regiónu priradiť, a potvrdí. 2. **Systém** priradí používateľa k regiónu a zobrazí prihlásenému používateľovi relevantnú správu. • *Koniec scenára.*

3.2.8 UC08 – Vygenerovať tabuľku bezpečnostných stavov

Aktéri	• Systém
Cieľ	Vygenerovať HTML tabuľku obsahujúcu regióny a ich aktuálne bezpečnostné stavy.
Vstupné podmienky	—
Ukončovacie podmienky	—
Konečný stav	Systém vrátil v odpovedi tabuľku aktuálnych bezpečnostných stavov.
Spôsoby vyvolania	Zavolaním príslušnej metódy systému.
Scenár	1. **Systém** z aktuálnych dát vygeneruje tabuľku a vráti ju v odpovedi. • *Koniec scenára.*

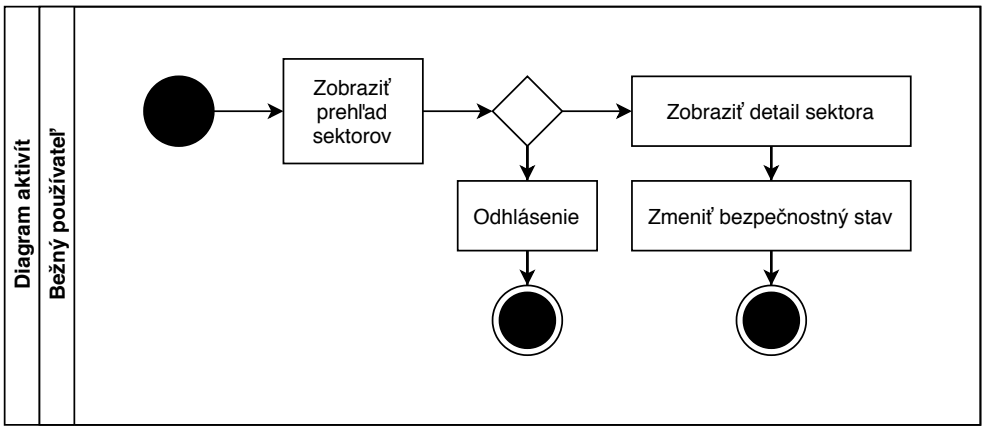
3.3 Diagramy aktivít

3.3.1 Diagram aktivít návštevníka



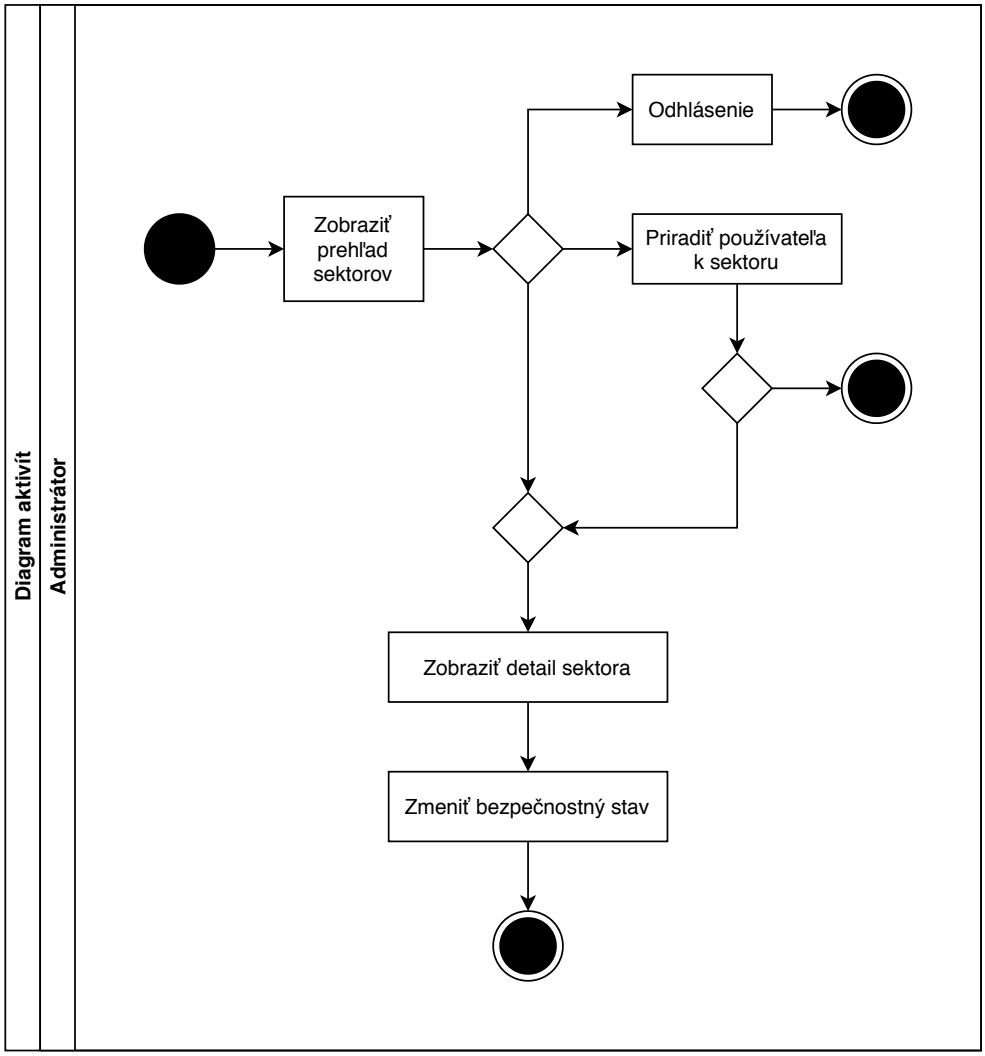
Obr. 3.1: Diagram aktivít návštevníka.

3.3.2 Diagram aktivít bežného používateľa



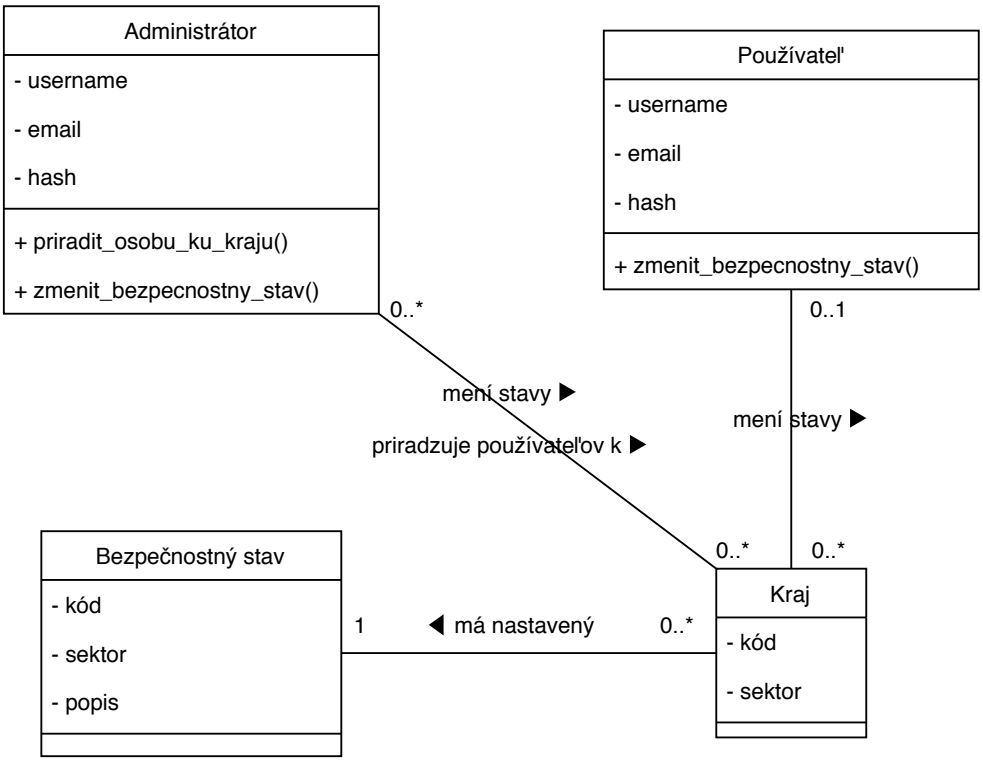
Obr. 3.2: Diagram aktivít bežného používateľa.

3.3.3 Diagram aktivít administrátora



Obr. 3.3: Diagram aktivít administrátora.

