

Semestrální projekt

Parkoviště – zadání č. 2

Jméno: Jaroslav Mihál, Jáchym A. Kolebacz, Alec Smyček, Mariusz Lisztwan

Login: MIH0051, KOL0472, SMY0017, LIS0112

Typ studia: prezenční

Cvičící: Ing. Jakub Němčík

Předmět: Řídicí systémy s počítači

Datum: 26. listopadu 2022

Obsah

Seznam obrázků	2
Seznam tabulek	3
1 Zadání	4
1.1 Specifikace parkoviště	4
1.2 Specifikace dokumentace	4
2 Analýza problému	7
3 Dokumentace	9
4 Softwarová analýza	12
4.1 Komunikace	12
4.2 Analýza dokumentů	13
4.3 Analýza obsahu a struktury informací	13
4.4 Analýza toku informací	13
4.5 Analýza slabých míst	13
5 Systémová specifikace	14
5.1 Výchozí situace a cíle	15
5.2 Vztah okolí k provozování systému	15
5.3 Funkční požadavky	15
5.4 Nefunkční (ostatní) požadavky	15
5.5 Uživatelská rozhraní	15
5.6 Chování za chybových situací	15
5.7 Požadavky na dokumentaci	15
5.8 Předávací podmínky	15
6 UML analýza	17
7 Závěr	19
Literatura	20
8 Přílohy	21

Seznam obrázků

2.1	Mapa parkoviště	7
3.1	Prvotní návrh	10
3.2	Prvotní návrh vizualizace	11
4.1	Komunikace	13
5.1	Vizualizace pro administrátora (v plné velikosti viz Přílohy)	15
5.2	Vizualizace pro uživatele (v plné velikosti viz Přílohy)	16
6.1	Sekvenční diagram nastavení intenzity	17
6.2	Use Case diagram	18
8.1	Prvotní návrh vizualizace	22
8.2	Prvotní návrh vizualizace	23

Seznam tabulek

1 Zadání

1.1 Specifikace parkoviště

Ovládání světel lampy na základě různých enviro podmínek. Aplikace bude měřit nějaké rozumné veličiny (tmu, déšť, vítr, smog) a z nich se vytvoří požadavek na přiměřené osvětlení parkoviště. Toto bude zasláno rozhodující aplikaci, jako v předchozím případě.

1.2 Specifikace dokumentace

1) Analýza technologického řešení

Na základě zadání je potřeba analyzovat hardware dané technologie, což znamená:

- volbu typu systému – distribuovaný, centralizovaný – třeba neopomenout důvod;
- volbu senzorů;
- zapojení – komunikace, řídících jednotek, silového vedení, aktuátorů, senzorů;
- volbu silových jednotek, aktuátorů;
- volbu vizualizační prostředí;
- sledování a ukládání dat;
- a jiné.

Výsledkem je sada výkresů subcelků, seznamy komponent, dokumentace komponent, zdroje informací.

2) Dokumentace technologie

Zde je očekáván výstup ve formě výkresu celé technologie (případně její části). Cílem je zachytit podstatu celku a jeho částí, najít případné mezery v rámci komunikace.

Poznámka: K tvorbě výkresů lze využít vektorové programy (Autocad, CorelDraw, atd.), tužku s papírem a scannerem či jiné projektové aplikace.

3) Softwarová analýza

Součástí softwarové analýzy je:

- Obecná analýza
 - slovní forma
- Analýza struktury vnějšího prostředí

- interakce lidí se softwarem – kdo a jak může se softwarem pracovat
- Analýza funkcí
 - funkce, které aplikace umožňuje
 - provádění funkcí – kdy a jak často se mají provádět
- Analýza komunikací
 - komunikace mezi jednotlivými částmi aplikace – jak komunikuje hlavní řídicí algoritmus s ostatními částmi aplikace
- Analýza dokumentů
 - všechny dokumenty, které jsou generovány nebo používány v aplikaci – co budou dané dokumenty obsahovat
- Analýza obsahu a struktury informací
 - typ a struktura dat v systému
 - frekvence zpracování a používané přenosy dat
 - délka uchovávání dat
- Analýza toku informací
 - toky dat mezi jednotlivými funkcemi
 - ochrana dat
- Analýza slabých míst
 - identifikace problémů, opomenucí a redundancí (funkcí i celého systému)

4) Systémová specifikace

- Výchozí situace a cíle
 - cíle a účel softwaru
 - aktuální funkcionalita – co lze nabídnout zákazníkovi
- Vztah okolí k provozování systému
 - podmínky pro provoz
 - jaká vnější data jsou potřeba
 - počet uživatelů, jejich činnosti, frekvence užití
- Funkční požadavky
 - seznam funkcí softwaru očekávané uživatelem – co očekáváme od technologie vzhledem k softwaru
- Nefunkční (ostatní) požadavky
 - požadavky na spolehlivost, přenositelnost
 - reakční časy a doba zpracování
- Uživatelská rozhraní
 - popis nedůležitějších bodů uživatelského rozhraní

- popisuje způsob a prostředky, jimiž uživatel komunikuje se systémem
- Chování za chybových situací
 - rozbor vlivů různých chyb a požadované chování systému při jejich výskytu
- Požadavky na dokumentaci
 - referenční příručka, manuál, systémová dokumentace
- Předávací podmínky
 - návrh testů a způsobu kontroly pro každý požadavek samostatně
- Přílohy
 - pojmy, bibliografie atd.

