

Závěrečná správa
Smelý zajko sprievodcom
Tvorba informačných systémov

Autori: Natália Ďurisová, Dominik Knechta,
Martin Mudroch, Miroslav Gregorec

Obsah

A.	Špecifikácia požiadaviek	5
1.	Úvod	5
A.1.1	Cieľ dokumentu	5
A.1.2	Rozsah projektu	5
A.1.3	Slovník pojmov	5
A.1.4	Zdroje	5
2.	Celkový opis	6
A.2.1	Kontext systému	6
A.2.2	Funkcie systému	7
A.2.3	Používateľské rozhranie	7
A.2.4	Všeobecné obmedzenia	7
A.2.5	Predpoklady a závislosti	8
A.2.5.1	Robot s nutným príslušenstvom	8
A.2.5.2	Notebook	8
A.2.5.3	Kamera	8
A.2.5.4	Ultrazvukové senzory	8
3.	Špecifikácia požiadaviek	8
A.3.1	Prechádzanie po pavilóne	8
A.3.2	Stretnutie s človekom	8
A.3.3	Navigácia	9
A.3.4	Ukončenie navigácie	9
A.3.5	Komunikácia so serverom	9
4.	Ostatné požiadavky	10
A.4.1	Požiadavky na admina	10
A.4.2	Požiadavky na používateľa	10
B.	Konceptuálna analýza	11
1.	Úvod	11
2.	Analýza používateľov	12
3.	Diagramy	13
B.3.1	Entitno-relačný diagram	13
B.3.2	Use-case diagram	14
B.3.2.1	Diagram pre používateľov webového rozhrania	14
B.3.2.2	Diagram pre používateľov robota	15

B.3.3	Stavový diagram	16
4.	Používateľské rozhranie	17
B.4.1	Spoločné rozhranie (Používateľský režim)	17
B.4.2	Administrátorský režim.....	19
C.	Analýza technológií, dekompozícia a dátový model	20
1.	Analýza technológií	20
C.1.1	Technológie na správu robota:	20
C.1.2	Technológie pre webové rozhranie :	21
2.	Dekompozícia	22
C.2.1	Komponentový diagram.....	23
C.2.2	Popis komponentov.....	23
C.2.2.1	Reakčný komponent.....	23
C.2.2.2	Komponent Jazdenie po pavilóne.....	23
C.2.2.3	Rozpoznávací komponent.....	24
C.2.2.4	Komponent komunikácia s osobou	24
C.2.2.5	Komponent plánovanie cesty	24
C.2.2.6	Komponent Navigácia.....	24
C.2.2.7	Komponent komunikácia s webom	24
D.	Návrh.....	25
1.	Triedny diagram	25
2.	Popis tried	25
D.2.1	Main:	25
D.2.2	Class SbotThread:.....	25
D.2.3	Class Jazda:	25
D.2.4	Class Stretnutie:	26
D.2.5	Class Navigácia:	26
E.	Testovacie scenáre	27
1.	Testovacie scenáre komponentov:.....	27
E.1.1	Reakčný komponent.....	27
E.1.2	Jazdenie po pavilóne	27
E.1.3	Rozpoznávací komponent.....	28
E.1.4	Komunikácia so serverom	29
E.1.5	Navigačný komponent.....	30
2.	Záver z testovania	30

F.	Zhodnotenie diela a tímovej práce	33
1.	Všeobecné zhodnotenie	33
F.1.1	Všeobecný pohľad	33
F.1.2	Komplikácie počas vývoja	33
F.1.3	Rozdelenie práce a komunikácia s tímom	34
F.1.4	Návrhy pre budúce verzie	34
2.	Zhodnotenie členmi tímu	35
F.2.1	Natália Ďurisová	35
F.2.2	Miroslav Gregorec	35
F.2.3	Dominik Knechta	35
F.2.4	Martin Mudroch	36
3.	Zhodnotenie zadávateľom	36

A. Špecifikácia požiadaviek

1. Úvod

A.1.1 Cieľ dokumentu

Tento dokument slúži ako špecifikácia na semestrálny projekt z predmetu Tvorba informačných systémov s názvom Smelý zajko sprievodcom. Cieľom špecifikácie je jasne, konkrétne a zrozumiteľne určiť požiadavky na spôsob spracovania informácií a samotnú funkcionálnosť. Témou projektu je naprogramovať robota, ktorý bude sprievodca po pavilóne informatiky na Fakulte Matematiky, Informatiky a Fyziky UK. Projekt je určený pre profesorov, študentov a zamestnancov fakulty ako aj pre bežných návštevníkov fakulty.

A.1.2 Rozsah projektu

Tento projekt slúži ako navigátor pri hľadaní miestností v informatickom pavilóne na Fakulte Matematiky, Informatiky a Fyziky UK. Jeho úlohou je správne navigovať používateľa na vybrané miesto. Výsledkom projektu bude robot, navigátor, ktorý odprevadí používateľa na vybrané miesto a to na základe zadaného čísla miestnosti zo vstupu. Robot bude prepojený s webom, na ktorom sa budú zobrazovať informácie o polohe a stave robota.

A.1.3 Slovník pojmov

Ultrazvuk – sú mechanické kmity prostredia s frekvenciou vyššou, než je schopné počuť ľudské ucho, teda približne nad 20 kHz (20000 Hz)

Webový server – Počítač, ktorý je zodpovedný za vykonávanie príkazov HTTP od klientov – programov zvaných webový prehliadač. Vykonaním požiadavky sa rozumie odoslanie webovej stránky. Webové stránky sú obvykle dokumenty HTML.

A.1.4 Zdroje

Potrebné informácie k existujúcemu robotovi:

<http://dai.fmph.uniba.sk/projects/smelyzajko/>

2. Celkový opis

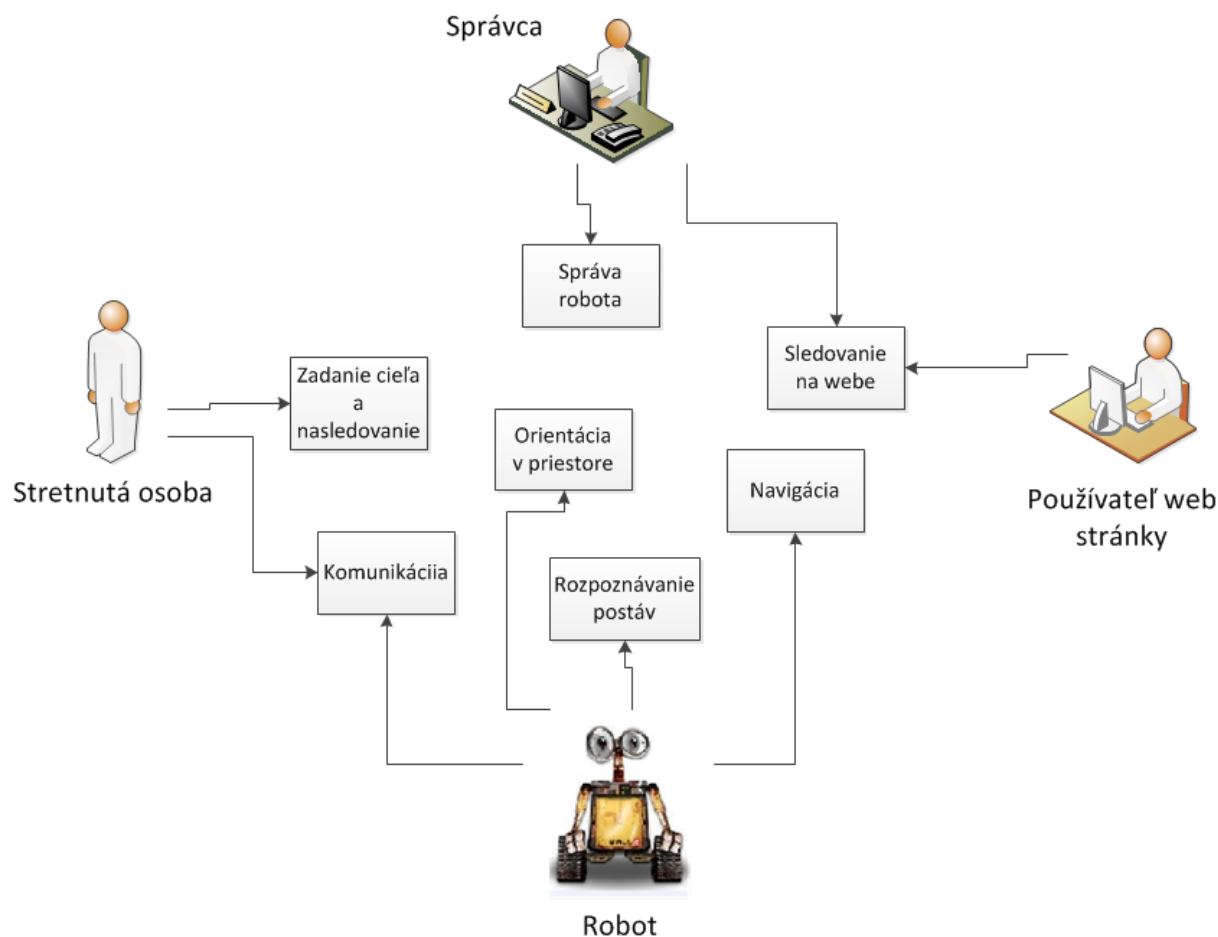
A.2.1 Kontext systému

Zapnutím robota sa spustí jeho jazdenie po pavilóne. Robot sa pohybuje dookola po pavilóne, kým nestretne osobu, ktorú rozpozná. Počas jazdy sa vyhýba kolíziám s prekážkami. Následne na to sa osobe pozdraví a povie: „Dobrý deň, môžem vám pomôcť? Zadajte prosím číslo miestnosti, ktorú hľadáte.“. Osoba zadá na numerickej klávesnici číslo a stlačí enter. Robot vypočíta najkratšiu cestu a následne na to vyzve osobu aby sa presunula zaňho a nasledovala ho. Po dosiahnutí cieľovej miestnosti, robot ukončí navigovanie a povie „Ste v cieľi. Pekný deň.“. Potom sa odznova spustí jeho „chôdza“ po pavilóne.