3.4 Diagram tried

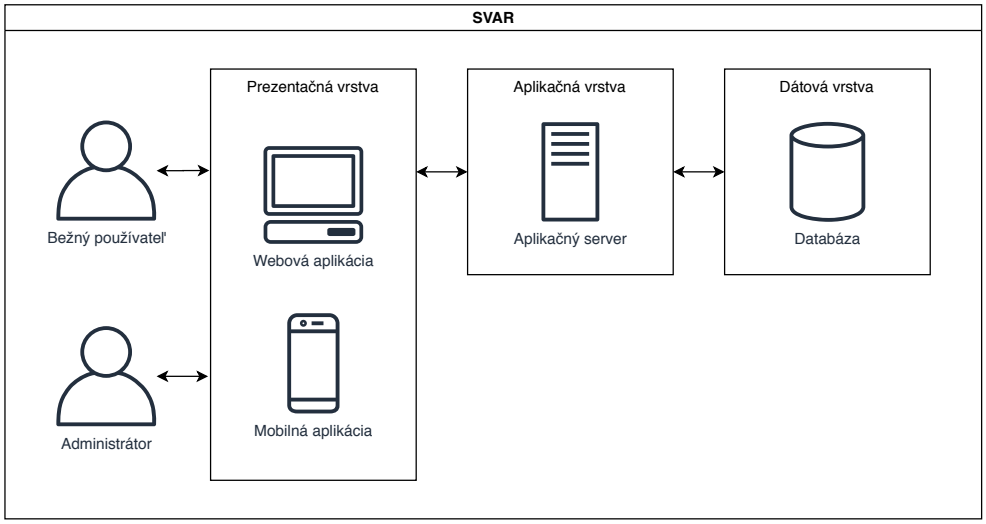


Obr. 3.4: Diagram tried systému SVAR.

3.5 Architektúra systému

Architektúra systému SVAR je založená na trojvrstvej architektúre. Jednotlivé vrstvy systému sú reprezentované nasledovne:

- **Prezentačná vrstva** (frontend) zložená z dvoch častí:
  - **Mobilná aplikácia**.
    - Návštevníkom umožňuje registráciu a prihlásenie.
    - Prihláseným používateľom umožňuje zmenu bezpečnostných stavov sektorov, ku ktorým sú priradení.
    - S backendom komunikuje zabezpečeným kanálom.
  - **Webová aplikácia**.
    - Disponuje funkcionalitou mobilnej aplikácie.
    - Administrátorovi navyše umožňuje meniť bezpečnostný stav ľubovoľného sektora a priraďovať používateľov k sektorom.
    - S backendom komunikuje zabezpečeným kanálom.
- **Aplikačná vrstva**. (backend)
  - Obsahuje všetkú business logiku a systémové kontroly.
  - Autentizuje a autorizuje používateľov.
  - Poskytuje rozhranie pre webový a mobilný frontend.
  - Generuje tabuľku bezpečnostných stavov.
  - S frontendom komunikuje zabezpečeným kanálom.
- **Dátová vrstva**.
  - Poskytuje úložisko dát vo forme relačnej databázy.
  - Uchováva dáta o používateľoch, sektoroch a bezpečnostných stavoch.



Obr. 3.5: Architektúra systému SVAR.

3.6 Požiadavky na systém

3.6.1 Požiadavky na frontend

- Webovú aplikáciu je možné používať bez nutnosti inštalovať dodatočný softvér.
- Webová aplikácia je použiteľná na aktuálnych verziách bežne používaných internetových prehliadačoch v Českej republike: [2]
  - Chrome/Chromium
  - Edge
  - Firefox
  - Internet Explorer
  - Opera
  - Safari
- Mobilná aplikácia je dostupná ako APK balíček pre operačný systém Android od verzie 4.1.x *Jelly Bean*.
- Nie sú použité technológie Flash ani Java.
- UI na používateľa nekladie kognitívnu záťaž.
- Podporuje šifrované spojenie.

3.6.2 Požiadavky na backend

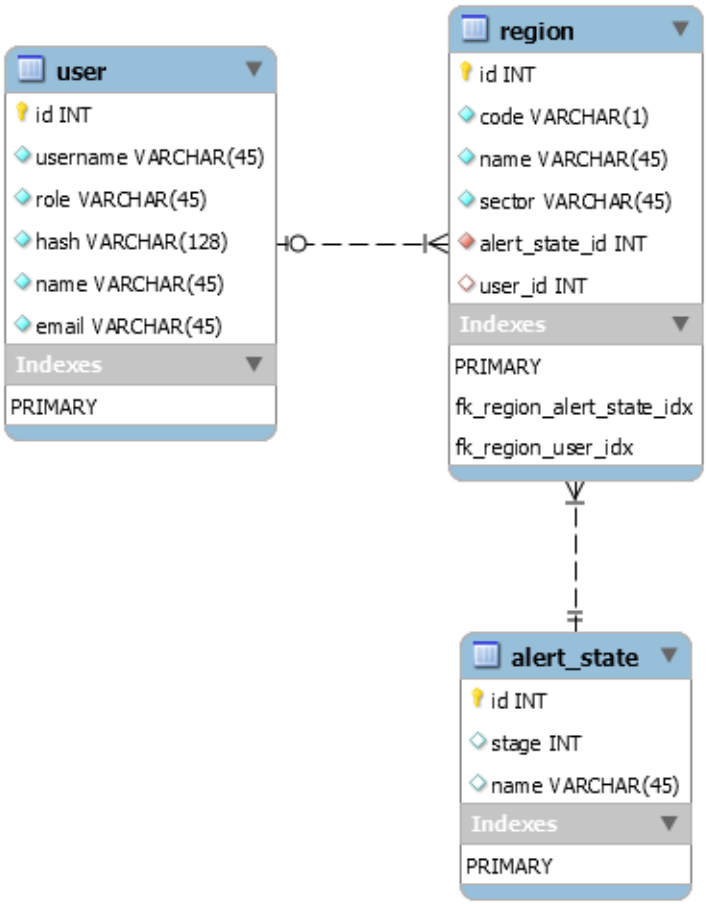
- Backend využíva technológie bežne používané pri implementácií podobných systémov.
- Použité technológie sú dostupné zdarma, bez licenčných poplatkov.
- Dodržiava základné zásady dobrého návrhu softvéru.
- Zdrojový kód backendu je ľahko rozšíriteľný a udržovateľný.
- Je relatívne nenáročný na systémové prostriedky.
- Je implementovaný modulárne.
- Podporuje šifrované spojenie.

3.7 Entity systému

Entita	**Atributy**
Používateľ	• Používateľské meno • Rola: – Administrátor – Bežný používateľ • Hash hesla • Celé meno • Email
Región	• Kód regiónu • Názov regiónu • Sektor: – Stred – Sever – Západ – Juh – Východ
Bezpečnostný stav	• Stupeň ohrozenia: – Stupeň 0 – Stupeň 1 – Stupeň 2 – Stupeň 3 • Názov stavu

3.8 ER diagram

Dátový model systému je zachytený v nasledujúcom ER diagrame, ktorý je použitý na vytvorenie tabuliek databázy:



Obr. 3.6: ER diagram systému SVAR.

Kapitola 4

Implementácia

4.1 Backend

Backend funguje ako *Node.js* aplikácia, ktorá číta a zapisuje z/do databázy implementovanej v *MySQL*. Komunikácia je zabezpečená techológiou *JSON Web Token*. Aplikácia sprístupňuje dáta vo formáte JSON prostredníctvom rozhrania s REST architektúrou.

4.1.1 REST rozhranie

REST API aplikácie je implementované pomocou *Node.js* frameworku *Express*. Framework funguje ako middleware, ktorý spracuváva HTTP požiadavky a odpovede a automaticky nastavuje HTTP hlavičky. Ďalej umožňuje jednoducho definovať REST zdroje a povolené HTTP metódy. Zdroje a povolené HTTP metódy sú uvedené v nasledujúcej tabuľke:

Zdroj	**Operácia**	**Popis**
/authenticate	POST	Prihlásenie existujúceho používateľa.
/register	POST	Registrácia nového používateľa.
/users	GET	Zoznam používateľov systému.
/region	GET	Zoznam regiónov.
/region/:rid/alert	PUT	Zmena bezpečnostného stavu regiónu s identifikátorom *rid*.
/region/:rid/user	PUT	Zmena používateľa, ktorý môže meniť bezpečnostné stavy regiónu s identifikátorom *rid*.
/alertState	GET	Zoznam bezpečnostných stavov.
/table	GET	HTML tabuľka regiónov a ich aktuálnych bezpečnostných stavov.

