

Návrh softvéru
Smelý zajko sprievodcom
Tvorba informačných systémov

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

Obsah

1	Úvod	3
2	Konceptuálna analýza	4
2.1	Analýza používateľov	4
2.2	Diagramy	4
2.2.1	Entitno-relačný diagram	4
2.2.2	Use-case diagram.....	5
2.2.2.1	Diagram pre používateľov webového rozhrania.....	5
2.2.2.2	Diagram pre používateľov robota.....	7
2.2.3	Stavový diagram	8
2.3	Používateľské rozhranie	9
2.3.1	Spoločné rozhranie (Používateľský režim)	9
2.3.2	Administrátorský režim.....	10
3	Analýza technológií, dekompozícia a dátový model	11
3.1	Analýza technológií	11
3.1.1	Technológie na správu robota:	11
3.1.2	Technológie pre webové rozhranie :	12
3.2	Dekompozícia	13
3.2.1	Komponentový diagram.....	14
3.2.2	Popis komponentov.....	14
3.2.2.1	Komponent Zapnutie robota cez web.....	14
3.2.2.2	Komponent Jazdenie po pavilóne.....	14
3.2.2.3	Komponent Rozpoznanie Osoby.....	15
3.2.2.4	Komponent Navigácia.....	15
3.2.2.5	Komponent Zaslание reportu	15
4	Návrh.....	16
4.1.1	Triedny diagram	16
4.1.2	Popis tried	16

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 (roboťa) 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 roboťa, popísané aj technológie použité pre webové prostredie a vzájomnú komunikáciu medzi robotom a webom.
- Návrh

2 Konceptuálna analýza

2.1 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.

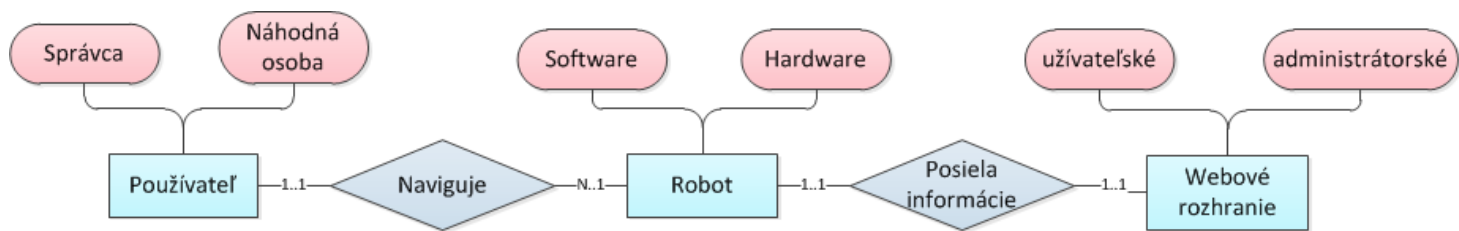
2) Správca

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, časť use-case diagram.