Počas prevádzky robot posiela informácie na server o jeho stave pomocou wifi. Rozpoznáva 4 stavy a to Zapnutý / Vypnutý / Prechádzanie / Navigovanie. Webový server si bude pamätať dennú históriu robota a to čas začatia navigácie, čas skončenia navigácie a cieľová miestnosť.

A.2.2 Funkcie systému

Prehľad funkcií je zobrazený na **Obrázok 1 Use-case diagram**



Obrázok 1 Use-case diagram

A.2.3 Používateľské rozhranie

Projekt zahŕňa viacero rozhraní. Webové rozhranie, ktoré má dva typy užívateľov a to bežného používateľa a admina. Ďalším rozhraním je rozhranie medzi robotom a náhodným okoloidúcom, ktoré spočíva vo zvukovej komunikácii robota s človekom a taktiež externá numerická klávesnica, ktorá slúži na zadanie čísla miestnosti.

A.2.4 Všeobecné obmedzenia

Priestorové obmedzenia umožňujú robotovi chodiť len po vonkajšom obvode pavilónu informatiky. Ďalej sa berie ohľad na schody, takže ich vynechá z trasy. A samozrejme robot nie je vodotesný.

A.2.5 Predpoklady a závislosti

A.2.5.1 Robot s nutným príslušenstvom

Základ robota tvorí 8-bitový jednočipový mikropočítač (AVR ATmega128), na ktorý sú pripojené dva radiče HB25, ktoré slúžia na ovládanie motorov. Na dvoch kolesách sú pripojené encodery slúžiace na počítanie otáčok, podľa ktorých sa určuje vzdialenosť, ktorú prejde a rýchlosť, ktorou sa pohybuje. Robot je napájaný 12V olovenou batériou, ktorej výdrž je približne jedna až dve hodiny.

A.2.5.2 Notebook

Robota riadi počítač, ktorý je jeho nutnou súčasťou. Robot je riadený programom, ktorý na ňom beží. Operačný systém notebooku je Linux. Súčasťou notebooku je Wifi, ktorá je potrebná pri komunikácii so serverom.

A.2.5.3 Kamera

K vybaveniu robota, patrí kamera (Camcorder Panasonic SDR-T50 alebo USB kamera), ktorá sníma prostredie a pre našu potrebu bude slúžiť na rozpoznávanie postáv.

A.2.5.4 Ultrazvukové senzory

Senzory využívajú ultrazvuk na určenie vzdialenosti okolitých prekážok. Robot používa päť senzorov, pomocou ktorých sa snaží držať v primeranej vzdialenosti od prekážok.

3. Špecifikácia požiadaviek

A.3.1 Prechádzanie po pavilóne

- A. Po spustení robota, robot začne jazdiť po pavilóne. Jeho trasa bude len po vonkajšom obvode pavilónu informatiky.
- B. Robot sa pomocou ultrazvukových senzorov snaží držať v strede medzi stenami.
- C. Počas jazdenia musí eliminovať prípadné kolízie s prekážkami, ktoré zachytí ultrazvukovými senzormi. Čo znamená zastaviť pred prekážkou a čakať pokiaľ nie je cesta pred ním opäť voľná.

A.3.2 Stretnutie s človekom

- A. Počas prechádzania robot neustále vyhľadáva ľudské postavy a vníma priestor pred sebou pomocou kamery

- B. Po vyhľadání postavy sa robot zastaví, pozdraví a opýta sa či môže pomôcť. Poprosí osobu aby zadala číslo hľadanej miestnosti na numerickej klávesnici.
- C. Robot po zadání cieľovej miestnosti prepočíta najkratšiu cestu zo svojej aktuálnej pozície. Následne na to sa vydá správnym smerom a vyzve osobu aby sa presunula zaňho a nasledovala ho.
- D. Pokiaľ do minúty od výzvy k zadaniu čísla miestnosti nedostane číslo, vráti sa znova do fázy jazdenia.

A.3.3 Navigácia

- A. Robot ide cestou, ktorú si sám zvolil.
- B. V stave navigácie nerozpoznáva ďalšie osoby.
- C. Musí sa držať v strede chodby.
- D. Počas navigácie by nemal naraziť do prekážok.

A.3.4 Ukončenie navigácie

- A. Keď sa robot dostane do okruhu 2 metrov od hľadanej miestnosti, oznámi osobe, že je v cieľi.
- B. Znova sa vráti do stavu prechádzania.

A.3.5 Komunikácia so serverom

- A. Robot bude komunikovať s webovým serverom a podávať informácie o jeho stave.
- B. Server prijíma informáciu o štyroch stavoch. Stavy sú nasledovné: Zapnutý/Vypnutý/Prechádzanie/Navigovanie.
- C. Admin bude mať možnosť prihlásiť sa a spustiť/zastaviť robota.
- D. Po otvorení stránky sa zobrazí stav, v ktorom sa robot nachádza, aktuálne zameraný cieľ a história vykonaných požiadaviek. Tvar záznamu bude nasledovný: čas začatia navigovania, čas skončenia navigovania, štart a cieľ navigovania.
- E. Admin ma možnosť vymazať históriu vykonaných požiadaviek robota.

4. Ostatné požiadavky

A.4.1 Požiadavky na admina

Kontrola baterky počas prevádzky robota.

A.4.2 Požiadavky na používateľa

Používateľ musí dodržiavať podmienky pre správny chod robota t.j. nebude mu stáť v ceste a inak brániť v pohybe.

B. Konceptuálna analýza

1. Úvod

Cieľom tohto dokumentu je špecifikovať návrh softvéru k projektu Smelý zajko sprievodcom

Dokument je štruktúrovaný do troch logických častí, ktoré sú:

- Konceptuálna analýza
Cieľom konceptuálnej analýzy je podľa katalógu požiadaviek schváleného zadávateľom projektu - analyzovať používateľov systému, prostredníctvom diagramov prezentovať funkcionality systému (robota) a predviesť prvotný návrh užívateľského prostredia.
- Analýza technológií, dekompozícia a dátový model
Cieľom tohto dokumentu je analyzovať a popísať použité technológie, ktoré budú použité pri tvorbe projektu smelý zajko sprievodcom. Dané technológie budú v dokumente popísané slovne a taktiež znázornené pomocou komponentového diagramu. V dokumente sú okrem technológií na správu robota, popísané aj technológie použité pre webové prostredie a vzájomnú komunikáciu medzi robotom a webom.
- Návrh

2. Analýza používateľov

Používateľov tohto projektu možno rozdeliť na dve skupiny :

- Používatelia webovej stránky
- Používatelia robota

Používatelia webovej stránky:

1) Anonymní používatelia

sú používatelia, ktorí získajú ľubovoľným spôsobom odkaz na stránku. Títo budú mať obmedzené práva – nebudú môcť ovládať robota. Cieľom anonymných používateľov je oboznámiť sa na stránke s projektom, prezerať si históriu využitia robota a zistiť či sa aktuálne robot používa.

2) Admin

Administrátor bude manuálne vložený do systému, pričom po prihlásení bude mať oproti anonymnému používateľovi možnosť zapnúť a vypnúť robota na diaľku či zrušiť aktuálnu navigáciu.

Používatelia robota :

1) Stretnutá osoba

Je používateľ, ktorý stretne robota pri fáze prechádzania a následne po vyzvaní robotom využije možnosť navigácie po zadaní miestnosti na numerickej klávesnici.

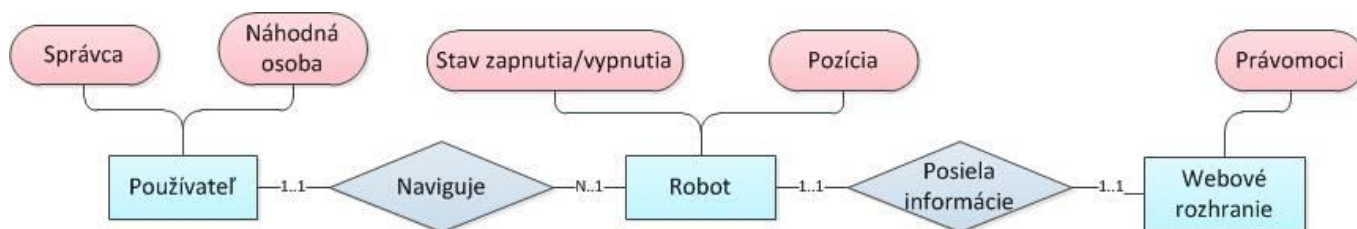
2) Správca

Je osoba, ktorá zapne robota a uvedie ho do chodu vo svojom pavilóne.

Jednotlivé kategórie používateľov webovej stránky spolu s akciami, ktoré môžu vykonávať, je možné názorne vidieť v kategórii Diagramy, čiast' use-case diagram.