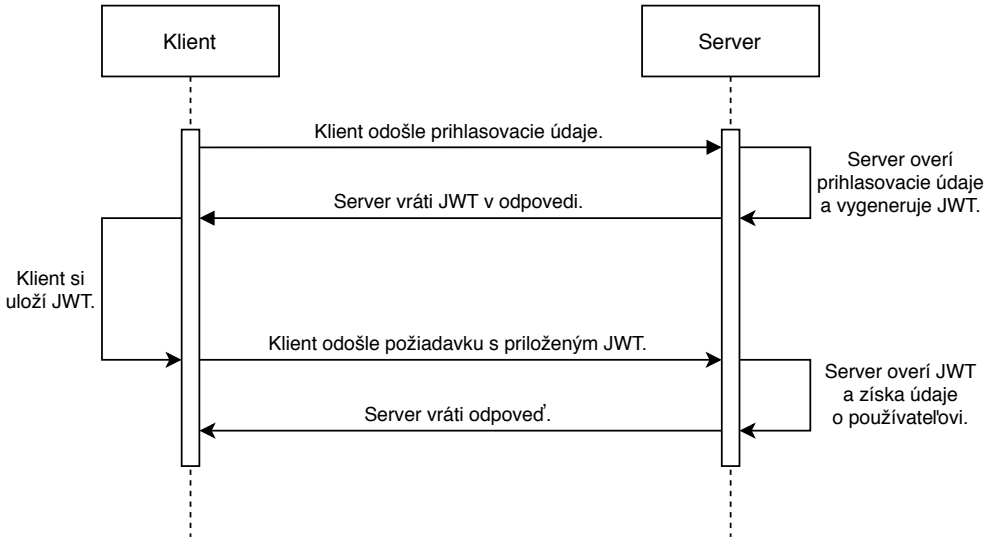
4.1.2 Zabezpečenie

Autentizácia je riešená pomocou *JSON Web Token* (JWT). Toto riešenie má výhodu v tom, že je bezstavové, a nie sú potrebné cookies. Po obdržaní prihlasovacích údajov server vygeneruje kľúč, ktorý v odpovedi vráti klientovi. Kľúč sa skladá z troch častí:

- Hlavička – obsahuje typ kľúča (JWT) a algoritmus použitý na vygenerovanie podpisu.
- Telo – obsahuje informácie o používateľovi. Backend systému prikladá ID používateľa a platnosť kľúča.
- Podpis – algoritmom uvedeným v hlavičke sa vygeneruje podpis zložený zo zakódovanej hlavičky, tela a tajomstva.

Klient si obdržaný kľúč uloží a následne ho prikladá do hlavičky každej požiadavky odoslanej na server. Server pred spracovaním každej požiadavky najprv overí platnosť a autenticitu obdržaného kľúča. V prípade, že kľúč vypršal alebo je falošný, server vyhodí výnimku. JWT okrem iného obsahuje zakódovanú rolu používateľa, na základe ktorej prebieha autorizácia používateľa k REST zdrojom.

Princíp autentizácie pomocou JWT zachycuje nasledujúci diagram:



Obr. 4.1: Autentizácia pomocou JWT.

4.1.3 Architektúra

Aplikácia je rozdelená do hlavných a pomocných komponentov. Metódy jednotlivých komponentov sú implementované asynchrónne. V praxi to znamená, že miesto v zdrojovom kóde, odkiaľ je metóda volaná, nečaká na odpoveď volaného kódu. Aplikácia je rozdelená do nasledujúcich komponentov:

- **server.js**
  - Základný kameň aplikácie, ktorý má na starosti spustenie servera na definovanej IP adrese (resp. *hostname*) a porte.
- **controller.js**
  - Plní úlohu smerovača (*router*) – definuje REST zdroje, povolené HTTP metódy a metódy služby, ktoré sa majú zavolať po zavolaní metódy daného zdroja. Po spracovaní požiadavky komponent nastaví telo odpovede na základe výsledku volanej metódy služby. Prípadné chyby sa vypisujú do konzoly.
- **service.js**
  - Služba obsahuje všetkú business logiku. Spracuváva telo požiadaviek, na základe ich obsahu volá metódy komponenty Model a v prípade potreby hádže výnimky.
- **model.js**
  - Komponent obsluhujúci databázu, obsahuje všetky *SQL* dotazy. Po obdržaní dát z databázy ich vracia službe. Prípadné chyby sa vypisujú do konzoly.
- **db.js**
  - Pomocný komponent, ktorý slúži na vytvorenie spojenia s databázou.
- **jwt.js**
  - Pomocný komponent pre operácie s JWT a definovanie verejných ciest k REST zdrojom.

4.2 Frontend

4.2.1 Mobilná aplikácia

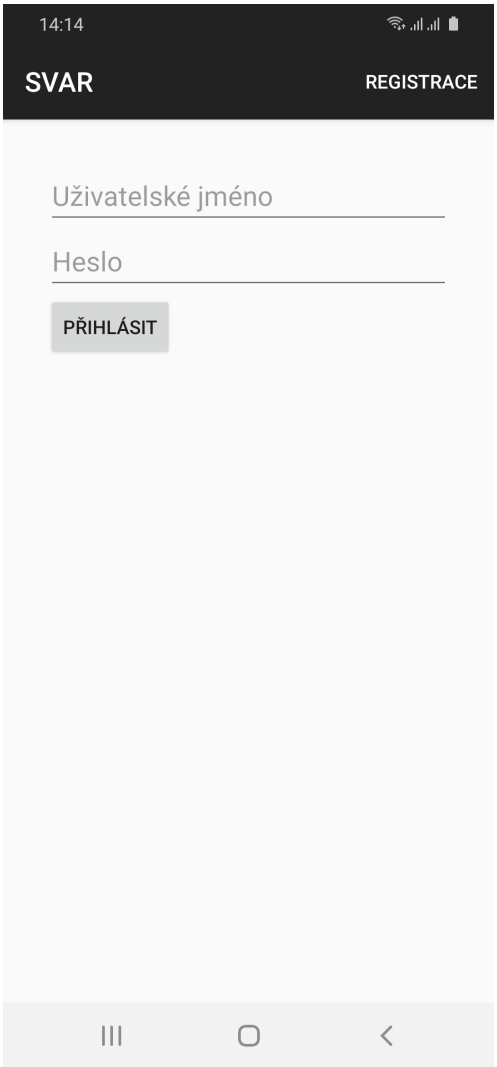
Mobilná aplikácia je naimplementovaná v technológií *Kotlin* – preferovanom programovacom jazyku pre Android. [3] Architektúra aplikácie je podobne ako backend rozdelená do logických komponentov. Hlavné komponenty aplikácie predstavujú:

- **Model**
  - Sú tu definované entity systému a ich atributy.
- **Interface**
  - Reprezentuje REST rozhranie poskytované backendom. Definuje zdroje rozhrania a povolené HTTP metódy. Každý zdroj má ďalej definované, aké dáta sú odosielané v požiadavkoch na daný zdroj a akého dátového typu sú obdržané odpovede z backendu.
- **Session manager**
  - Správa relácie (*session manager*) implementuje metódy pre operácie s JWT – uloženie, získanie a odstránenie.
- **Aktivity**
  - Aktivita predstavuje konkrétnu obrazovku aplikácie. Skladá sa z kontroléra a pohľadu. Aplikácia obsahuje nasledovné aktivity:
    - Prihlásenie – prihlasovací formulár.
    - Registrácia – registračný formulár.
    - Prehľad regiónov – obsahuje zoznam regiónov obdržaných z backendu, ku ktorým má používateľ prístup.
    - Detail regiónu – slúži na zmenu bezpečnostného stavu daného regiónu.

4.2.1.1 Používateľské rozhranie

Používateľské rozhranie mobilnej aplikácie využíva zabudované Android UI komponenty a skladá sa zo štyroch pohľadov:

- **Prihlasovací formulár**:
  - Obsahuje textboxy pre vyplnenie používateľského mena a hesla.
  - Tlačítko *PŘIHLÁSIT* odošle prihlasovacie údaje.
  - Tlačítko *REGISTRACE* presmeruje používateľa na registračný formulár.



Obr. 4.2: Prihlasovací formulár.

- **Registračný formulár**:
  - Obsahuje štyri povinné textboxy pre vyplnenie celého mena, emailu, používateľského mena a hesla.
  - Tlačítko *ODESLAT* odošle registračné údaje.
  - Tlačítko ← presmeruje používateľa späť na prihlasovací formulár.

Obr. 4.3: Registračný formulár.

- **Prehľad regiónov**:
  - Obsahuje regióny, ktoré má prihlásený používateľ priradené a môže meniť ich bezpečnostný stav. Administrátor vidí všetky regióny a môže meniť bezpečnostný stav ľubovoľného regiónu.
  - U každého regiónu je zobrazený jeho aktuálny bezpečnostný stav a farebne odlíšený stupeň ohrozenia.
  - Po kliknutí na región sa zobrazí jeho detail.
  - Tlačítko *ODHLÁSIT* odhlási používateľa z aplikácie.

Obr. 4.4: Prehľad regiónov.

- **Detail regiónu**:
  - Obsahuje zoznam bezpečnostných stavov. Aktuálny bezpečnostný stav je zvýraznený **tučne**.
  - Tlačítko *POTVRDIT* nastaví regiónu vybraný bezpečnostný stav.
  - Tlačítko ← vráti používateľa na prehľad regiónov.

Obr. 4.5: Detail regiónu.