5) UML analýza

Analýza pomocí UML diagramů bude obsahovat minimálně tolik diagramů UML, kolik je studentů ve skupině (například: diagram užití, aktivitní diagram, diagram tříd, stavový diagram, sekvenční diagram, časování). Každý ze studentů tedy vytvoří alespoň jeden z těchto diagramů.

6) Výstupy projektu

Jsou očekávány dva výstupy, jež budou uloženy pomocí GIT ve vzdáleném repositáři včetně všech dodatečných souborů a příloh. Výstupy jsou:

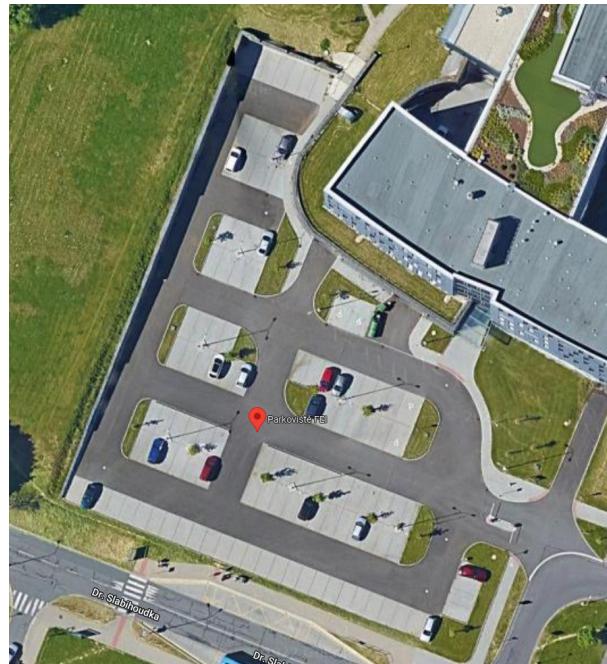
- dokument splňující veškeré body zadání 1 až 5 a obsahující titulní list, obsah, patřičné formátování, schémata, obrázky, diagramy, přílohy a self-assessment (sebehodnocení přínosu jednotlivých členů týmu – kdo co udělal);
- prezentace výsledku projektu (28.11.2022).

2 Analýza problému

Jako semestrální projekt jsme si ve skupině vybrali ovládání světel lampy na základě různých environmentálních podmínek. Konkrétně se jedná o osvětlení parkoviště mezi budovou FEI a FNO (souřadnice 49.830987838430644, 18.160041318757802) o rozloze zhruba přibližně 40x60 m. Naše řešení je centralizovaný systém s distribuovanou vizualizací. Za pomocí jednoho PC (případně i Raspberry Pi) zpracováváme data ze světel a senzorů na chtěné hodnoty a následně je posíláme jako příkazy na ovládání světel. Tento počítač pak posílá data pomocí UDP protokolu po síti, na kterou se lze připojit vizualizaci. Kdyby se kdokoliv rozhodl tento systém uplatnit a chtěl by použít bezdrátovou komunikaci, pak bude systém spíše distribuovaný, neboť hlavní počítač bude posílat pouze data systému jednotlivých světel.

Rozhodli jsme se použít senzory VISIC620 od společnosti SICK. Volba padla právě na tento typ senzoru, protože je schopen snímat v podstatě všechny změny environmentálních podmínek. Používáme konkrétně alespoň čtyři senzory pro určení viditelnosti, ze kterých se vytváří průměrná hodnota viditelnosti, dle které systém reaguje na změny. Následně bude systémově ošetřeno, aby se při případném poškození senzoru (tzn. hodnoty se budou výrazně lišit), zobrazilo upozornění ve vizualizaci.[1]

PC je propojeno se světly pomocí kabelů dle druhu světla, případně bezdrátově, dle možnosti daných světel. PC musí být připojeno k wifi za účelem vizualizace.