3. Diagramy

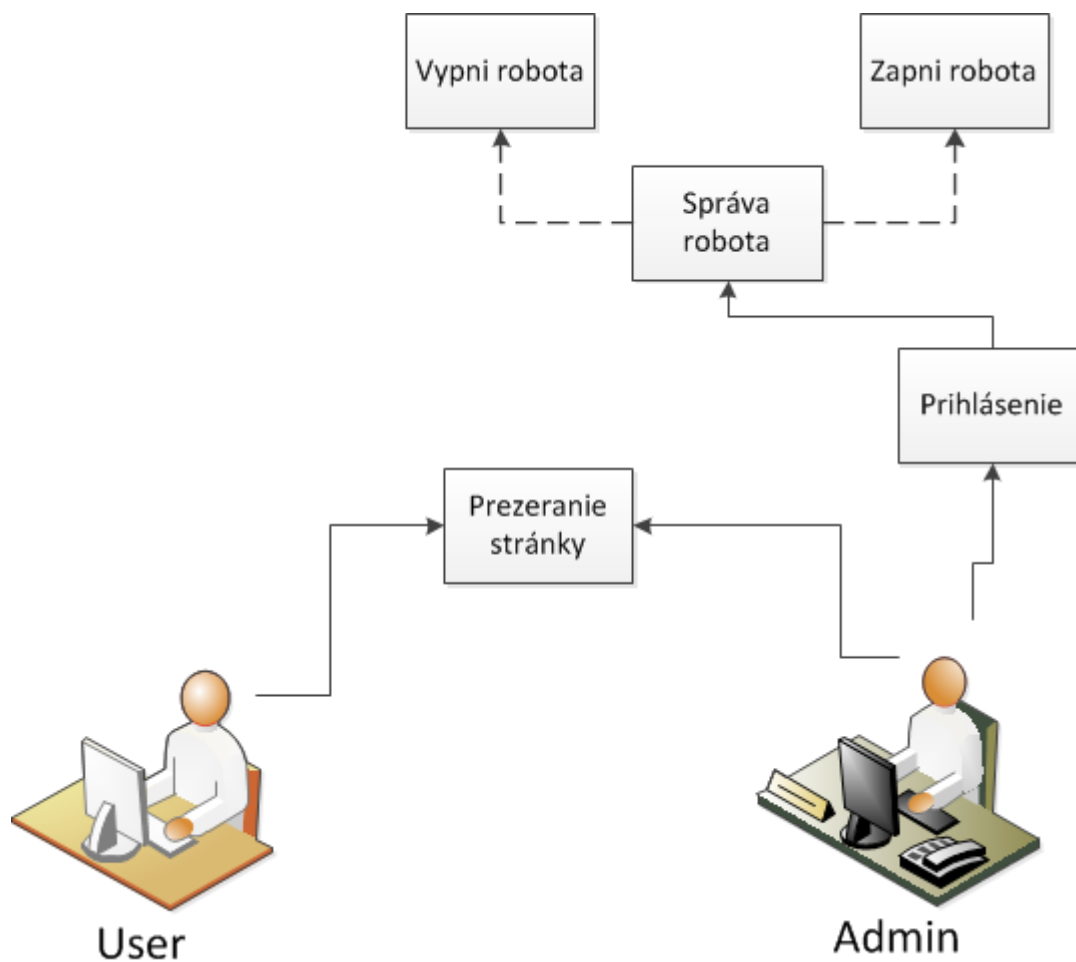
B.3.1 Entitno-relačný diagram



Obrázok 2 Entitno-relačný diagram

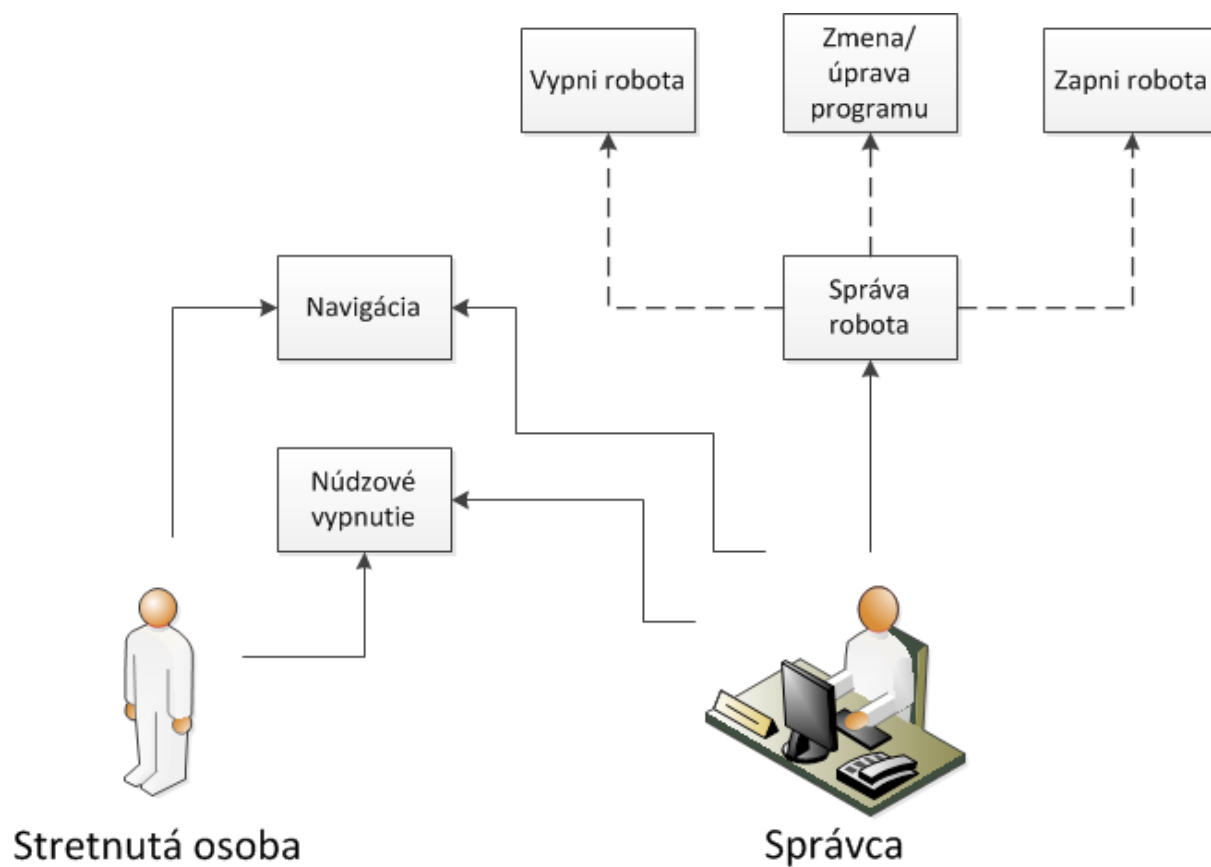
B.3.2 Use-case diagram

B.3.2.1 Diagram pre používateľov webového rozhrania



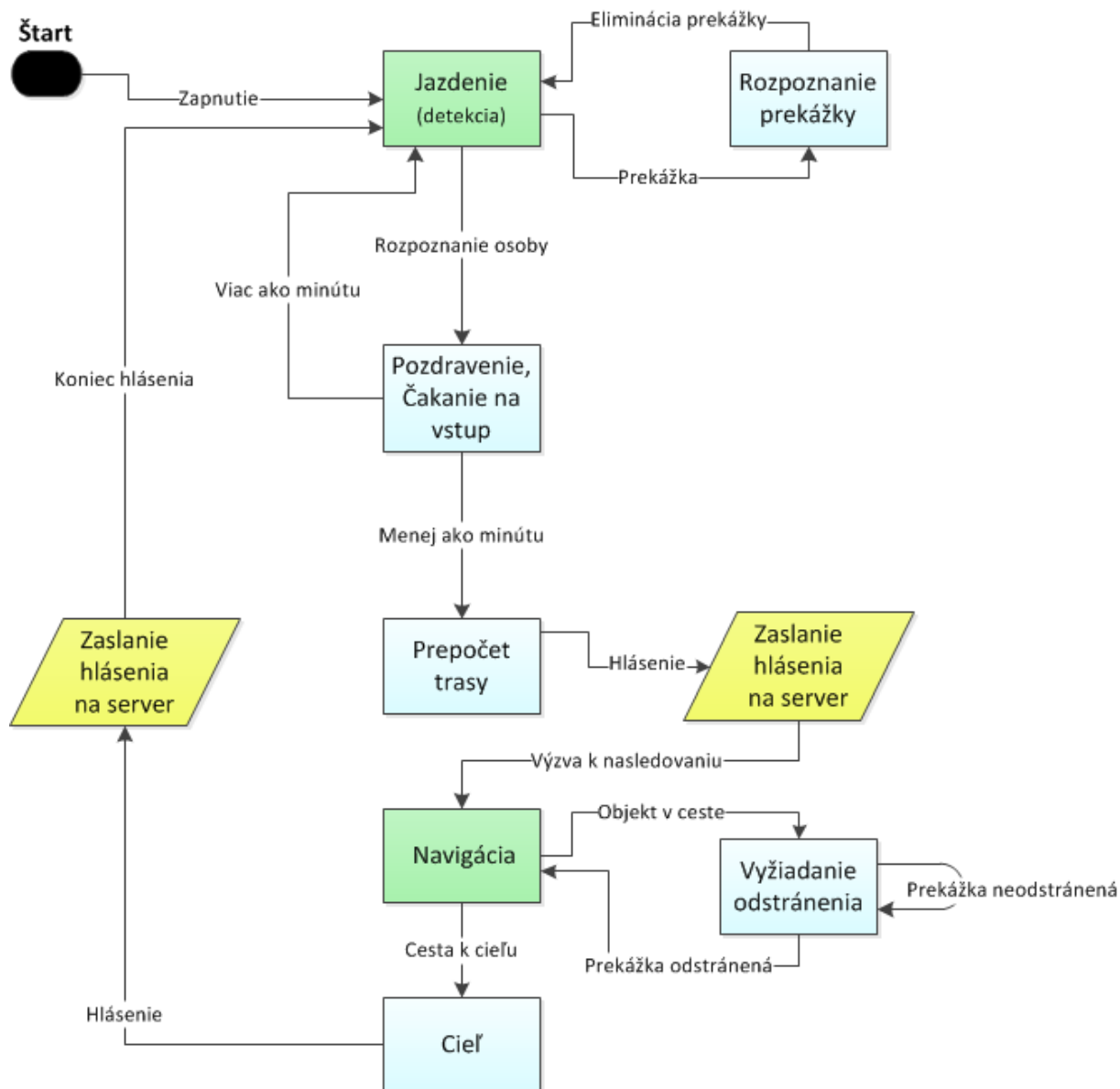
Obrázok 3 Use Case diagram pre webové rozhranie