4.2.2 Webová aplikácia

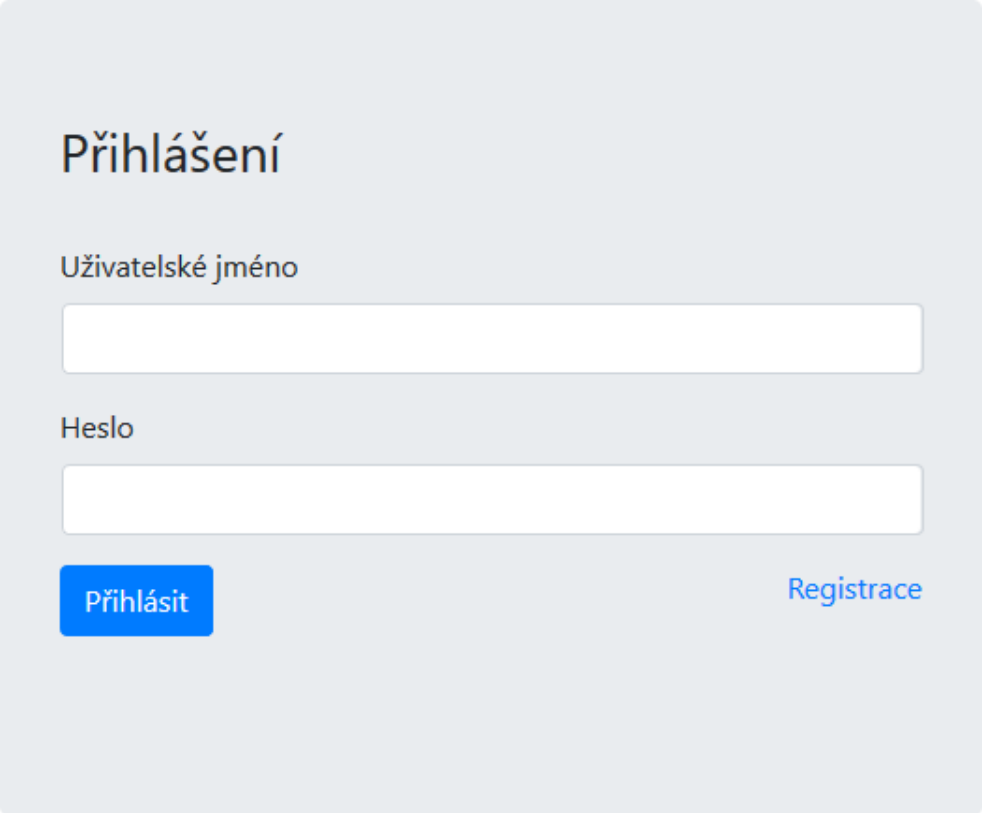
Aplikácia využíva technológie *Vue.js* a *Bootstrap*. V aplikácií sú využité asynchrónne volania metód. Architektúra je znovu rozdelená do logických komponentov:

- **main.js**
  - Kmeňový súbor obsahujúci import závislostí a inštanciu *Vue* aplikácie.
- **Router**
  - Smerovač obsahujúci cesty k zdrojom REST rozhrania a prislúchajúce *Vue komponenty*.
- **Service**
  - Volajú sa tu HTTP metódy. Obsah odpovede z backendu je následne sprístupnený *Vue komponentu*, ktorý zavolal príslušnú metódu.
- **Vue komponenty**
  - Vue komponent je obdobou aktivity v prostredí Android – jedná sa o pohľad a jeho kontrolér. Každý Vue komponent obsahuje HTML šablonu a Vue.js kód, v ktorom sú definované premenné, metódy, a ďalšie Vue.js konštrukty. Vue komponenty konzumujú rozhranie služby. Webová aplikácia obsahuje nasledujúce Vue komponenty:
    - Prihlásenie – prihlasovací formulár.
    - Registrácia – registračný formulár.
    - Domovská stránka – obsahuje zoznam regiónov, bezpečnostných stavov a v prípade, že je prihlásený administrátor, aj priradené osoby oprávnené meniť bezpečnostné stavy.

4.2.2.1 Používateľské rozhranie

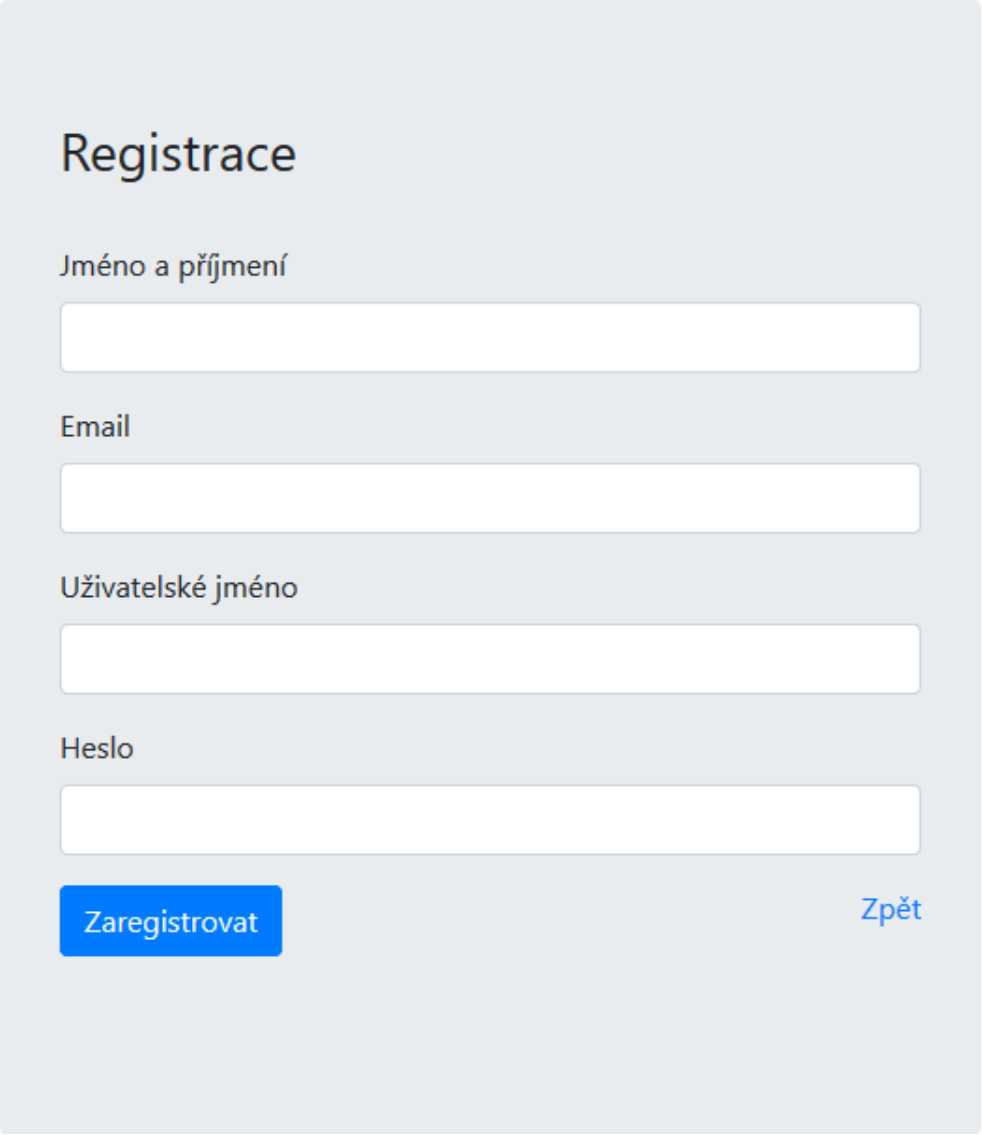
Používateľské rozhranie využíva technológiu *Bootstrap*. Hlavným prínosom je *responzívnosť*. V praxi to znamená, že pri zmenšovaní okna sa automaticky mení rozloženie elementov tak, aby boli viditeľné bez nutnosti horizontálneho posúvania. Používateľské rozhranie pozostáva z troch pohľadov:

- **Prihlasovací formulár** obsahujúci:
  - Textboxy pre vyplnenie používateľského mena a hesla.
  - Tlačítko *Přihlásit* pre odoslanie prihlasovacích údajov.
  - Preklik *Registrace* na registračný formulár.