2.2 Diagramy

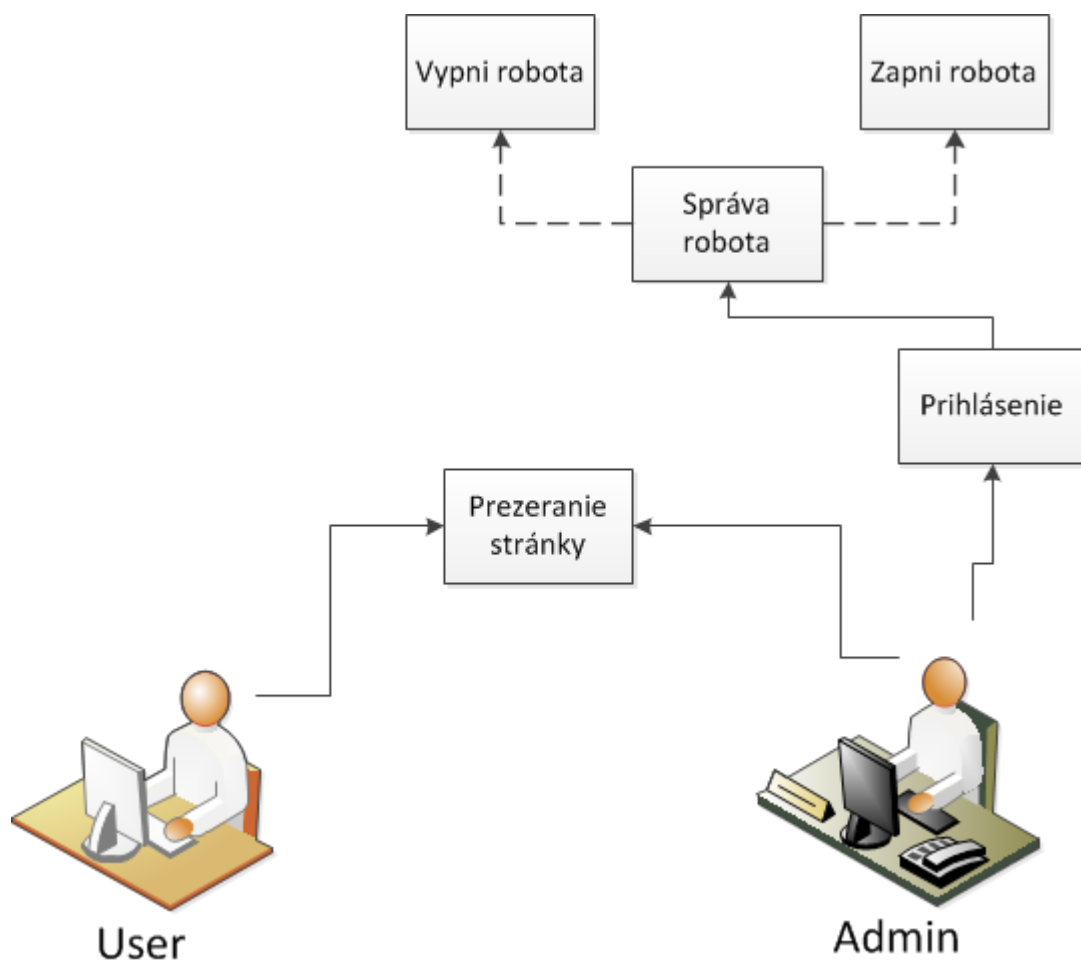
2.2.1 Entitno-relačný diagram



Obrázok 1 Entitno-relačný diagram

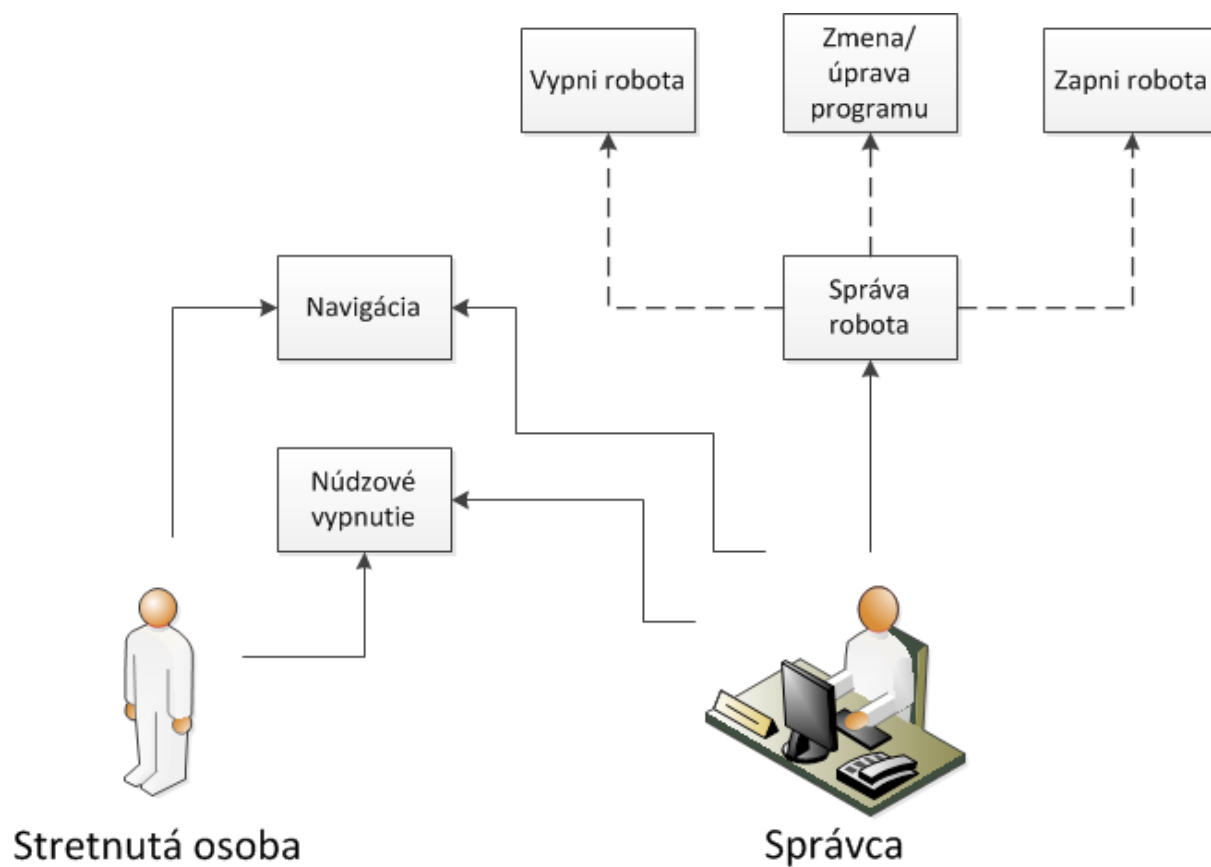
2.2.2 Use-case diagram

2.2.2.1 Diagram pre používateľov webového rozhrania



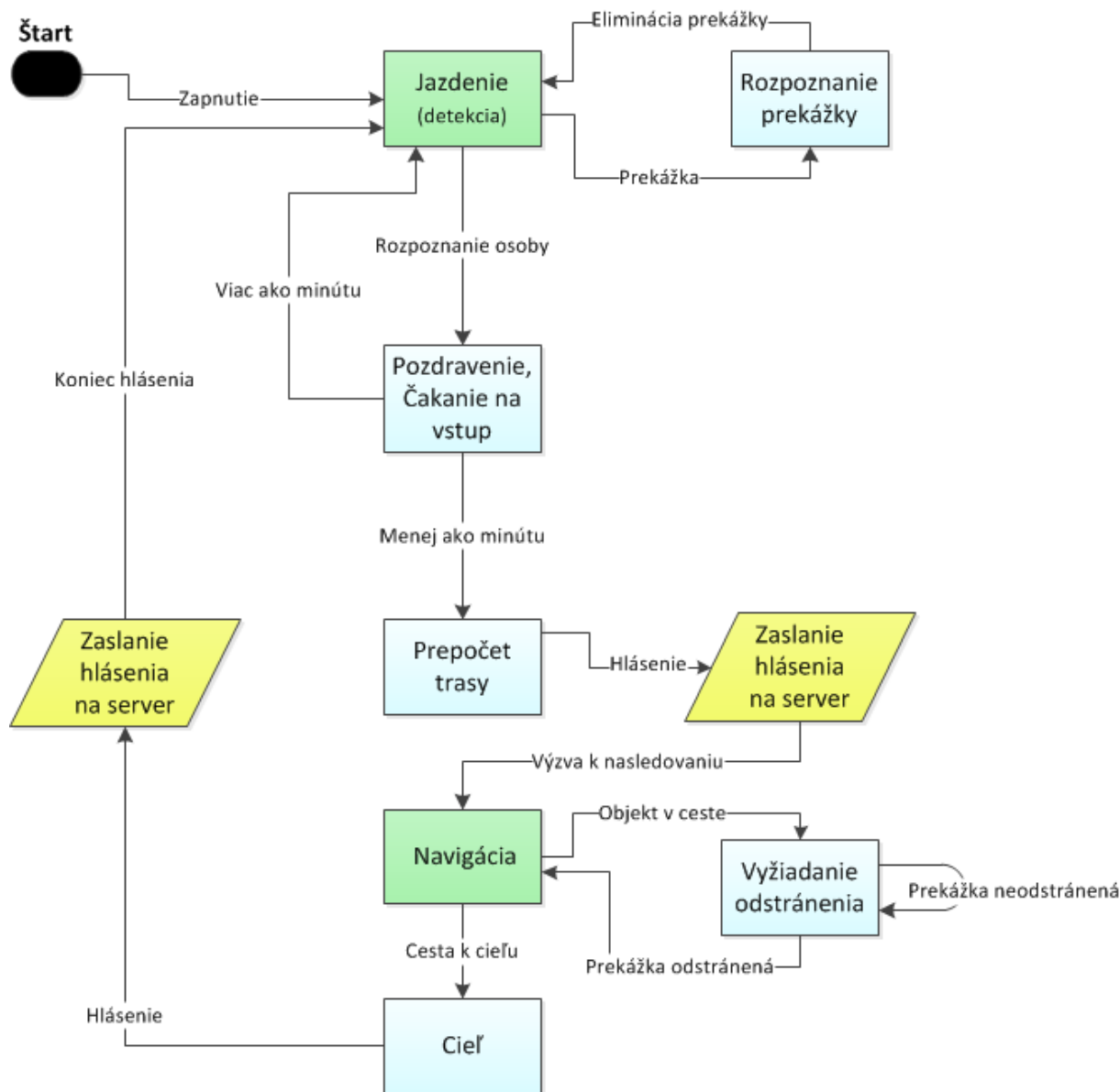
Obrázok 2 Use Case diagram pre webové rozhranie

2.2.2.2 Diagram pre používateľov robota



Obrázok 3 Use Case diagram pre robota

2.2.3 Stavový diagram



Obrázok 4 Stavový diagram robota

2.3 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.

2.3.1 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 5 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 6 Grafické znázornenie tabuľky spracovaných úloh

2.3.2 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.

3 Analýza technológií, dekompozícia a dátový model

3.1 Analýza technológií

Technológie môžeme rozdeliť na 2 skupiny podľa využitia:

- Technológie na správu robota
- Technológie pre webové rozhranie

3.1.1 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 jazdenia 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.

3.1.2 Technológie pre webové rozhranie :

1. Hypertext Markup Language (HTML)

HTML je jednoduchý značkovací jazyk bežne používaný na tvorbu statických webových stránok 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.

3.2 Dekompozícia

Projekt bude rozdelený na dve časti:

1. Program pre robota.

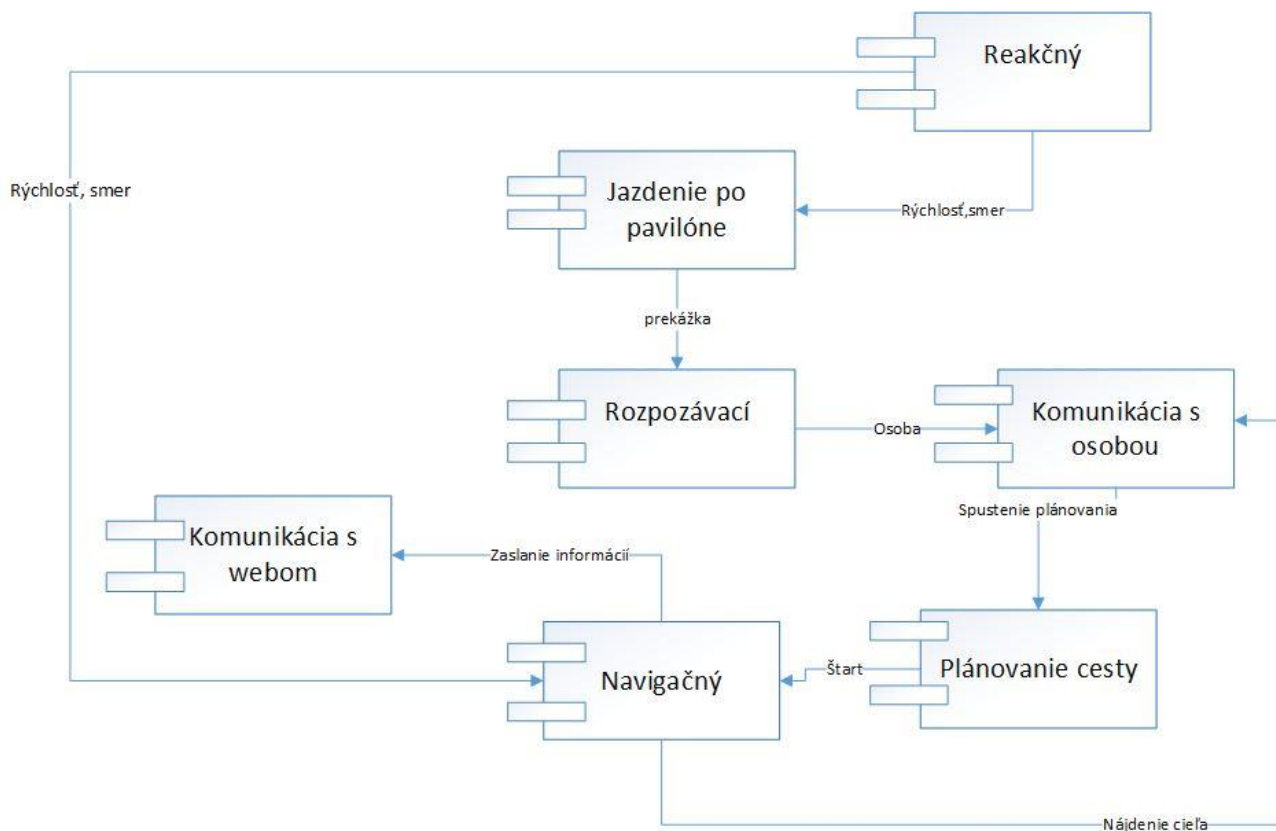
Tento program bude obsahovať triedy potrebné pre jazdu robota po pavilóne, komunikáciu s človekom, komunikáciu so serverom a vyhľadávanie miestností.

- 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.

3.2.1 Komponentový diagram



Obrázok 7

3.2.2 Popis komponentov

3.2.2.1 Komponent Zapnutie robota cez web

Komponent je potrebný pre samotný štart robota, po spustení sa zapnú senzory kamera a spustí sa aj samotný program prechádzania robota. Robot už však musí byť umiestnený v správnom pavilóne so zapnutými motormi.

3.2.2.2 Komponent Jazdenie po pavilóne

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

3.2.2.3 Komponent Rozpoznanie Osoby

V 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 použije hlasový syntetizátor a osobu osloví.