B.3.2.2 Diagram pre používateľov robota



Obrázok 4 Use Case diagram pre robota

B.3.3 Stavový diagram



Obrázok 5 Stavový diagram robota

4. Používateľské rozhranie

V tejto časti sa venujeme predbežnému náčrtu našej webovej stránky, na ktorej si môže používateľ pozrieť aktuálny stav robota a jeho predchádzajúce splnené úlohy.

Webová stránka bude mať dva rozhrania, pre používateľa a admina, ktoré budú takmer podobné s výnimkou, že v administrátorskom prostredí bude možnosť spravovať robota.

B.4.1 Spoločné rozhranie (Používateľský režim)

Naše rozhranie bude jednoduchá webová stránka, ktorá bude pozostávať z názvu a popisu projektu, ďalej sa tu bude nachádzať informácia o stave robota (či je zapnutý alebo vypnutý a či sa prechádza alebo práve naviguje). Pod momentálnym stavom robota bude zobrazená história spracovaných úloh. História spracovaných úloh bude spracovaná v tabuľke, ktorá bude vyzeráť približne ako na znázornenej ukážke.

Stavy robota:



Obrázok 6 Grafické znázornenie stavu robota

História spracovaných úloh:

Začiatok navigácie	Koniec navigácie	Štart navigácie	Cieľ navigácie	Celkový čas navigácie
09:50:00	11:40:23	I-11	I-9	15min
12:20:03	12:24:25	I-11	I-24	20 min
12:20:03	12:24:25	I-11	I-24	06 min
13:50:16	15:24:25	I-11	I-24	18 min
21:20:03	21:24:25	I-11	I-24	9 min
22:20:03	23:24:25	I-11	I-24	04 min

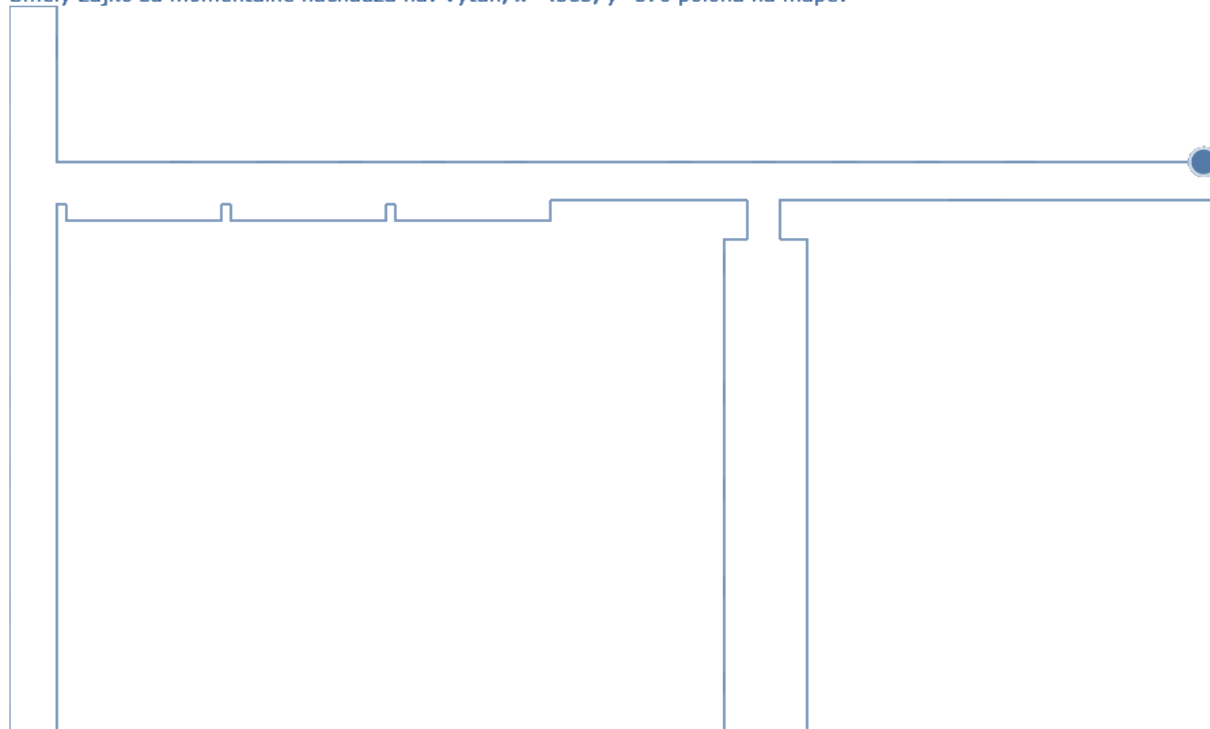
Obrázok 7 Grafické znázornenie tabuľky spracovaných úloh

Aktuálna poloha robota:

Robot bude posilať na web aj svoju aktuálnu polohu. Túto informáciu zašle ako textovú v informáciu v tvare: *Práve sa nachádzam na „miestnosť xy“*.

Okrem toho bude robot zobrazovať svoju polohu aj v grafickom formáte – na mape. Stránka s mapou sa obnovuje automaticky preto nie je potrebné ju aktualizovať pre zistenie najnovšej polohy.

Smelý zajko sa momentálne nachádza na: vytah, x=4363, y=570 poloha na mape:



Obrázok 8 - poloha robota na mape

B.4.2 Administrátorský režim

Na stránke bude okrem hore spomínaných komponentov taktiež tlačidlo „Administrácia prostredia“. Po kliknutí na toto tlačidlo, bude užívateľ vyzvaný zadať heslo, ktoré ho presmeruje na stránku určenú pre administrátorov. Heslo budú poznať len vopred určení správcovia robota. Administrátorský režim stránky vyzerá veľmi podobne ako používateľský, avšak v tomto režime má admin právo vypnúť a zapnúť robota na diaľku a taktiež zrušiť robotovi momentálne zadanú úlohu.

C. Analýza technológií, dekompozícia a dátový model

1. Analýza technológií

Technológie môžeme rozdeliť na 2 skupiny podľa využitia:

- Technológie na správu robota
- Technológie pre webové rozhranie

C.1.1 Technológie na správu robota:

1. Programovací jazyk (C++)

Podstatou celého projektu je programovací jazyk, v ktorom budú naprogramované všetky metódy na riadenie robota. Keďže úlohou robota je aj rozpoznávať postavy rozhodli sme pre jazyk C++, pretože najlepšie spolupracuje s knižnicou OpenCV bližšie opísanou nižšie. C++ je viacparadigmaticý programovací jazyk vyššej úrovne na všeobecné použitie, ktorý umožňuje pracovať aj s prostriedkami nízkej úrovne. Má statickú typovú kontrolu, podporuje procedurálne programovanie, dátovú abstrakciu, objektovo orientované programovanie, ale aj generické programovanie.

2. Knižnica na rozpoznávanie postáv (OpenCV)

Počas jazdy robot sníma prostredie pomocou webkamery. Obrázok z webkamery sa odosiela na rozpoznanie či sa na ňom nachádza nejaká postava. Takúto úlohu dokáže spracovať vybraná knižnica OpenCV. OpenCV je tzv. open-source, multiplatformová knižnica určená pre manipuláciu s obrazom. Je zameraná predovšetkým na počítačové videnie a spracovanie obrazu v reálnom čase.

3. Mapy vo formáte SVG

Nato aby robot mohol správne navigovať po informatickom pavilóne potrebuje poznať polohu miestností a taktiež kde sa práve nachádza. Na tento účel použijeme už vopred pripravené SVG mapy informatického pavilónu, v ktorých sú zaznamenané súradnice potrebných miestností. Scalable Vector Graphics (SVG) je značkový jazyk z rodiny značkových jazykov XML, ktorý je určený na opis dvojrozmernej, statickej alebo animovanej vektorovej grafiky.

4. Hlasový syntetizátor pre komunikáciu s človekom

Keďže robot komunikuje s človekom aj verbálne, potrebujeme na daný problém takzvaný TTS (text-to-speech) program. Robot komunikuje s človekom slovenským jazykom, preto bude potrebné použiť syntetizátor, ktorý podporuje daný jazyk.