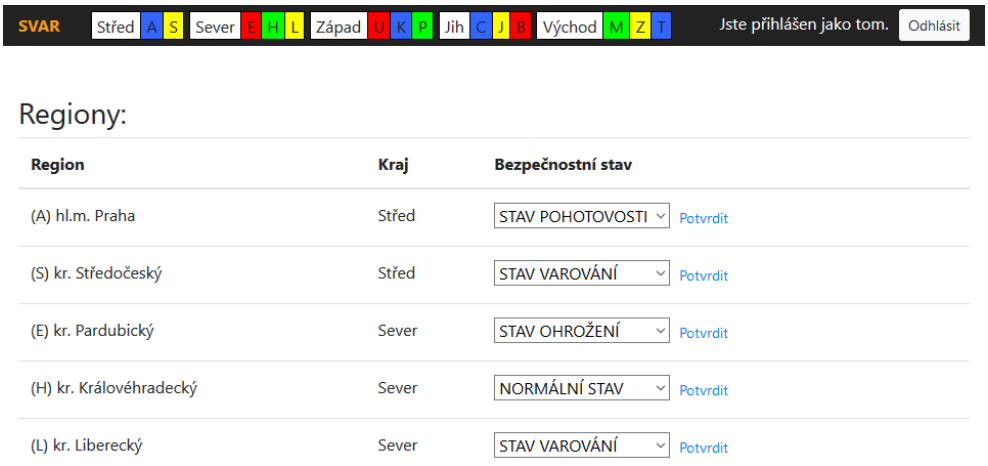
Obr. 4.6: Prihlasovací formulár.

- **Registračný formulár**:
  - Nachádzajú sa tu 4 povinné textboxy – celé meno, email, používateľské meno a heslo.
  - Vyplnené údaje sa odošlú kliknutím na tlačítko *Zaregistrovat*.
  - Preklik *Zpět* na prihlasovací formulár.



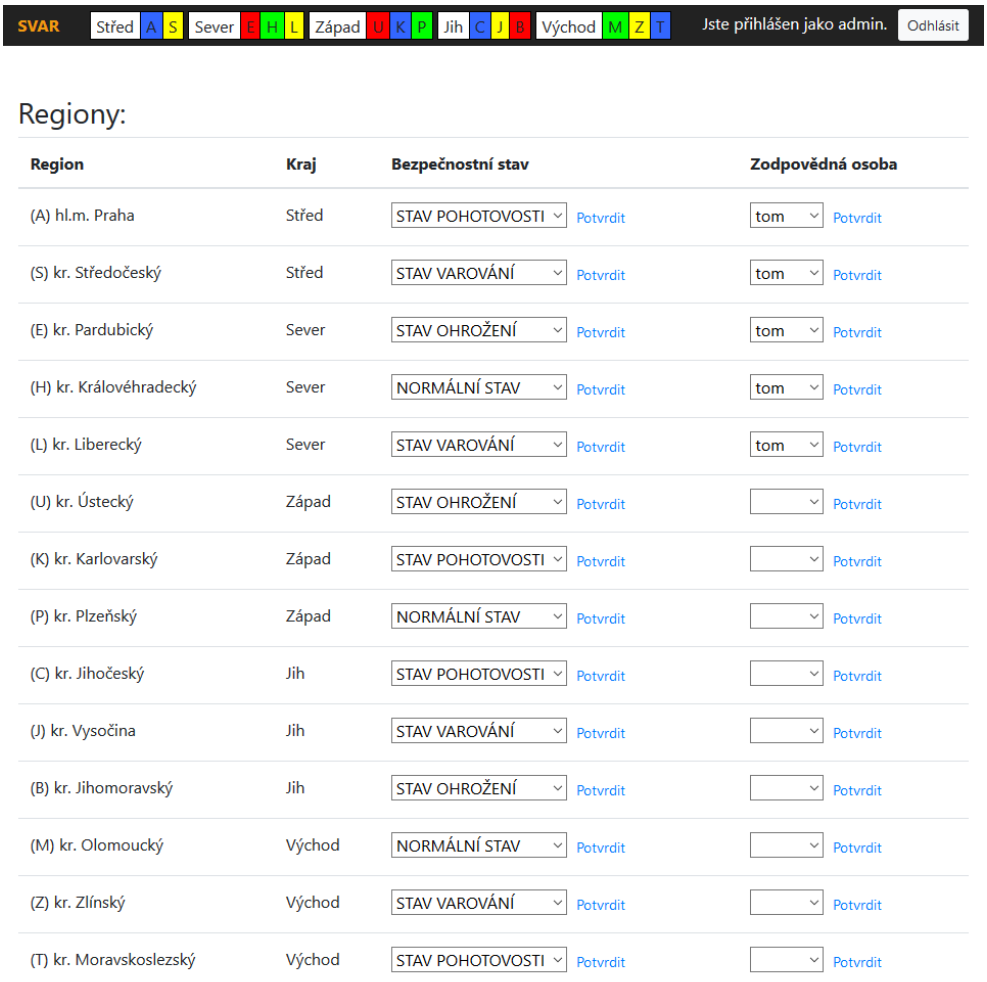
Obr. 4.7: Registračný formulár.

- **Hlavná stránka** zložená z dvoch častí:
  - **Lišta** obsahujúca nasledujúce prvky:
    - Tabuľka bezpečnostných stavov.
    - Username prihláseného používateľa.
    - Tlačítko *Odhlásit* pre odhlásenie.
  - **Zoznam regiónov**:
    - Obsahuje regióny, ku ktorým je prihlásený používateľ priradený a môže meniť ich bezpečnostné stavy. Každý bezpečnostný stav má svoju farbu.
    - Regióny, ktorých bezpečnostný stav používateľ nemôže meniť, sa v zozname nevyskytujú.
    - Tlačítko *Potvrdit* nastaví regiónu vybraný bezpečnostný stav.



Obr. 4.8: Hlavná stránka bežného používateľa.

- Používateľské rozhranie administrátora je identické s rozhraním pre bežného používateľa s tým rozdielom, že administrátor má zobrazené všetky regióny systému vrátane priradených používateľov. Administrátor môže meniť bezpečnostný stav všetkých regiónov a priradiť ľubovoľného používateľa systému k ľubovoľnému regiónu.



Obr. 4.9: Používateľské rozhranie administrátora.

4.3 Kľúčové vlastnosti systému

- Systém je implementovaný modulárne a skladá sa z nasledujúcich častí:
  - Databáza
  - Backend
  - Frontend:
    - Mobilná aplikácia
    - Webová aplikácia
- Jednotlivé časti systému využívajú moderné technológie so solídnou podporou, medzi ktoré patrí *Node.js*, *Vue.js*, či *Kotlin*.
- Moduly systému fungujú nezávisle na sebe a sú zameniteľné. Je napríklad možné naimplementovať ďalšiu frontend aplikáciu konzumujúcu REST API poskytované backendom. Ďalej je možné napríklad pridať ďalšiu entitu do databázy bez toho, aby došlo k znefunkčneniu zvyšných častí systému.
- Backend poskytuje dáta prostredníctvom REST rozhrania vo formáte JSON.
- Metódy backendu sú implementované asynchrónne. Vďaka tomu napríklad nedôjde k zaseknutiu v prípade, že databáza posiela oneskorené odpovede na dotazy.
- Autentizácia využíva JWT, vďaka čomu nie je nutný session management ani použitie cookies.
- Systém umožňuje použitie zabezpečenej komunikácie prostredníctvom SSL a je pripravený na nasadenie do ostrej prevádzky.

Kapitola 5

Testovanie

5.1 Integračné testy

Sekcia obsahuje testovacie prípady integrácie REST rozhrania backendovej aplikácie a databázy. Účelom je overiť, či:

- Backend korektne prijíma a odpovedá na požiadavky.
- Databáza korektne ukladá dáta.

Testovanie prebiehalo kontinuálne počas implementácie backendu použitím utility *Postman*.

5.1.1 IT01 – Autentizácia

5.1.1.1 Pozitívny prípad

- **REST zdroj:** /authenticate
- **HTTP metóda:** POST
- **Prerekvizity:** V systéme je zaregistrovaný používateľ s používateľským menom admin a heslom admin.
- **Požiadavka:**

```
{ "username": "admin", "password": "admin" }
```

- **Očakávaná odpoveď:**

```
{ "user": { "id": $i, "username": $u, "role": $r, "name": $n,
"email": $e, "token": $t } }
```

kde $i, $u, $r, $n, $e sú príslušné atribúty používateľa *admin* a $t je JWT vygenerovaný backend aplikáciou.