Obrázek 2.1: Mapa parkoviště

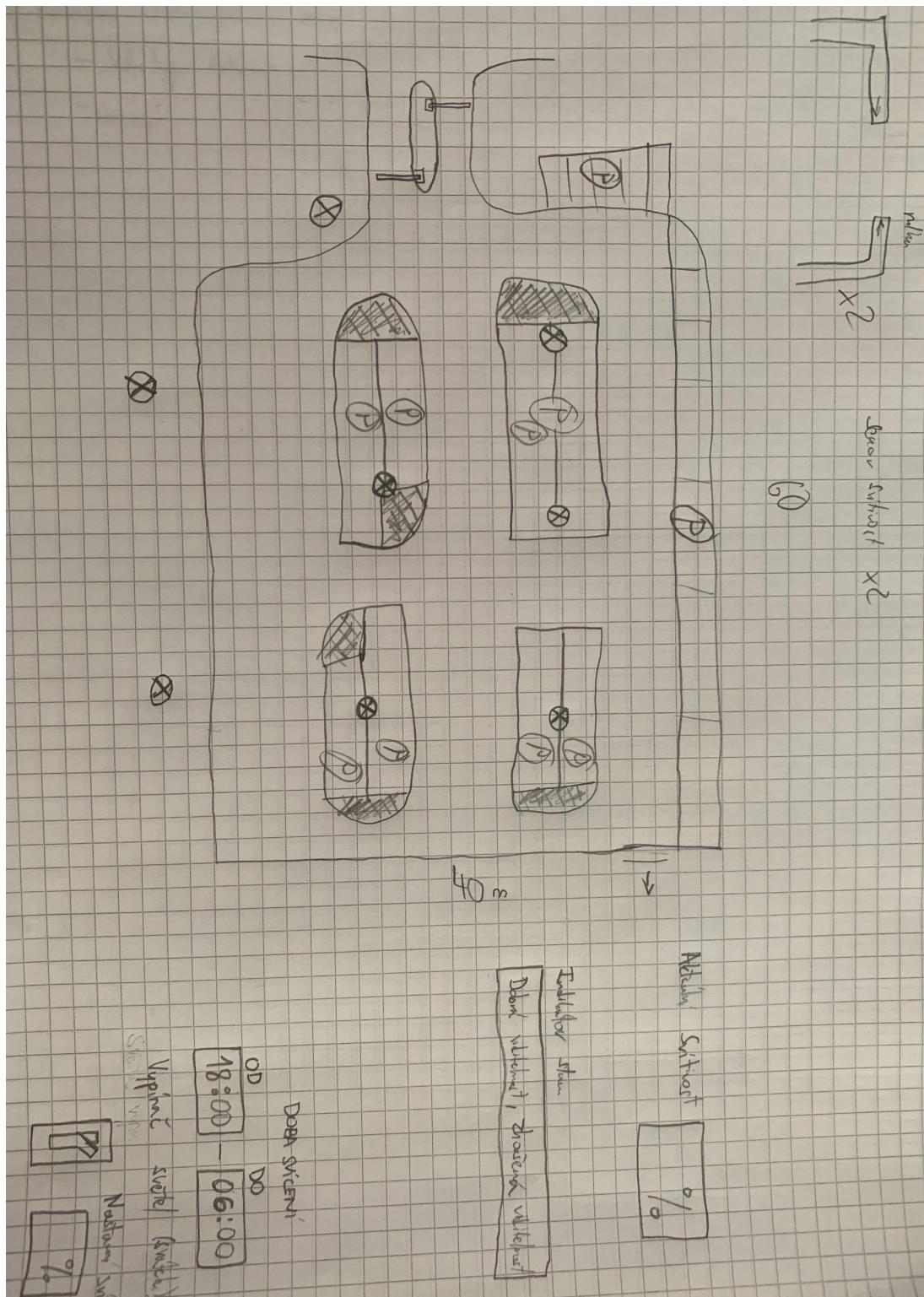
Vizualizaci aplikace jsme se rozhodli realizovat v prostředí Promotic. Budou dva účty – administrátor a uživatel. Administrátor bude mít možnost měnit nastavení svítivosti světel v procentech pro každou hodinu pomocí posuvníků.

V aplikaci (vizualizaci), v našem řešení jsme se rozhodli pro Promotic, existují dva účty. Jeden účet je administrátora, který může měnit nastavení svítivosti světel v procentech pro každou hodinu pomocí posuvníků. V případě, že se zhorší viditelnost, svítivost se dle „závažnosti“ situace procentuálně zvýší. Systém je schopen reagovat téměř okamžitě, záleží tedy na programátorovi, jakou frekvenci vyčítání dat ze senzorů zvolí. Ve druhém účtu je možné pouze sledovat stav, tedy na kolik procent daná lampa svítí. V případě například energetické krize, svátků, je možné světla vypínačem vypnout pro úsporu energie. Promotic není zásadní pro běh aplikace, můžeme využít i jiné vizualizační programy, pokud podporují komunikaci přes UDP protokol (např. LabVIEW). Ukládáme hodnoty senzorů při určování osvětlení, požadovanou intenzitu osvětlení administrátorem, tato data jsou pak sledována právě vizualizací a slouží pro zpětnou kontrolu funkčnosti.