C.1.2 Technológie pre webové rozhranie :

1. Hypertext Markup Language (HTML)

HTML je jednoduchý značkovací jazyk bežne používaný na tvorbu statických webstránok zobrazovaných vo webovom prehliadači. Tvorí základ pre rozšírenie stránok o dynamické prvky pomocou jazykov ako napr. PHP. Jeho výhodou je jednoduchosť a bežná podpora zo strany internetových prehliadačov. Veľmi často je používaný spolu s kaskádovými štýlmi (CSS). Jazyk bude použitý pre statický návrh webového prostredia k robotovi, ktorý má informačný aj spravovací charakter.

2. Cascading Style Sheets (CSS)

Kaskádové štýly alebo CSS je všeobecné rozšírenie (X)HTML. Organizácia W3C označuje CSS ako jednoduchý mechanizmus na vizuálne formátovanie internetových dokumentov. Štýly umožňujú oddeliť štruktúru HTML alebo XHTML od vzhľadu. Pre náš projekt bude CSS použité na design webovej stránky.

3. Hypertext Preprocessor (PHP)

PHP je populárny open source skriptovací jazyk, ktorý sa používa najmä na programovanie klient-server aplikácií (na strane servera) a pre vývoj dynamických webových stránok. Pre náš projekt bude PHP využitý na prijímanie údajov o stavoch robota, ich následné spracovanie a taktiež na prihlasovanie do administrátorského prostredia.

4. MySQL

MySQL je slobodný a otvorený viacvláknový, viac užívateľský SQL relačný databázový server. MySQL je podporovaný na viacerých platformách (ako Linux, Windows či Solaris) a je implementovaný vo viacerých programovacích jazykoch ako PHP, C++ či Perl. Databázový systém je relačný typu DBMS (database

management system). Každá databáza je v MySQL tvorená z jednej alebo z viacerých tabuliek, ktoré majú riadky a stĺpce. Práca s MySQL databázou je vykonávaná pomocou takzvaných dotazov, ktoré vychádzajú z programovacieho jazyka SQL (Structured Query Language).

2. Dekompozícia

Projekt bude rozdelený na dve časti:

1. Program pre robota.

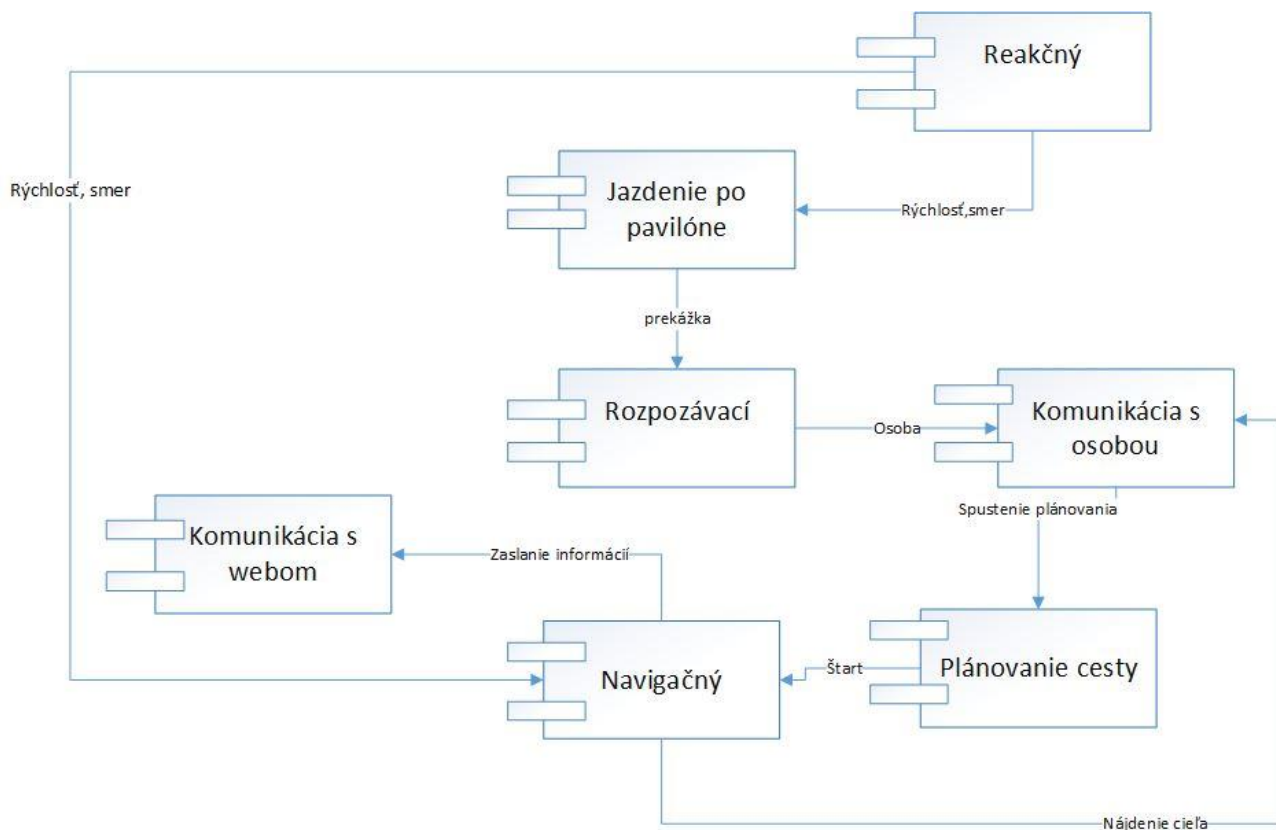
Tento program bude obsahovať triedy potrebné pre jazdu robota po pavilóne, komunikáciu s človekom, komunikáciu so serverom a vyhľadávanie miestností.

- Jazda:
Má metódy na korigovanie smeru a rýchlosti jazdy, výpočet aktuálnej pozície, kontrolu výskytu osoby alebo prekážky. Pamätá si svoju aktuálnu pozíciu a cieľ jazdy.
- Rozpoznávanie prítomnosti človeka:
Pomocou OpenCV sa spracúva výstup z kamery a reaguje na prítomnosť človeka.
- Komunikácia s človekom:
Obsahuje metódy na hlasový výstup robota, spracovanie a kontrolu vstupu a časovač.
- Komunikácia so serverom:
Metódy na odosielanie dát na server a na prijímanie dát zo serveru.

2. Webová stránka.

Na webovej stránke budú bežnému užívateľovi k dispozícii údaje o momentálnom stave robota a história jeho úloh. Admin má možnosť robota vypnúť alebo zapnúť.

C.2.1 Komponentový diagram



Obrázok 9

C.2.2 Popis komponentov

C.2.2.1 Reakčný komponent

Komponent je potrebný pre samotný štart robota, po spustení sa zapnú senzory kamera a spustí sa aj samotný program prechádzania robota. V reakčnom komponente sa ďalej sleduje vzdialenosť robota od stien a zaisťuje sa správny pohyb po chodbe.

C.2.2.2 Komponent Jazdenie po pavilóne

Po spustení robota sa zapína program, ktorý zabezpečuje samotnú jazdu po pavilóne, čo zahŕňa pamätanie pozície robota, ako aj kontrolovanie prekážok pri robotovi.

C.2.2.3 Rozpoznávací komponent

Tento komponent využíva predovšetkým knižnicu OpenCV, pomocou ktorej sa snaží rozpoznať objekt nachádzajúci sa pred robotom. Po zistení, že pred robotom sa nachádza osoba, komponent odošle informáciu a spustí sa komunikačný komponent .

C.2.2.4 Komponent komunikácia s osobou

V tomto komponente sa prevažne využíva hlasový syntetizátor, ktorý oboznámi používateľa o postupe navigácie.

C.2.2.5 Komponent plánovanie cesty

Po zadaní vstupu od užívateľa sa v tomto komponente vypočíta najkratšia cesta k zvolenému miestu a následne sa spustí komponent navigácie.

C.2.2.6 Komponent Navigácia

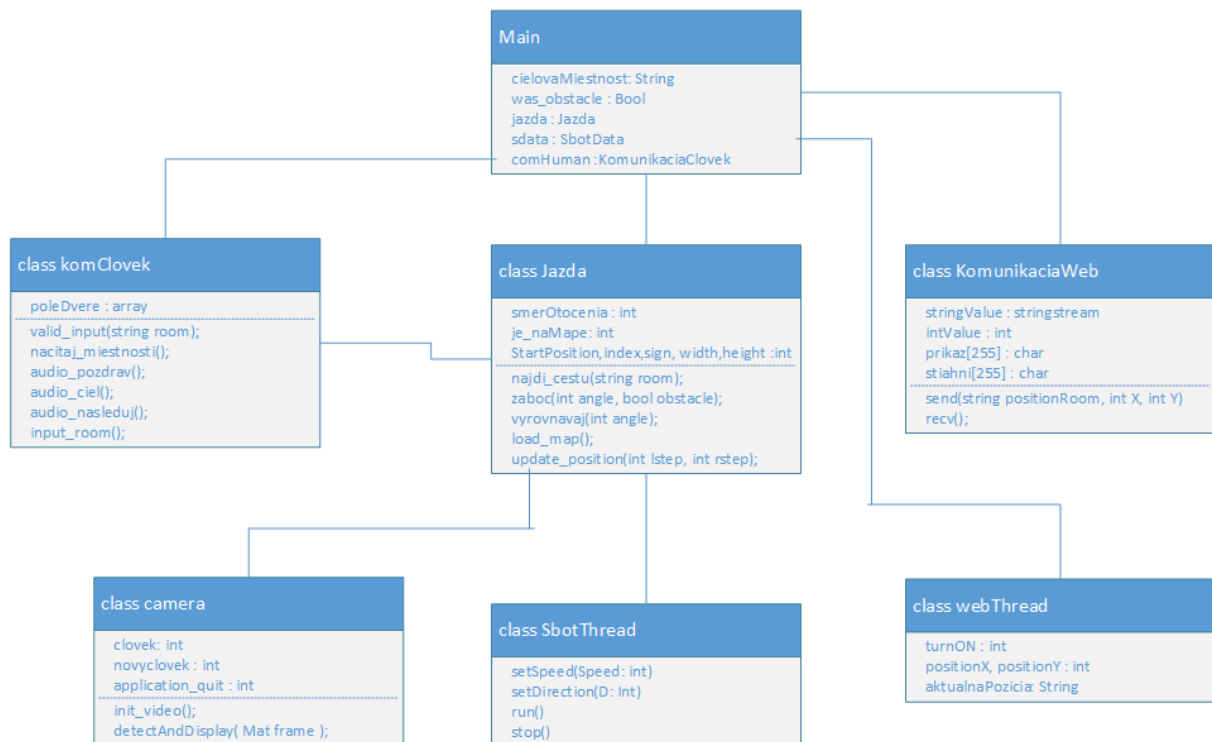
Po úspešnom zadaní vstupu od rozpoznannej osoby sa spustí modul navigácia, ktorý bude postupovať podľa vypočítanej najkratšej cesty.