- **Postrekvizity:** Vygenerovaný JWT je platný.
- **Výsledok testu:** ÚSPECH

5.1.1.2 Negatívny prípad I

- **REST zdroj:** /authenticate
- **HTTP metóda:** POST
- **Prerekvizity:** V systéme je zaregistrovaný používateľ s username admin a heslom admin.
- **Požiadavka:**

```
{ "username": "admin", "password": "test" }
```

- **Očakávaná odpoveď:**

```
{ "message": "Nesprávne přihlašovací jméno nebo heslo." }
```

- **Postrekvizity:** —
- **Výsledok testu:** ÚSPECH

5.1.1.3 Negatívny prípad II

- **REST zdroj:** /authenticate
- **HTTP metóda:** POST
- **Prerekvizity:** V systéme neexistuje používateľ s username admin.
- **Požiadavka:**

```
{ "username": "admin", "password": "admin" }
```

- **Očakávaná odpoveď:**

```
{ "message": "Nesprávne přihlašovací jméno nebo heslo." }
```

- **Postrekvizity:** —
- **Výsledok testu:** ÚSPECH

5.1.2 IT02 – Registrácia

5.1.2.1 Pozitívny prípad

- **REST zdroj:** /register
- **HTTP metóda:** POST
- **Prerekvizity:** V systéme nie je zaregistrovaný používateľ s username admin alebo používateľ s emailovou adresou admin@cvut.cz.
- **Požiadavka:**

```
{ "name": "Admin", "username": "admin", "password": "admin",
"email": "admin@cvut.cz" }
```

- **Očakávaná odpoveď:**

```
{ "message": "Byl jste úspěšně zaregistrován." }
```

- **Postrekvizity:** Databáza obsahuje používateľa definovaného v odoslanej požiadavke.
- **Výsledok testu:** ÚSPECH

5.1.2.2 Negatívny prípad I

- **REST zdroj:** /register
- **HTTP metóda:** POST
- **Prerekvizity:** V systéme je zaregistrovaný používateľ s username admin a emailovou adresou admin@cvut.cz.
- **Požiadavka:**

```
{ "name": "Admin", "username": "admin", "password": "admin",
"email": "test@test.cz" }
```

- **Očakávaná odpoveď:**

```
{ "message": "Uživatel \"admin\" již v systému existuje." }
```

- **Postrekvizity:** Databáza je bez zmeny.
- **Výsledok testu:** ÚSPECH

5.1.2.3 Negatívny prípad II

- **REST zdroj:** /authenticate
- **HTTP metóda:** POST
- **Prerekvizity:** V systéme je zaregistrovaný používateľ s username admin a emailovou adresou admin@cvut.cz.
- **Požiadavka:**

```
{ "name": "Tom", "username": "tom", "password": "tom",
"email": "admin@cvut.cz" }
```

- **Očakávaná odpoveď:**

```
{ "message": "Email \"admin@cvut.cz\" již byl zaregistrován." }
```

- **Postrekvizity:** Databáza je bez zmeny.
- **Výsledok testu:** ÚSPECH

5.1.3 IT03 – Zoznam používateľov

5.1.3.1 Pozitívny prípad

- **REST zdroj:** /users
- **HTTP metóda:** GET
- **Prerekvizity:**
  - V systéme sa nachádza používateľ s username admin s rolou ADMIN.
  - $token je JWT obdržaný po úspešnej autentizácií používateľa admin.
  - $p je požiadavka obsahujúca hlavičku:

    ```
    Authorization: Bearer $token
    ```

- **Požiadavka:** $p
- **Očakávaná odpoveď:** Pole objektov obsahujúce používateľov systému v nasledujúcom formáte:

```
[ { "id": $i, "username": $u, "role": $r, "hash": $h,
"name": $n, "email": $e } ]
```

  kde $i, $u, $r, $h, $n, $e sú príslušné atribúty daného používateľa.

- **Postrekvizity:** —
- **Výsledok testu:** ÚSPECH

5.1.3.2 Negatívny prípad

- **REST zdroj:** /users
- **HTTP metóda:** GET
- **Prerekvizity:**
  - V systéme sa nachádza používateľ s username tom s rolou USER.
  - $token je JWT obdržaný po úspešnej autentizácií používateľa tom.
  - $p je požiadavka obsahujúca hlavičku:

    ```
    Authorization: Bearer $token
    ```
- **Požiadavka:** $p
- **Očakávaná odpoveď:**

  ```
  { "message": "Neoprávněný přístup." }
  ```
- **Postrekvizity:** —
- **Výsledok testu:** ÚSPECH

5.1.4 IT04 – Zoznam regiónov

5.1.4.1 Pozitívny prípad

- **REST zdroj:** /region
- **HTTP metóda:** GET
- **Prerekvizity:**
  - V systéme sa nachádza používateľ s username tom s rolou USER.
  - V systéme sa nachádza región $r s atribútmi $id, $code, $rname, $sector.
  - V systéme sa nachádza bezpečnostný stav $s s atribútmi $stage, $aname.
  - Používateľ tom má priradený región $r.
  - Región $r má priradený bezpečnostný stav $s.
  - $token je JWT obdržaný po úspešnej autentizácií používateľa tom.
  - $p je požiadavka obsahujúca hlavičku:

    ```
    Authorization: Bearer $token
    ```

- **Požiadavka:** $p
- **Očakávaná odpoveď:** Pole objektov obsahujúce región $r v nasledujúcom formáte:

  ```
  [ { "id": $id, "code": $code, "name": $rname, "sector": $sector,
  "user": "tom", "stage": $stage, "alert": $aname } ]
  ```

  kde $id, $code, $rname, $sector, $stage, $aname sú príslušné atribúty regiónu $r.
- **Postrekvizity:** —
- **Výsledok testu:** ÚSPECH

5.1.4.2 Negatívny prípad

- **REST zdroj:** /region
- **HTTP metóda:** GET
- **Prerekvizity:**
  - V systéme sa nachádza používateľ s username tom s rolou USER.
  - Používateľ tom nemá priradený žiaden región.
  - $token je JWT obdržaný po úspešnej autentizácii používateľa tom.
  - $p je požiadavka obsahujúca hlavičku:

    ```
    Authorization: Bearer $token
    ```
- **Požiadavka:** $p
- **Očakávaná odpoveď:**

  ```
  [ ]
  ```
- **Postrekvizity:** —
- **Výsledok testu:** ÚSPECH

5.1.5 IT05 – Zmena bezpečnostného stavu

5.1.5.1 Pozitívny prípad

- **REST zdroj:** /region/$rid/alert
- **HTTP metóda:** PUT
- **Prerekvizity:**
  - V systéme sa nachádza používateľ s username tom s rolou USER.
  - V systéme sa nachádza región $r s *ID* = $rid.
  - V systéme sa nachádzajú bezpečnostné stavy s_1$ s *ID* = 1 a s_2$ s *ID* = 2.
  - Používateľ tom má priradený región $r.
  - Región $r má priradený bezpečnostný stav s_1$.
  - $token je JWT obdržaný po úspešnej autentizácií používateľa tom.
  - $p je požiadavka obsahujúca hlavičku:

    ```
    Authorization: Bearer $token
    ```
- **Požiadavka:** $p :=

  ```
  { aid: 2 }
  ```
- **Očakávaná odpoveď:**

  ```
  { "message": "Bezpečnostní stav úspěšně změněn." }
  ```
- **Postrekvizity:** Región $r má na svojom riadku tabuľky *svar.region* nastavený cudzí kľúč *alert_state_id* = 2.
- **Výsledok testu:** ÚSPECH

5.1.5.2 Negatívny prípad

- **REST zdroj:** /region/$rid/alert
- **HTTP metóda:** PUT
- **Prerekvizity:**
  - V systéme sa nachádza používateľ s username tom s rolou USER.
  - V systéme sa nachádza región $r s *ID* = $rid.

  - V systéme sa nachádzajú bezpečnostné stavy $s_1 s $ID = 1$ a $s_2 s $ID = 2$.
  - Používateľ tom nemá priradený región $r.
  - Región $r má priradený bezpečnostný stav s_1.
  - $token je JWT obdržaný po úspešnej autentizácii používateľa tom.
  - $p je požiadavka obsahujúca hlavičku:

    ```
    Authorization: Bearer $token
    ```

- **Požiadavka:** $p :=

  ```
  { aid: 2 }
  ```

- **Očakávaná odpoveď:**

  ```
  { "message": "Neoprávněný přístup." }
  ```

- **Postrekvizity:** Databáza je bez zmeny.
- **Výsledok testu:** ÚSPECH

5.1.6 IT06 – Priradenie používateľa

5.1.6.1 Pozitívny scenár

- **REST zdroj:** /region/$rid/user
- **HTTP metóda:** PUT
- **Prerekvizity:**
  - V systéme sa nachádza používateľ s username admin s rolou ADMIN.
  - V systéme sa nachádza región $r s $ID = $rid.
  - V systéme sa nachádza používateľ $user s $ID = 2$.
  - Používateľ $user nemá priradený región $r.
  - $token je JWT obdržaný po úspešnej autentizácii používateľa admin.
  - $p je požiadavka obsahujúca hlavičku:

    ```
    Authorization: Bearer $token
    ```

- **Požiadavka:** $p :=

  ```
  { uid: 2 }
  ```

- **Očakávaná odpoveď:**

  ```
  { "message": "Uživatel úspěšně přiřazen." }
  ```

- **Postrekvizity:** Región $r má na svojom riadku tabuľky *svar.region* nastavený cudzí kľúč *user_id* = 2.
- **Výsledok testu:** ÚSPECH