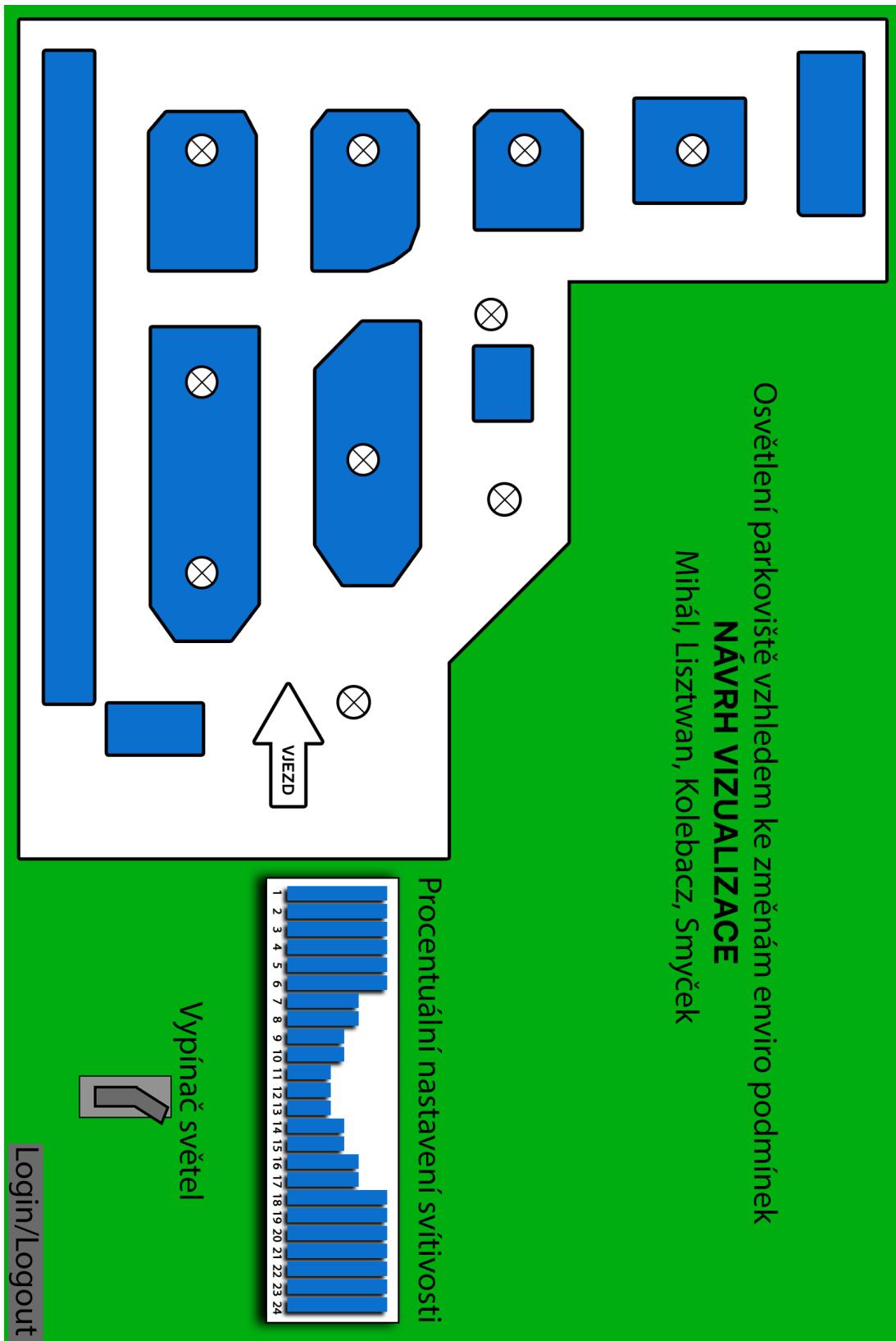
Jako potencionální chyby řešení můžou být:

- vliv záškodníků na senzory
- špatně (nedostatečně) nastavené hodnoty osvětlení ve vizualizaci adminem
- delší počáteční ladění systému

3 Dokumentace



Obrázek 3.1: Prvotní návrh



Obrázek 3.2: Prvotní návrh vizualizace

4 Softwarová analýza

Bude zapotřebí hlavní program (nepoptřebující GUI), který bude zpracovávat data získaná ze senzorů a ty pak kontrolovat, zda jsou správné. Poté přes konfigurační soubor bude periodicky načítat chtěnou hodnotu, kterou porovná s daty ze senzoru a nastaví je na světlech. Mezitím bude periodicky posílat přes UDP protokol data pro vizualizaci (vlastní komunikační protokol, který bude obsahovat, hodnoty konfiguračního souboru, poslední data ze senzorů, boolean podezdření poruchy senzoru a port pro příjem dat k ovládání přes vizualizaci).

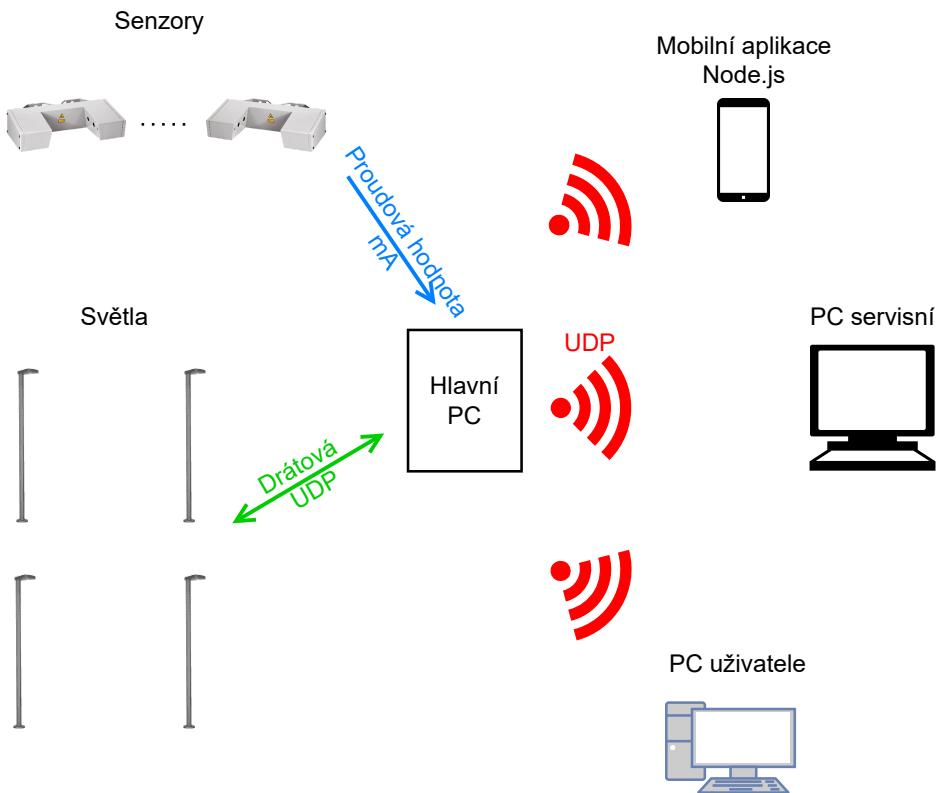
K aplikaci budou mít přístup dva typy lidí – administrátor a uživatel. Na nejnižší úrovni je uživatel, který může pouze sledovat stavy světel a senzorů. Správce (technik) navíc může vypínat a zapínat celkové osvětlení, upravovat config soubor a manuálně nastavovat intenzitu osvětlení (viz Obrázek 6.2).