C.2.2.7 Komponent komunikácia s webom

Tento komponent slúži na komunikáciu s web stránkou na ktorú po ukončení navigácie odošle informácie o navigovaní. Po odoslaní sa znova spúšťa program na jazdenie po pavilóne.

D. Návrh

1. Triedny diagram



Obrázok 2 : class-diagram

2. Popis tried

D.2.1 Main:

Už naprogramovaný modul. Základný program.

D.2.2 Class SbotThread:

Toto je už naprogramovaná trieda na ovládanie robota.

- `setSpeed(speed: Int)` – nastaví rýchlosť pohybu
- `setDirection(d: Int)` – nastaví smer jazdy
- `run()` - uvedie robota do pohybu
- **`stop()`** - zastaví robota

D.2.3 Class Jazda:

Kontrola výskytu prekážok, držania smeru jazdy a aktualizácia pozície. Na ovládanie robota využíva už naprogramovanú triedu `SbotThread`.

- **Najdi_cestu(string room)** – v argumente dostane číslo miestnosti a podľa toho kde sa miestnosť nachádza, určí ktorým smerom sa má robot vybrať
- **Zaboc(int angle, bool obstacle)** – v argumentoch dostane výstup bočných senzorov, a uhol otočenia, následne otočí robota na rohu chodby
- **Vyrovnavaj(int angle)** – vyrovnáva robota do stredu chodby, v argumente dostáva uhol o koľko má robota vyrovnať
- **Load_map()** – načíta mapu objektu
- **Update_position(int lstep, int rstep)** – aktualizuje súradnice robota

D.2.4 Class KomunikaciaClovek:

Stretnutie s človekom.

- **Audio_pozdrav()** – spustí nahrávku pozdravu
- **Audio_ciel()** – spustí nahrávku pre cieľ
- **Audio_nasleduj()** – spustí nahrávku pre nasledovanie
- **Input_room()** – vyžiada zadanie miestnosti
- **Nacitaj_miestnost()** – načíta zadanú miestnosť
- **Valid_input(string room)** – skontroluje správnosť vstupu

D.2.5 Class camera:

- **Init_video()** – načíta video z kamery
- **detectAndDisplay (Mat frame)** – zobrazí načítané video a detekuje v ňom postavy

D.2.6 Class KomunikaciaWeb:

Komunikácia s webom.

- **send(string positionRoom, int X, int Y)** – pošle na web súradnice robota a miestnosť pri ktorej sa robot nachádza
- **recv()** – stiahne potrebné súbory z webu

D.2.7 Class webThread:

Stiahnutie potrebných premenných z webu.

E. Testovacie scenáre

Testovacie scenáre sme si rozdelili do dvoch kategórií, všeobecný testovací scenár a testovacie scenáre pre jednotlivé komponenty:

1. Testovacie scenáre komponentov:

E.1.1 Reakčný komponent

Po manuálnom spustení robota tlačidlom a po štarte programu by sa mal robot začať pohybovať po pavilóne a snažiť sa rozpoznať postavu. Jeho úlohou je jazdiť dokola a snažiť sa držať v strede chodby.

Testovacie prípady:

- **Hardware a batéria je OK:** Nutnou podmienkou aby robot jazdil po pavilóne je nabitá batéria a funkčné ultrazvukové senzory. V opačnom prípade robot nebude schopný korektnej jazdy.
- **Vstup – rýchlosť (speed, int) :** Pri spustení robota je jedným zo vstupných parametrov rýchlosť robota. Rýchlosť je číslo typu integer v rozsahu $\langle 1, 10 \rangle$. V prípade zadania iného čísla alebo nesprávneho typu program vypíše chybu.
- **Vstup – smer (direction, int) :** Ďalším vstupom je smer, ktorým sa robot pohybuje. Smer je defaultne nastavený na priamy pohyb rovno, avšak keď sa robot dostane na koniec uličky musí podľa mapy nájsť správny smer, v ktorom bude pokračovať. V opačnom prípade robot narazí na stenu a zastane.
- **Výstup – senzory (array):** Robot má 5 senzorov, ktoré neustále merajú vzdialenosť od prekážok. Tieto čísla dostane ako výstup typu pole (array). Pomocou bočných senzorov sa robot snaží udržať v strede. Pokiaľ sa hodnota pravého alebo ľavého senzoru konštantne zmenšuje znamená to že robot ide do jednej strany preto sa bude snažiť vyrovnať do opačnej strany. V prípade, že hodnota, niektorého zo senzorov bude menšia ako stanovený limit, znamená to že robot je blízko prekážky. V tomto prípade robot zastane a čaká kým prekážka nezmizne.

E.1.2 Jazdenie po pavilóne

Z reakčného komponentu na vstupe dostáva rýchlosť a smer podľa pozície kde sa robot nachádza. Ak bude smer a rýchlosť správne nastavená nemala by nastať kolízia robota s okolím alebo s prekážkou.

Testovacie prípady:

- Vstup: dobre zvolená rýchlosť a smer Výstup: jazdenie v strede pavilónu bezproblémová detekcia prekážok
- Vstup: príliš veľká rýchlosť , vhodný smer Výstup: možné kolízie, nepresné detekcie ultrazvukových senzorov
- Vstup: vhodná rýchlosť, nesprávny smer : takmer istá nefunkčnosť robota, vysoké riziko kolízie
- Vstup: nevhodná rýchlosť, nevhodný smer Výstup: riziko ujmy na zdravý, možné poškodenie robota či okolitého prostredia

E.1.3 Rozpoznávací komponent

Tento komponent využíva predovšetkým knižnicu OpenCV, pomocou ktorej sa snaží rozpoznať objekt nachádzajúci sa pred robotom. Pomocou ultrazvukových senzorov robot rozporná či sa pred ním nachádza prekážka. Pokiaľ áno, spustí sa kamera pomocou ktorej zistí či pred ním stojí osoba alebo nie. Po zistení, že pred robotom sa nachádza osoba, komponent odošle informáciu a spustí sa komunikačný komponent .

Testovacie prípady:

Predný ultrazvukový senzor:

- **Cesta voľná** – kamera sa nezapne a robot pokračuje ďalej v jazde a spracovávaní výstupov zo senzoru.
- **Prekážka** – Podľa vzdialenosti od prekážky robot najprv zistí či je potrebné zastať alebo je ešte v dostatočnej vzdialenosti aby nenabúral do prekážky. V momente ako zaznamená prekážku sa zapne kamera aby spracovala obraz pred sebou. Pomocou knižnice OpenCV sa spracuje výstup z kamery a vráti informáciu, či pred robotom stojí osoba alebo čokoľvek iné. Pokiaľ je to osoba tak rozpoznávací komponent pošle informáciu a spustí sa komunikačný komponent. Pokiaľ to nie je osoba robot čaká na odstránenie prekážky.

Predno-bočné ultrazvukové senzory:

- **Cesta voľná** - kamera sa nezapne a robot pokračuje ďalej v jazde a spracovávaní výstupov zo senzorov.
- **Prekážka** - Podľa vzdialenosti od prekážky robot najprv zistí či je potrebné zastať alebo je ešte v dostatočnej vzdialenosti aby nenabúral do prekážky. V momente ako zaznamená prekážku sa zapne kamera aby spracovala obraz pred sebou. Pomocou knižnice OpenCV sa spracuje výstup z kamery a vráti informáciu, či pred robotom stojí osoba alebo čokoľvek iné. Vzhľadom na to, že objekt v tomto prípade nemusí byť celý

nasnímaný kamerou a teda, že knižnica OpenCV ho nedokáže spracovať. V takomto prípade sa vyhodnotí objekt pred robotom za prekážku a teda robot bude čakať na odstránenie. Pokiaľ OpenCV rozpozná postavu tak rozpoznávací komponent pošle informáciu a spustí sa komunikačný komponent.

Bočné ultrazvukové senzory:

- **Cesta voľná** – v tomto prípade nastane len pokiaľ sa robot nachádza na križovatke chodieb. Kamera sa nezapne a robot pokračuje v jazde smerom, ktorým sa sám rozhodne na základe mapy.
- **Prekážka** – v tomto prípade sa kamera zapínať nebude. Pretože bočné senzory slúžia na udržiavanie robota v strede chodby. Pokiaľ nastane situácia, že osoba prejde popri robotovi zprava alebo zľava tak robot v tomto prípade nereaguje a považuje človeka za stenu.