5.1.6.2 Negatívny scenár

- **REST zdroj:** /region/$rid/user
- **HTTP metóda:** PUT
- **Prerekvizity:**
  - V systéme sa nachádza používateľ s username tom s rolou USER.
  - V systéme sa nachádza región $r s *ID* = $rid.
  - V systéme sa nachádza používateľ $user s *ID* = 2.
  - Používateľ $user nemá priradený región $r.
  - $token je JWT obdržaný po úspešnej autentizácií používateľa tom.
  - $p je požiadavka obsahujúca hlavičku:

    ```
    Authorization: Bearer $token
    ```

- **Požiadavka:** $p :=

  ```
  { uid: 2 }
  ```

- **Očakávaná odpoveď:**

  ```
  { "message": "Neoprávněný přístup." }
  ```

- **Postrekvizity:** Databáza je bez zmeny.
- **Výsledok testu:** ÚSPECH

5.1.7 IT07 – Zoznam bezpečnostných stavov

- **REST zdroj:** /alertState
- **HTTP metóda:** GET
- **Prerekvizity:**
  - V systéme sa nachádza používateľ s username tom s rolou USER.
  - V systéme sa nachádza bezpečnostný stav $s s atribútmi $stage, $aname.
  - $token je JWT obdržaný po úspešnej autentizácií používateľa tom.
  - $p je požiadavka obsahujúca hlavičku:

    ```
    Authorization: Bearer $token
    ```
- **Požiadavka:** $p
- **Očakávaná odpoveď:** Pole objektov obsahujúce bezpečnostný stav $s v nasledujúcom formáte:

  ```
  [ { "id": $id, "stage": $stage, "name": $name } ]
  ```

  kde $id, $stage, $name sú príslušné atribúty bezpečnostného stavu $s.
- **Postrekvizity:** —
- **Výsledok testu:** ÚSPECH

5.1.8 IT08 – Vygenerovanie tabuľky

- **REST zdroj:** /table
- **HTTP metóda:** GET
- **Prerekvizity:** —
- **Požiadavka:**

  ```
  {}
  ```
- **Očakávaná odpoveď:** HTML kód v prostom texte obsahujúci regióny a ich aktuálne priradené bezpečnostné stavy.
- **Postrekvizity:** —
- **Výsledok testu:** ÚSPECH

5.2 Funkčné testy

Sekcia obsahuje testovacie prípady (*TC – test cases*) pokrývajúce funkcionalitu systému. Cieľom funkčného testovania je zistiť, či implementovaný systém odpovedá funkčnej špecifikácií a požiadavkam zadávateľa. Testy boli uskutočnené dňa 4. októbra 2020 na nasledujúcej testovacej konfigurácií:

- **Backend:**
  - PC: Lenovo Yoga 330-11IGM, CPU Intel Pentium N5000, RAM 4GB, eMMC 32GB
  - Platforma: Windows 10 Pro 64-bit build 1909
- **Frontend:**
  - **Webová aplikácia:**
    - PC: Dell Latitude 5400, CPU Intel Core i7-8665U, RAM 16GB, SSD 512GB
    - Platforma: Windows 10 Pro 64-bit build 1909
    - Prehliadač: Microsoft Edge 85
  - **Mobilná aplikácia:**
    - Mobil: Samsung Galaxy A40
    - Platforma: Android 10 (API level 29)

5.2.1 TC01 – Zaregistrovať sa

5.2.1.1 Pozitívny scenár

Moduly	Mobilná a webová aplikácia.
Prerekvizity	V systéme nie je zaregistrovaný používateľ s username tom alebo používateľ s emailovou adresou tom@cvut.cz.
Scenár	1. Vyplniť povinné atribúty: • Celé meno: Tom • Username: tom • Email: tom@cvut.cz • Heslo: tom 2. Potvrdiť registráciu.
Postrekvizity	V systéme je zaregistrovaný používateľ so zadanými atribútmi.
Výsledok	ÚSPECH

5.2.1.2 Negatívny scenár

Moduly	Mobilná a webová aplikácia.
Prerekvizity	V systéme je zaregistrovaný používateľ s username tom2 a používateľ s emailovou adresou tom2@cvut.cz.
Scenár	1. Vyplniť povinné atribúty: • Celé meno: Tom 2 • Username: tom2 • Email: tom2@cvut.cz • Heslo: tom2 2. Potvrdiť registráciu.
Postrekvizity	Systém nezaregistruje používateľa so zadanými atribútmi z dôvodu, že username tom2 a email tom2@cvut.cz už sú zaregistrované.
Výsledok	ÚSPECH

5.2.2 TC02 – Prihlásiť sa

5.2.2.1 Pozitívny scenár

Moduly	Mobilná a webová aplikácia.
Prerekvizity	V systéme je zaregistrovaný používateľ s username tom a heslom tom.
Scenár	1. Vyplniť povinné atribúty: • Username: tom • Heslo: tom 2. Potvrdiť prihlásenie.
Postrekvizity	Systém autentizoval používateľa a po autorizácií mu zobrazil relevantné dáta na základe jeho role.
Výsledok	ÚSPECH

5.2.2.2 Negatívny scenár

Moduly	Mobilná a webová aplikácia.
Prerekvizity	V systéme je zaregistrovaný používateľ s username tom a heslom tom. Používateľ s username tom2 neexistuje.
Scenár I	1. Vyplniť povinné atribúty: • Username: tom • Heslo: test 2. Potvrdiť prihlásenie.
Scenár II	1. Vyplniť povinné atribúty: • Username: tom2 • Heslo: tom2 2. Potvrdiť prihlásenie.
Postrekvizity	Systém ani v jednom prípade neprihlásil používateľa z dôvodu nesprávne zadaného username, resp. neexistujúceho používateľa.
Výsledok	ÚSPECH

5.2.3 TC03 – Odhlásiť sa

5.2.3.1 Pozitívny scenár

Moduly	Mobilná a webová aplikácia.
Prerekvizity	Používateľ je prihlásený do systému.
Scenár	1. Odhlásiť sa.
Postrekvizity	Systém odhlásil používateľa a presmeroval ho na prihlasovací formulár.
Výsledok	ÚSPECH

5.2.4 TC04 – Zobraziť prehľad regiónov

5.2.4.1 Pozitívny scenár

Moduly	Mobilná a webová aplikácia.
Prerekvizity	Používateľ je prihlásený do systému a má priradený minimálne región R.
Scenár	1. Otvoriť mobilnú resp. webovú aplikáciu.
Postrekvizity	Používateľ vidí minimálne región R a jeho bezpečnostný stav. Vo webovej aplikácií má možnosť tento stav rovno zmeniť.
Výsledok	ÚSPECH

5.2.4.2 Negatívny scenár

Moduly	Mobilná a webová aplikácia.
Prerekvizity	Používateľ je prihlásený do systému a nemá priradený žiaden región.
Scenár	1. Otvoriť mobilnú resp. webovú aplikáciu.
Postrekvizity	Používateľ vidí správu, že nemôže meniť bezpečnostný stav žiadneho regiónu.
Výsledok	ÚSPECH

5.2.5 TC05 – Zobraziť detail regiónu

5.2.5.1 Pozitívny scenár

Moduly	Mobilná aplikácia.
Prerekvizity	Používateľ je prihlásený do systému a má priradený minimálne región R.
Scenár	1. Kliknúť na región R v prehľade regiónov.
Postrekvizity	Používateľ vidí atributy regiónu R a má možnosť zmeniť jeho bezpečnostný stav.
Výsledok	ÚSPECH

5.2.6 TC06 – Zmeniť bezpečnostný stav regiónu

5.2.6.1 Pozitívny scenár

Moduly	Mobilná a webová aplikácia.
Prerekvizity	Používateľ je prihlásený do systému a má priradený minimálne región R s bezpečnostným stavom S_1.
Scenár	1. Vybrať bezpečnostný stav $S_2 \neq S_1$. 2. Potvrdiť zmenu bezpečnostného stavu.
Postrekvizity	• Systém zmenil bezpečnostný stav regiónu R. • Používateľ vidí zmenu v prehľade regiónov. • Vo webovej aplikácií sa zmena prejaví v tabuľke bezpečnostných stavov v lište.
Výsledok	ÚSPECH

5.2.7 TC07 – Priradiť používateľa k sektoru

5.2.7.1 Pozitívny scenár

Moduly	Webová aplikácia.
Prerekvizity	• Používateľ je prihlásený do systému a má rolu administrátora. • V systéme je zaregistrovaný používateľ *P*. • V systéme sa nachádza región *R*.
Scenár	1. Vybrať používateľa *P* na riadku regiónu *R*. 2. Potvrdiť priradenie používateľa.
Postrekvizity	• Systém priradil používateľa *P* k regiónu *R*. • Používateľ *P* môže meniť bezpečnostné stavy regiónu *R*.
Výsledok	ÚSPECH