Seznam funkcí, které aplikace umí:

- sběr procentuálních hodnot osvětlení
- sběr viditelnosti ze senzorů
- sběr požadovaných procentuálních hodnot skrz vizualizaci
- posílaní a následné nastavení chtěné intenzity světla
- kontrola senzorů - vždy po sběru dat senzorů
- posílaní aktuálního stavu vizualizaci
- výpočet chtěné intenzity: vezme chtěnou základní hodnotu s configu a upraví ji podle dat získaných senzory
- nastavení config souboru pomocí vizualizace
- ukladání dat
- kontrola přihlášení od vizualizace.
- přepočítání intenzity (zavolané vizualizací či změnou config souboru)

4.1 Komunikace

Komunikace s osvětlením bud' pomocí bezdrátového připojení opět pomocí UDP pravděpodobně. Jinak připojení ke světlem drátovým vedením. Komunikace s vizualizací přes UDP



Obrázek 4.1: Komunikace

Obrázek 4.1 nám popisuje komunikaci našeho řešení pro osvětlení parkoviště. Senzory posílají na hlavní PC analogovou proudovou hodnotu. Pomocí drátového UDP protokolu posílají světla aktuální hodnoty na hlavní PC, který zpětně je schopný přes UDP světla ovládat a nastavovat tedy jejich hodnotu svítivosti. Dále je schopen komunikovat s PC uživatele, který má naší aplikaci vytvořenou v prostředí Promotic, případně se servisním PC, a to bezdrátově (wifi). Dále existuje možnost pomocí Node.js komunikace s mobilním telefonem, kterou momentálně nevyužíváme.