E.1.4 Komunikácia so serverom

Tento komponent slúži na komunikáciu s web stránkou na ktorú po ukončení navigácie odošle informácie o navigovaní. Po odoslaní sa znova spúšťa program na jazdenie po pavilóne.

Testovacie prípady:

- **Vstup : send_stav(stav: string):** Počas celej navigácie robot posiela na server informácie o svojej pozícii, ktoré sa následne zobrazujú robotu na mapke. V prípade zadania nesprávneho stringu alebo iného typu server nebude schopný spracovať údaje a vypíše chybové hlásenie, poprípade vykreslí neaktuálnu mapu.
- **Vstup : send_quest(quest: string):** Po skončení navigácie sa na server odošlú informácie o úlohe. V prípade odoslania nesprávneho typu informácií server vypíše chybové hlásenie a nebude možné skontrolovať splnené úlohy robota.
- **Vstup: recv():** Cez server je možné aj zapnutie a vypnutie robota. V prípade, že by nastali nejaké problémy so serverom nebude nijako ohrozená samotná funkčnosť robota. Pri odoslaní nekorektných dát na zapnutie či vypnutie robota sa robot bude ovládať priamo na mieste vykonávania činnosti.

E.1.5 Navigačný komponent

Po stretnutí s človekom a po zadaní korektného vstupu, by robot mal vyhľadať najkratšiu cestu k požadovanému cieľu. Následne by sa mal začať pohybovať podľa vypočítanej trasy až kým nedorazí do cieľa.

Testovacie prípady:

Korektný vstup – aby sa dala nájsť najkratšia cesta, je nutné aby robot dostal správny vstup, teda číslo existujúcej miestnosti v správnom formáte.

- **najdi_cestu(ciel: Int)**
 - **vstup** – číslo zodpovedajúce niektorej miestnosti v pavilóne
 - **výstup** – najkratšia cesta od aktuálnej polohy k zadanej miestnosti vo formáte poľa čísiel vrcholov (pravdepodobne)
 - **vstup** – hocijaký nečíselný vstup, alebo číslo, ktoré nezodpovedá žiadnej miestnosti v pavilóne
 - **výstup** – program spadne na chybe, avšak zlý vstup by sa sem dostať nemal
- **update_position()**
 - **vstup** – hodnota nameraná otáčkomerom: podľa hodnoty, ktorú ako výstup dá otáčkomer sa aktualizuje pozícia. Otáčkomer by mal dávať korektný výstup, v opačnom prípade program spadne na chybe
- **naviguj (cieľ (int,int))**
 - **vstup** – trasa k cieľovej miestnosti ako pole ID-čiek miestností
 - **výstup** – funkcia robota ovláda a naviguje podľa vstupnej trasy
nekorektný vstup – hocičo iné ako pole ID-čiek miestností, alebo pole obsahujúce nekorektné ID, potom na takomto vstupe program spadne na chybe

2. Záver z testovania

Po záverečnom testovaní nášho projektu môžeme konštatovať, že projekt funguje aj keď nie je úplne vyladený. Keďže sa jednalo o projekt kde sme museli pracovať s hardvérovými časťami (ultrazvukové senzory, webkamera), ktoré nie sú úplne spoľahlivé, v projekte nastali chyby a odchýlky, ktoré sme neboli schopní odstrániť / vyladiť aby projekt fungoval bezchybne. V nasledujúcej časti sú popísané ako reálne fungujú jednotlivé komponenty, aké v nich vznikali problémy a čo ich pravdepodobne mohlo spôsobovať.

- **Reakčný komponent** – v tomto komponente funguje všetko ako má. Po štarte programu sa robot automaticky zapne do stavu prechádzanie a rozpoznáva postavy. Jediný problém, ktorý nastal bol, že pri neopatrnom prenášaní robota sa narušili kontakty so senzormi a základnou doskou čo zapríčinilo nefunkčnosť robota. Po opätovnom napravení sa však problém vyriešil.
- **Komponent jazda po pavilóne** – tento komponent funguje z väčšej miery v poriadku, avšak nastávajú tu isté problémy. Hlavným problémom je zabáčanie na chodbách pavilónu, kde robot prijme príliš skoro alebo neskoro hodnoty zo senzorov, pri ktorých má zabojiť a tým pádom sa nestihne včas korektne otočiť a narovnať, v dôsledku toho vznikne kolízia a robota je nutné manuálne napraviť. Vyrovnávanie v strede chodby funguje takmer dokonale, menšie problémy vznikajú pri výklenkoch na chodbách alebo neočakávaných predmetoch umiestnených pri stenách. Vo väčšine prípadoch však robot napriek týmto problémom dokázal cúvnuť a vyrovnať sa. Rozpoznanie prekážok funguje, ktoré sa vyskytnú pred robotom funguje bezchybne, pri rozpoznaní takejto prekážky robot zastane a čaká na uvoľnenie trasy, potom pokračuje v ceste.
- **Rozoznávací komponent** – rozpoznávanie postáv taktiež nefunguje úplne bezchybne. Pri testovaní sme zistili, že v dôsledku mnohých úloh, ktoré musí robot vykonávať je obraz z webkamery oneskorený preto je potrebné stáť pred kamerou dlhší čas kým robot rozpozná osobu. Ďalším problémom je samotné rozpoznávanie, ktoré ma na starosti knižnica openCV. Knižnica dokáže rozpoznať tvár avšak nie je úplne spoľahlivá a, v niektorých prípadoch rozpoznala tvár aj tam kde sa nenachádzala. Problém mohol vzniknúť kvôli nedostatočnému osvetleniu, kvality snímok alebo samotnými algoritmami knižnice na rozpoznávanie. Tento problém sme sa snažili eliminovať tak, že za postavu sme považovali len objekt, ktorý sa pri rozpoznávaní objavil na kamere viac ako raz. Táto metóda znížili frekvenciu chybných rozpoznaní avšak neodstránila problém úplne.
- **Komponent komunikácia s osobou** – funguje bezchybne. Po rozpoznaní postavy sa spustila pripravená nahrávka, robot čakal na vstup z klávesnice. V prípade správneho vstupu začal robot navigovať, v prípade nesprávneho vstupu bol užívateľ opätovne vyzvaný zadať konkrétny vstup.

- **Komponent plánovanie trasy** – funguje bezchybne. Robot dokáže vypočítať najkratšiu trasu v danej miestnosti. V prípade potreby sa dokáže otočiť o 180 stupňov, v opačnom prípade pokračuje rovno.
- **Komponent navigácia** – funguje bezchybne. V prípade, že robot nenarazil na žiadnu prekážku zvládol úspešne dôjsť do cieľa. Následne prebehol oznam o úspešnej navigácii a robot sa prepol do stavu prechádzania. Zadaná miestnosť sa zhodovala s aktuálnou miestnosťou.
- **Komponent komunikácia s webom** – všetky naprogramované časti fungujú bezchybne. Robot posielal informácie o aktuálnej pozícii na webovú stránku v textovej aj grafickej forme na mape. Stránka s polohou sa automaticky obnovuje každú sekundu. Po prihlásení na stránku je možné robota zapnúť/ vypnúť na diaľku – toto taktiež funguje bez problémov. Zaznamenávať históriu navigácií sa nám žiaľ kvôli nedostatku času nepodarilo stihnúť dokončiť.

F. Zhodnotenie diela a tímovej práce

1. Všeobecné zhodnotenie

F.1.1 Všeobecný pohľad

Na projekte smelý zajko sprievodcom sme ako skupina intenzívne pracovali počas celého zimného semestra, školského roku 2014/15. Na začiatku roka sme zozbierali potrebné informácie od nášho zadávateľa dr. Petroviča, ktoré sme spísali do katalógu požiadaviek, ktorý následne zadávateľ schválil. Potom nastala fáza analyzovania. Keďže ešte žiadny z členov tímu nemal dostatočné skúsenosti v oblasti robotiky a programovania robotov bola táto téma pre nás nová a museli sme venovať veľa času premýšľaním ako bude náš projekt reálne vyzeráť. Po naštudovaní problematiky sme napísali konceptuálnu analýzu, kde nás usmerňoval a radil náš cvičiaci Mgr. Škoviera. Po úspešnom napísaní analýzy nás čakala najťažšia časť – samotný návrh a programovanie. Keďže bola pre nás daná problematika úplne nová zo začiatku sme postupovali veľmi pomaly, no po malých krokoch sme sa všetci postupne oboznámili s problematikou robotiky a postupne dokončili náš projekt. Keďže bol pre nás projekt náročný k jeho úspešnému dokončeniu a odovzdaniu došlo až na konci skúškového obdobia – začiatkom februára. Napriek náročnosti projektu hodnotíme celkovo projekt pozitívne, naučili sme veľa nielen z oblasti robotiky ale aj ako správne postupovať pri tvorbe informačných systémov. Poďakovať by sme sa chceli nášmu cvičiacemu Mgr. Škovierovi, ktorý nás usmerňoval a radil ako správne postupovať pri písaní potrebných dokumentov a taktiež nášmu zadávateľovi dr. Petrovičovi, ktorý nám pomáhal riešiť hardvérové aj softvérové problémy, ktoré vznikali počas tvorby a programovania projektu.