5.2.8 TC08 – Vygenerovať tabuľku bezpečnostných stavov

5.2.8.1 Pozitívny scenár

Moduly	Backend.
Prerekvizity	—
Scenár	1. Odoslať HTTP GET požiadavku na zdroj */table*.
Postrekvizity	Systém v odpovedi vráti HTML kód obsahujúci tabuľku regiónov a ich aktuálnych bezpečnostných stavov.
Výsledok	ÚSPECH

Kapitola 6

Inštalačná príručka

6.1 Backend

6.1.1 Databáza

- **Prerekvizity:**
  - Pre intuitívne sprevádzkovanie databázy je doporučené nainštalovať *MySQL Workbench*.
  - Inštalátor obsahuje všetky potrebné komponenty vrátane *MySQL Server*.
  - Návod predpokladá nainštalovaný *MySQL Workbench* a *MySQL Server* s prednastavenou možnosťou *Developer default* a ponechanými predvolenými hodnotami.
- Spustíme *MySQL Server* a *MySQL Workbench*.
- V *MySQL Workbench* otvoríme súbor *database/model.mwb* pomocou *File → Open model...* V prípade, že aplikácia vyhodí chybu „*Error unserializing GRT data, string too long*“, reštartujeme *MySQL Workbench*.
- Po načítaní modelu zvolíme možnosť *Database → Forward engineer* a preklikáme sa jednotlivými obrazovkami sprievodcu. Všetky hodnoty sprievodcu ponecháme predvolené.
- Zavrieme otvorený model a EER diagram.
- Pomocou *File → Open SQL script...* otvoríme súbor *database/data.sql*, ktorý obsahuje číselník regiónov a bezpečnostných stavov. Po spustení skriptu je databáza pripravená na použitie.

6.1.2 Server

- **Prerekvizity**:
  - Návod predpokladá nainštalovaný *Node.js* a správcu balíčkov *npm* v systéme.
- V termináli otvoríme adresár *server* a nainštalujeme potrebné závislosti príkazom:

  ```
  $ npm install
  ```

- V súbore *server/src/db.js* nastavíme adresu, na ktorej beží spustená databáza. V prípade, že databáza beží na tom istom počítači, na ktorom inštalujeme *Node.js* aplikáciu, ponecháme vyplnené *localhost*.
- V tom istom súbore nastavíme používateľa a heslo, ktoré je nastavené v *MySQL* databáze.
- V súbore *server/src/config.json* nastavíme IP adresu a port počítača, na ktorom bude spustený *Node.js* server.
- V tom istom súbore voliteľne nastavíme tajomstvo, ktoré bude použité pri podpisovaní *JWT*.
- V termináli otvoríme adresár *server/src* a spustíme server príkazom:

  ```
  $ node server.js
  ```

6.1.3 Webová aplikácia

- **Prerekvizity**:
  - Návod predpokladá nainštalovaného správcu balíčkov *npm* v systéme.
- V termináli otvoríme adresár *client_web* a nainštalujeme potrebné závislosti príkazom:

  ```
  $ npm install
  ```

- V súbore *client_web/src/config.json* nastavíme IP adresu a port serveru, na ktorom bude nasadená webová aplikácia.
- V súbore *client_web/src/service.js* je nutné prepísať adresy ciest k REST zdrojom na adresu servera, na ktorom beží backend.
- V prípade, že backend beží na HTTPS serveri, je v tomto súbore nutné prepísať „http“ na „https“ u každej cesty k REST zdroju.
- Pre testovacie účely je možné spustiť aplikáciu príkazom:

  ```
  $ npm run serve
  ```

6.1.4 Mobilná aplikácia

- **Prerekvizity**:
  - Návod predpokladá nainštalovaný *Gradle* a *Android SDK*.
- Adresár *client_android* obsahuje inicializovaný *Gradle* projekt.
- V súbore *client_android/app/src/main/java/com/example/svar/Service.kt* je nutné v metóde *.baseUrl* nutné prepísať IP adresu na adresu servera, na ktorom beží backend.
- V prípade, že backend beží na HTTPS serveri, je v tomto súbore nutné prepísať „http“ na „https“.
- Testovaciu verziu aplikácie je možné nainštalovať priamo do smartfónu.
- Pre nainštalovanie testovacej verzie aplikácie do smartfónu je najprv v smartfóne nutné povoliť *USB debugging* v nastaveniach pre vývojárov. Následne po pripojení zariadenia k počítaču v termináli otvoríme adresár *client_mobile* a spustíme príkaz:

  ```
  $ gradlew installDebug
  ```

- Pre vytvorenie a distribúciu APK balíčka je nutné mať balíček podpísaný privátnym kľúčom. Podrobný návod je k dispozícii na: `https://developer.android.com/studio/build/building-cmdline#sign_cmdline`

6.1.5 Pridanie administrátora

- Po sprevádzkovaní niektorej z frontendových aplikácií je nutné zaregistrovať prvého používateľa, ktorému priradíme rolu administrátora. Inak by bežných používateľov nemal kto priradiť k regiónom.
- Zmena roly nie je podporovaná frontendom, vykonáva sa manuálne odoslaním dotazu do databázy.
- Po zaregistrovaní prvého používateľa je nutné mu priradiť rolu administrátora nasledujúcim dotazom: (predpokladáme, že má ID = 1)

  ```
  $ UPDATE svar.user SET role="ADMIN" WHERE id=1;
  ```

- Následne sa môžu registrovať bežní používatelia a administrátor ich môže priradiť k regiónom.

Záver

Cieľom práce bolo pre spolok Česká civilní ochrana obyvatelstva (CESCOO) vytvoriť systém pre zadávanie bezpečnostných stavov SVAR (Systému včasného varovania). Práca si kládla za cieľ vytvoriť systém, ktorý by zlepšil súčasný proces zadávania bezpečnostných stavov. Tento cieľ bol úspešne naplnený.

Prvým krokom bola analýza súčasneho stavu a identifikácia slabých miest v procese zadávania bezpečnostných stavov. Nasledovala analýza požiadaviek zadávateľa a definovanie prípadov použitia. Na základe analýzy bol navrhutý konceptuálneho modelu nového systému. Po dôkladnej analýze prišla na rad samotná implementácia jednotlivých častí systému – databázy, backendu a frontendových aplikácií. Po implementácií boli zostavené funkčné a integračné testy pozostávajúce z testovacích prípadov. Systém bol kontinuálne testovaný v priebehu implementácie a následne aj po nej. Účelom funkčného testovania bolo overiť, či systém spĺňa funkčnú špecifikáciu. Práca ďalej obsahuje popis používateľského rozhrania frontendových aplikácií. Postup na sprevádzkovanie systému je popísaný v inštalačnej príručke.

Výsledkom práce je moderný, ľahko rozšíriteľný systém, ktorý je nenáročný na prevádzku a systémové prostriedky, a je implementovaný v populárnych, aktívne vyvíjaných technológiách. Na záver nezostáva nič, než vyjadriť nádej, že systém bude pre členov spolku CESCOO prínosom.

Literatúra

[1] SYSTÉM VČASNÉHO VAROVÁNÍ (SVAR). Dostupné z https://coobra.cz/index.php/kategorie-prj/83-syst%C3%A9m-v%C4%8Dasn%C3%A9ho-varov%C3%A1n%C3%AD-svar, [Online, cit. 2020-09-28].

[2] Browser Market Share Czech Republic. Dostupné z https://gs.statcounter.com/browser-market-share/desktop/czech-republic/#yearly-2020-2020-bar, [Online, cit. 2020-10-04].

[3] Lardinois, F.: Kotlin is now Google's preferred language for Android app development. Dostupné z https://techcrunch.com/2019/05/07/kotlin-is-now-googles-preferred-language-for-android-app-development, [Online, cit. 2020-10-05].

Dodatok **A**

Zoznam použitých skratiek

API Application programming interface

APK Android application package

CESCOO Česká civilní ochrana obyvatelstva

CPU Central processing unit

ER Entity relation

eMMC Embedded multi-media controller

HTML Hypertext markup language

HTTP Hypertext transfer protocol

HTTPS Hypertext transfer protocol secure

ID Identification

IP Internet protocol

IT Integration test

JSON Javascript object notation

JWT JSON web token

PC Personal computer

RAM Random access memory

REST Representational state transfer

SQL Structured query language

SSD Solid-state drive

SSL Secure sockets layer

SVAR Systém včasného varovania

TC Test case

UC Use case

UI User interface

USB Universal serial bus

Dodatok **B**

Obsah priloženej pamäťovej karty

- `client_android` ..... zdrojové súbory mobilnej aplikácie
- `client_web` ..... zdrojové súbory webovej aplikácie
- `database`
  - `data.sql` ..... SQL skript číselníkov
  - `model.mbw` ..... ER model
- `server` ..... zdrojové súbory backendu
- `thesis` ..... zdrojové súbory textovej časti práce