4.2 Analýza dokumentů

4.3 Analýza obsahu a struktury informací

4.4 Analýza toku informací

4.5 Analýza slabých míst

5 Systémová specifikace

5.1 Výchozí situace a cíle

5.2 Vztah okolí k provozování systému

5.3 Funkční požadavky

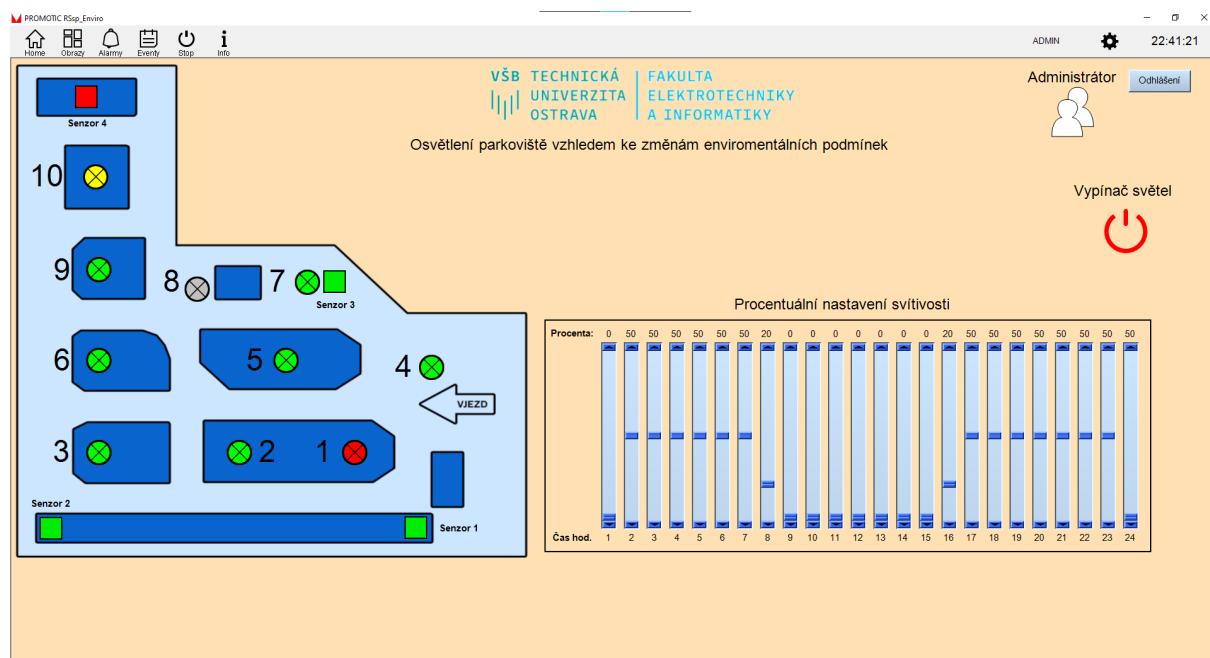
5.4 Nefunkční (ostatní) požadavky

5.5 Uživatelská rozhraní

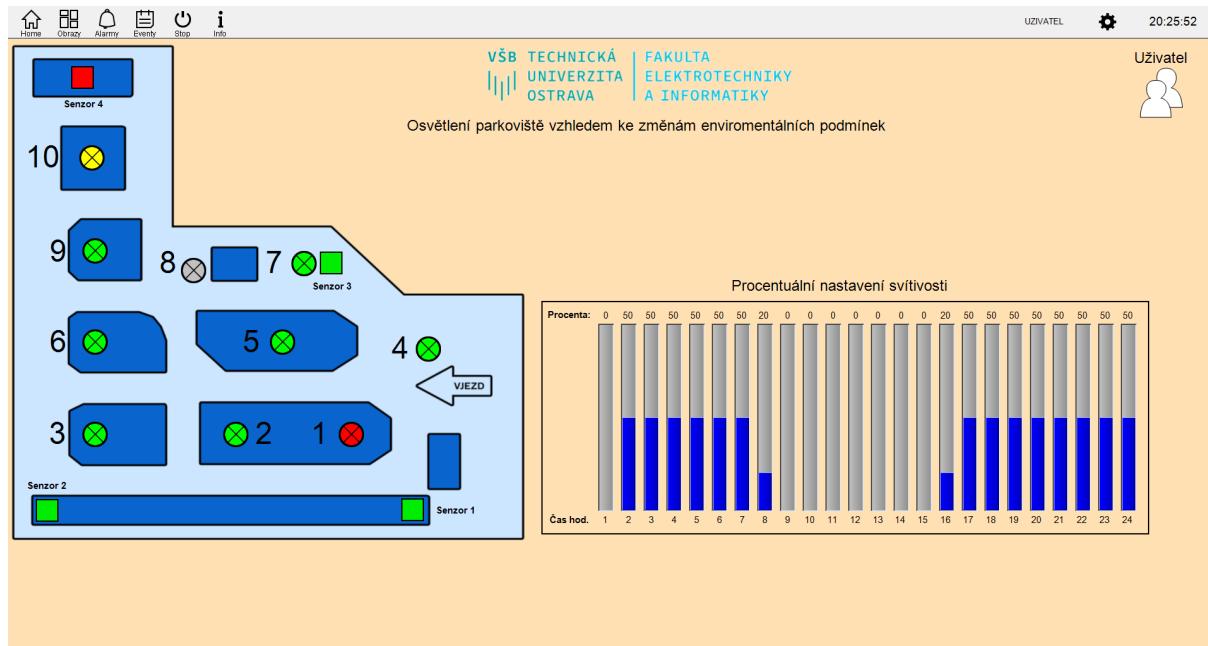
5.6 Chování za chybových situací

5.7 Požadavky na dokumentaci

5.8 Předávací podmínky

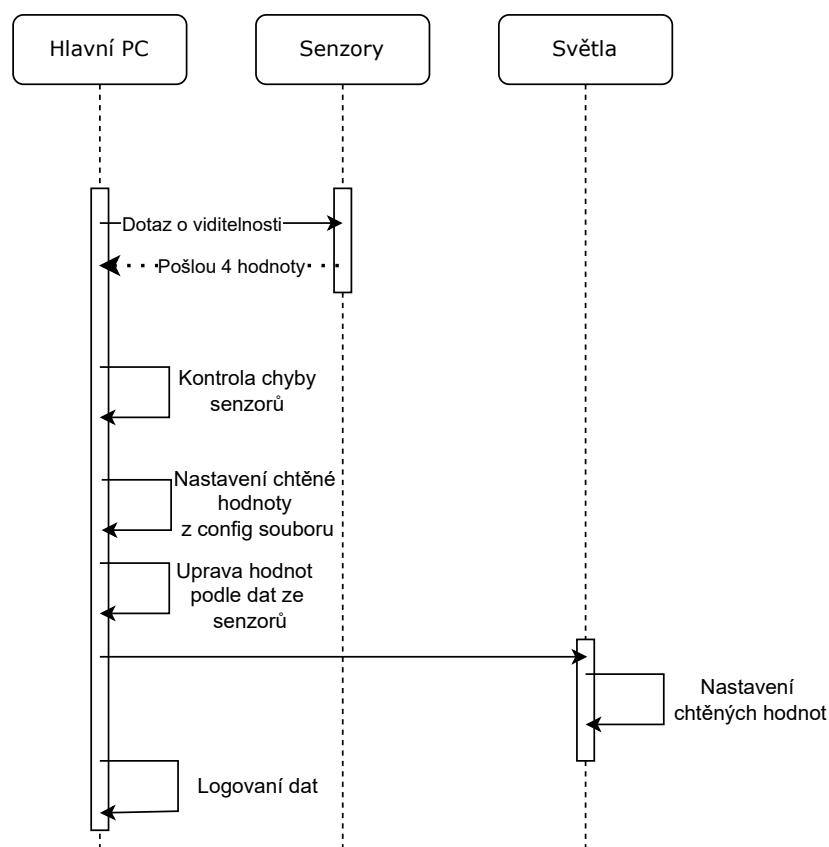


Obrázek 5.1: Vizualizace pro administrátora (v plné velikosti viz Přílohy)