F.1.2 Komplikácie počas vývoja

Vývoj projektu môžeme všeobecne zhodnotiť ako bezproblémový aj napriek niektorým komplikáciám, ktoré počas neho nastali. Za najväčší problém považujeme fakt, že k práci sme neustále museli mať k dispozícii robota, ktorý je umiestnený na fakulte matematiky, fyziky a informatiky. Z tohto dôvodu sme sa museli stretávať väčšinu času len v škole, čo bolo hlavne časovo veľmi náročné. Možnosť rozdeliť si úlohy aby každý mohol robiť niečo súčasne v rovnakom čase bola tiež ťažko realizovateľná, keďže k dispozícii sme mali len jeden notebook a robota na testovanie. Z tohto dôvodu vznikali pri projekte veľké časové medzery a odovzdávanie projektu sa natiahlo až na február. Ďalším problémom boli nedostatočné znalosti celého tímu ohľadom programovacieho jazyka C++ a zároveň práca pod operačným systémom Linux. Študovanie týchto technológií zabralo taktiež mnoho času z našej práce, avšak môžeme konštatovať, že všetci členovia tímu sa vďaka projektu v tejto problematike výrazne zlepšili čo hodnotíme ako výrazné plus. Poslednou komplikáciu pri našom projekte je samotný robot, s ktorým sme často mali

nejaké hardvérové problémy, ktoré sme nevedeli samostatne vyriešiť, ako napríklad nereagujúce senzory, odpojené konektory alebo neschopnosť zapojiť komponenty robota do správnych portov.

F.1.3 Rozdelenie práce a komunikácia s tímom

Prácu s tímom hodnotíme všetci pozitívne a nemali sme žiadny výrazný problém. Počas tvorby katalógu požiadaviek a písaní konceptuálnej analýzy sme sa ako tím minimálne raz do týždňa stretli s našim cvičiacim kde sme prediskutovali našu činnosť z minulého týždňa a dohodli sa a naplánovali úlohy na ďalší týždeň. V rámci týždňa sme mali ešte jedno tímové stretnutie, kde sme prediskutovali aktuálne problémy ohľadom projektu a spoločne sme sa snažili napísať potrebné veci, ktoré sme naplánovali s cvičiacim. Úlohy, ktoré sme nestihli urobiť spoločne sme si rozdelili v rovnakom pomere v rámci tímu a stanovili termín dokončenia. Môžeme konštatovať, že v rámci normy si každý svoje povinnosti plnil dôsledne a načas.

Okrem pravidelných tímových stretnutí sme často komunikovali aj on-line, prostredníctvom sociálnej siete Facebook, kde sme mali vytvorenú spoločnú skupinu a sem sme nahrávali a konzultovali naše úlohy. Výsledné dokumenty sme nahrávali do systému github, ku ktorému mal prístup aj náš cvičiaci aby nám mohol dať spätnú väzbu k našej aktivite.

Pri tvorbe návrhu a samotnom programovaní robota sme sa museli pravidelne stretávať v škole na našej fakulte. Prácu sme si rozdelili nasledovne:

Dominik mal na starosti postarať sa o komunikáciu robota s webom a rozpoznávanie postáv v openCV.

Maťo mal za úlohu jazdenie a chod robota, t.j. vyrovňovanie na chodbe zatáčanie, zisťovanie svojej polohy atď.

Miro mal za úlohu plánovanie trasy robota a prácu s svg mapou, t.j. naparsovať svg mapu do programu tak aby ju robot vedel správne čítať a využiť a aby vedel vypočítať najkratšiu trasu k miestnosti kam má navigovať.

Natália mala za úlohu komunikáciu s robotom t.j. naučiť robota rozprávať, spraviť aby komunikoval s človekom keď ho rozpozná alebo zavedie do cieľa.

Napriek tomu, že sme mali takto rozdelené úlohy, konštatujeme že toto rozdelenie ostalo skôr len na teoretickej úrovni a podstatnú časť programu sme vymysleli spoločnými nápadmi a brainstormingom. Dôvodom je zrejme to, že sme sa všetci ako tím museli stretávať, keďže sme potrebovali k práci robota a tým zanikala možnosť samostatnej práce na pridelených častiach. Okrem pridelenej programovacej časti mal každý za úlohu vytvoriť aj niektorú časť požadovaných dokumentov a diagramov.

Vo všeobecnosti však môžeme konštatovať že tímová práca a komunikácia prebiehala bez problémov, každý si plnil svoje povinnosti a všetci sme spokojní s výsledkom.

F.1.4 Návrhy pre budúce verzie

Samotný projekt smelý zajko je len v prvej fáze vývoja a v budúcnosti by mohol disponovať mnohými ďalšími funkciami a vylepšeniami. V budúcej verzii bude hlavne

potrebné doladiť všetky detaily, ktoré sa nestihli dokončiť v tej súčasnej. Ďalej by sa dala zlepšiť samostatná komunikácia s webom. Robot by napríklad mohol byť celkovo ovládaný na diaľku cez web, teda že by napríklad streamoval záznam z webkamery priamo na stránku a užívateľ by vedel na diaľku zadať koho má robot osloviť alebo kam ísť. Zlepšiť by sa mohol aj rozsah navigácie robota aby nenavigoval len v rámci pavilónu ale v celej budove univerzity. Poloha robota sa mohla určovať presnejšie napríklad využitím intenzity Wifi signálov v pavilóne.

2. Zhodnotenie členmi tímu

F.2.1 Natália Ďurisová

Naša tímová práca a komunikácia bola bezproblémová a výborná. Všetci členovia tímu sme sa poznali veľmi dobre už pred začatím projektu, takže nebol problém v komunikácii a pri stretnutiach. Téma projektu nás moc nepotešila, nakoľko sme ani jeden z nás nemali skúsenosti v tejto oblasti. Nakoniec sme sa však s tým zmierili a pustili sa do toho. Práca bola náročná na čas a nebolo veľa možností na samostatnú prácu alebo prácu mimo školy. Každý z tímu splnil svoju časť práce aj keď sme to nemali úplne presne rozdelené. Všetci sme obetovali náš voľný čas aby sme sa vedeli stretávať spoločne a pracovať, niekedy až dlho po večierke. Nakoniec sme projekt dokončili úspešne aj keď sme si mysleli, že sa nám to nikdy nepodarí. Veľmi nás potešilo keď nám projekt schválil zadávateľ. Takže keď to zhrniem spolupráca bola výborná a každý urobil všetko pre to aby sme projekt doviedli do finálnej verzie.

F.2.2 Miroslav Gregorec

Projekt hodnotím ako veľmi pekný nápad, avšak myslím že časovou náročnosťou dosť predbehol ostatné projekty z TIS. Celkovú prácu tímu hodnotím veľmi pozitívne, osobne ma prekvapil najmä prístup Kamila, ktorý venoval najviac času programovaniu a vďaka ktorému sme sa dosť dostali do tempa. Samozrejme aj ostatní členovia pracovali ako vedeli čo ma veľmi teší. Myslím, že sme sa naučili pri projekte veľa nových vecí aj keď sme pri ňom strávili veľa času. Som však rád že stretnutia neboli vždy len o programovaní a trápení sa s projektom ale zažili sme na nich veľa zábavy a tým sme združili tímového ducha. Celkové hodnotenie projektu je pre mňa teda veľmi kladné.

F.2.3 Dominik Knechta

Moje dojmy z tímovej práce a komunikácie by som zhrnul ako veľmi dobré. Keďže sme sa všetci členovia tímu poznali už vopred, trochu som sa obával, že to môže pokaziť tímovú spoluprácu a že budú veci odkladať na poslednú chvíľu. Nakoniec však boli moje obavy zbytočné a tímová práca prekonala moje očakávania. Hoci sme mali problémy s časom, nakoniec sme projekt stihli dokončiť a odovzdať včas. Všetci členovia tímu si svoje úlohy plnili dôkladne. Komunikácia s tímom prebiehala bez problémov, všetci pravidelne chodili na stretnutia a aktívne sa zapájali do diskusií o problémoch ohľadne projektu a navrhovali svoje riešenia. Myslím, že každý si svoju stanovenú časť projektu splnil a celkovo sme pracovali ako dobrý tím a vytvorili akceptovateľný projekt.

F.2.4 Martin Mudroch

V týme musím pochváliť dobrú komunikáciu. Stretnutia, z ktorých vznikali aj zápisnice sa konali minimálne raz do týždňa. Okrem týchto stretnutí komunikácia prebiehala aj na internete. Termíny úloh sa plnili relatívne v termíne. Aj keď nebolo ľahké sa vždy stretnúť na testovania s robotom, každý pre projekt spravil maximum čo mohol. Celkovo práca v týme bola zaujímavá a získali sme mnoho skúseností. Čo sa výsledku projektu ako takého týka som vcelku spokojný, keďže som nečakal že robot bude lietať alebo zabíjať ľudí.

3. Zhodnotenie zadávateľom

akceptacia projektu



Doručená pošta x



Pavel Petrovic

komu: Natália, Miroslav, mne, Martin ▾

Dobry den,

v utorok 10.2.2015 mi bola predvedena uspesna navigacia robota v pavilone informatiky, robot doviedol zameraneho navstevnika k pozadovanej miestnosti, gratulujem. Tymto mozete povazovat implementacnu cast za hotovu, hoci stale nefunguje dokonale. Najvacsi nedostatok vidim v nespravnom zatacani v zakrutach na konci rovných chodieb. V podobe, v akej je teraz, navrhujem 85% bodov (=42.5). Ak by ste chceli dorobiť aj otacanie v zakrutach, tak sa hodnotenie moze prípadne zlepšiť. Pre poriadok mozete tento mail preposlat svojmu cviciacemu.

S pozdravom,
Pavel.