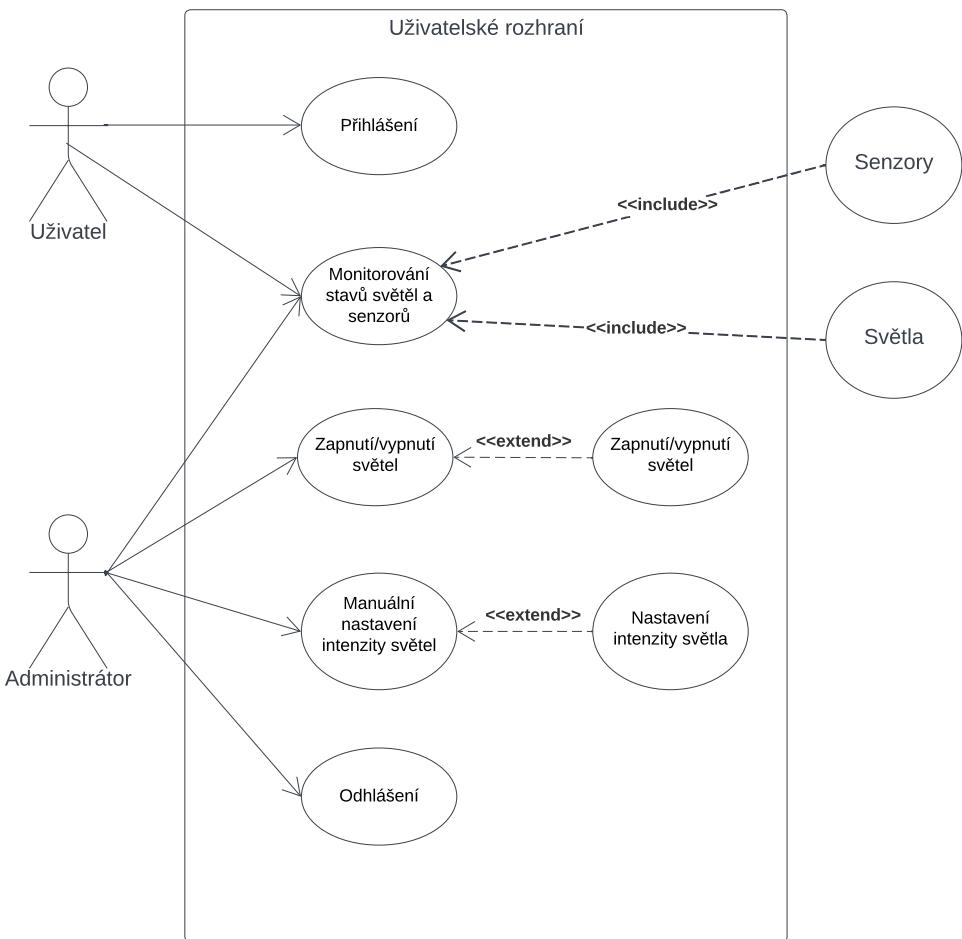


Obrázek 5.2: Vizualizace pro uživatele (v plné velikosti viz Přílohy)

6 UML analýza



Obrázek 6.1: Sekvenční diagram nastavení intenzity



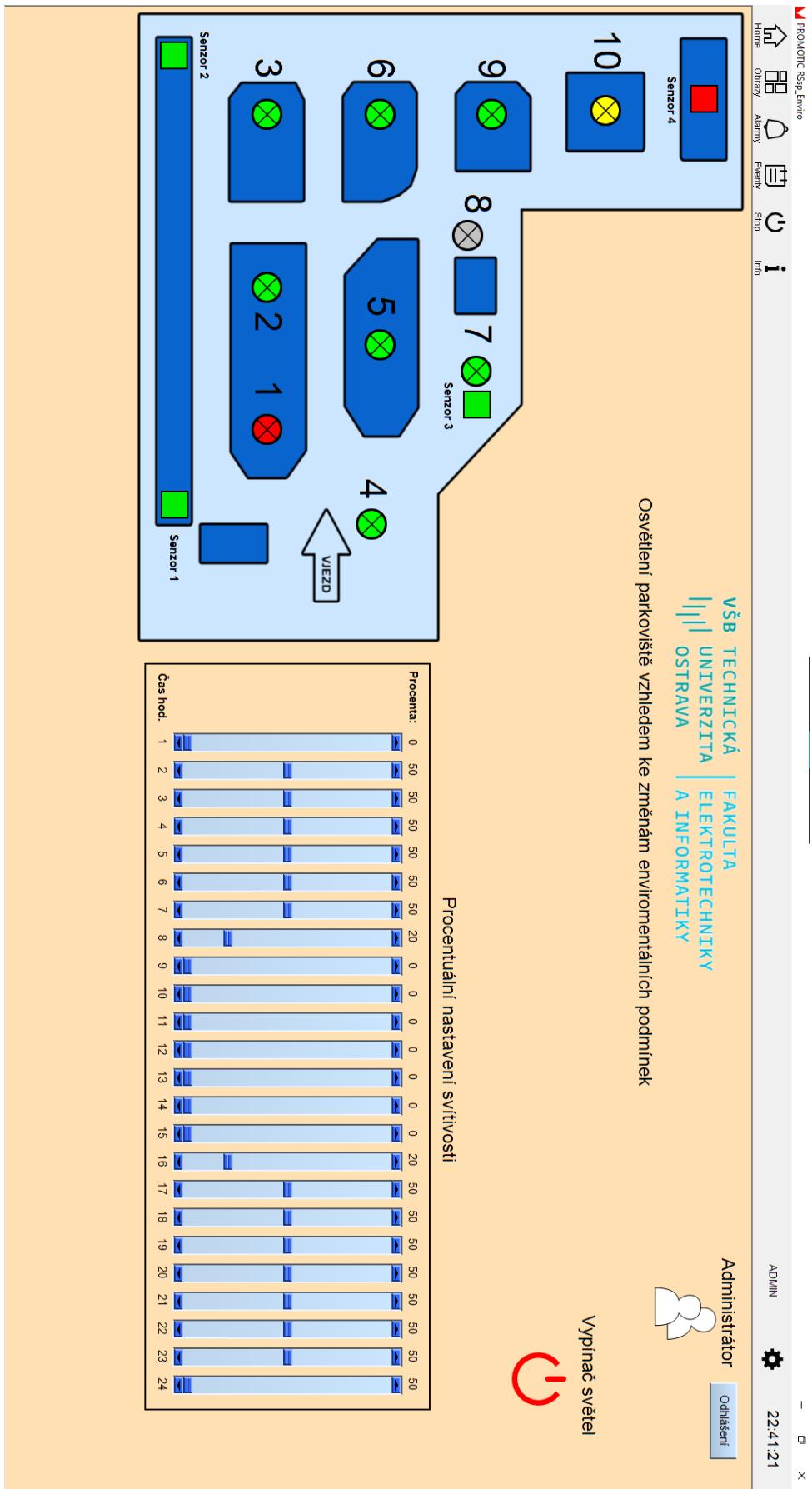
Obrázek 6.2: Use Case diagram

7 Závěr

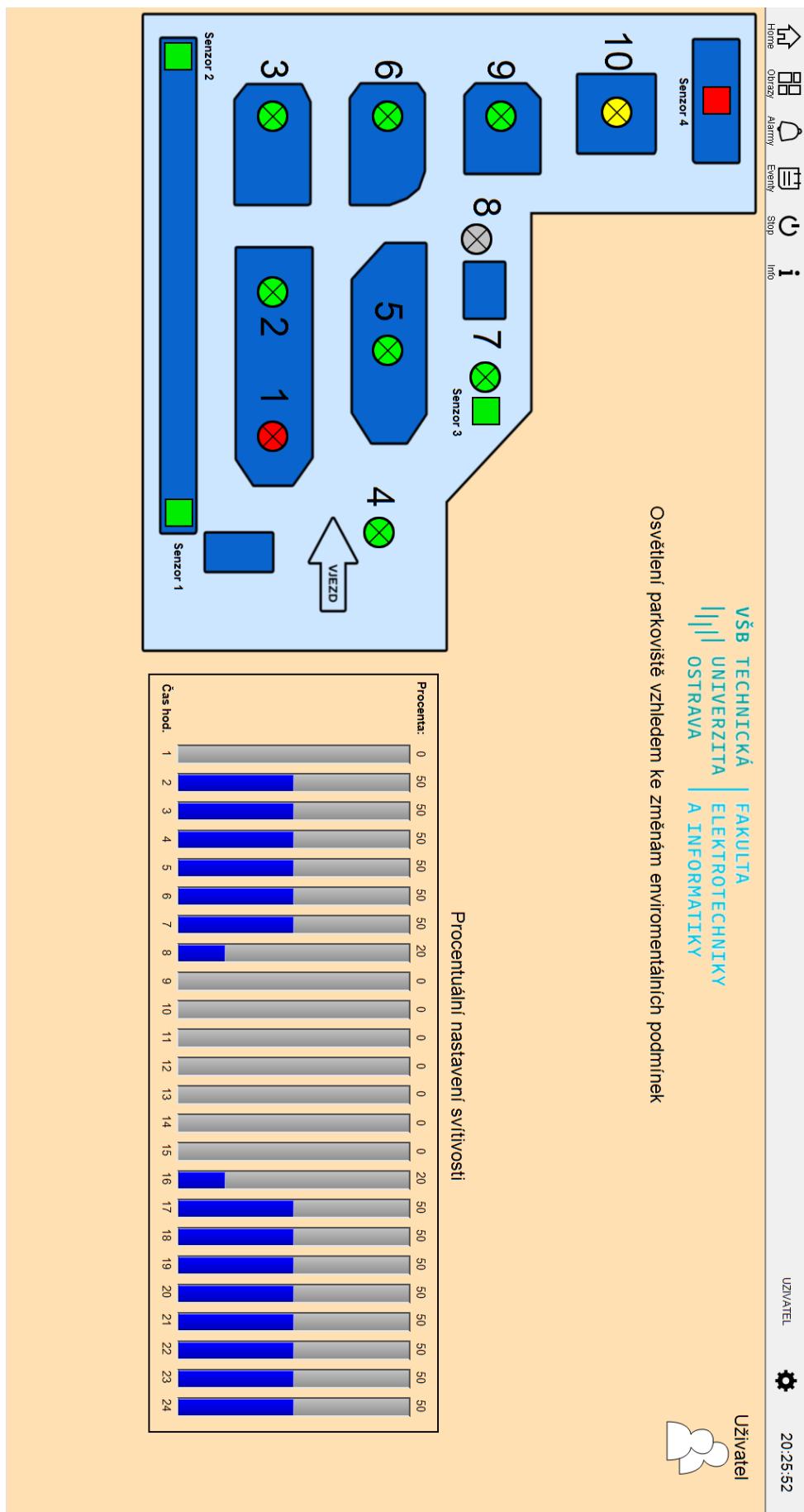
Literatura

- [1] sick.com, ed. *Přístroje pro měření viditelnosti VISIC620*. [Online]. URL: <https://www.sick.com/cz/cs/dopravní-senzory/prístroje-pro-merení-viditelnosti/vsic620/vsic620-1111/p/p494044> (cit. 2022-11-20).

8 Přílohy



Obrázek 8.1: Prvotní návrh vizualizace



Obrázek 8.2: Prvotní návrh vizualizace