3.2.2.4 Komponent Navigácia

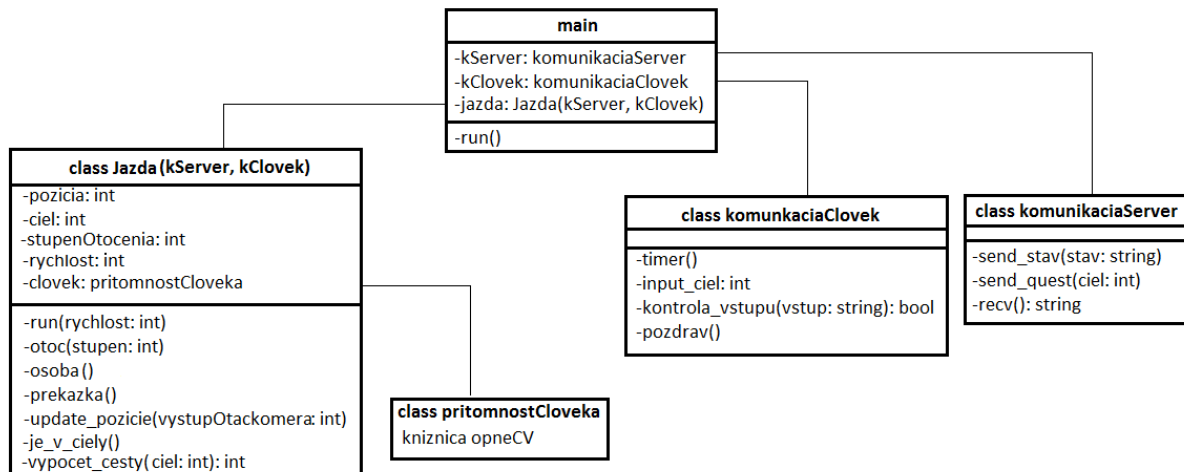
Po úspešnom zadaní vstupu od rozpoznanej osoby sa spustí modul navigácia ktorý najskôr vypočíta najkratšiu cestu k hľadanej miestnosti, vyzve osobu aby ho nasledovala a odvedie ju k hľadanému miestu.

3.2.2.5 Komponent Zaslanie reportu

Tento komponent sa používa 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.

4 Návrh

4.1.1 Triedny diagram



Obrázok 2 : class-diagram

4.1.2 Popis tried

- **Trieda Jazda**

Obsahuje metódy a premenné potrebné k ovládaniu jazdy robota a počítanie jeho polohy.

pozicia – aktuálna pozícia

ciel – aktuálny cieľ, ak sa len „prechádza“ hodnota je None

stupenOtocenia – aktuálny stupeň otočenia, na začiatku je 0

rychlost – aktuálna rýchlosť

clovek – využíva triedu pritomnostCloveka na rozpoznávanie prítomnosti osoby

run(rychlost: int) – uvedie robota do pohybu pod aktuálnym uhlom natočenia rýchlosťou ktorú dostane v argumente

otoc(stupen: int) – otočí robota o daný stupeň, tento stupeň si pamätá v premenej stupenOtocenia

osoba() – kontroluje, či sa pred robotom nachádza človek

prekazka() – udržiava robota v strede chodby a kontroluje výskyt prekážok, koriguje jazdu pomocou otoc

update_pozicie() – podľa výstupu z otáčkomera a zapamätaných hodnôt uhlu a rýchlosti zapisuje do premenej pozicia aktuálnu pozíciu

je_v_cieli() – kontroluje, či je robot v cieľi

vypocet_cesty(ciel: int):int – aktualizuje cieľ a podľa potreby robota otočí o 180° alebo o 0°

- **Trieda komunikaciaClovek**

Obsahuje metódy na komunikáciu s človekom, ako aj na zadanie vstupu.

pozdrav() – spustí sa hlasová nahrávka s pozdravom

timer() – ak do 30 sekúnd robot nedostane vstup, pokračuje v jazde

input_ciel() – čaká na zadanie cieľovej miestnosti

kontrola_vstupu(vstup) – kontroluje korektnosť zadaného cieľa

- **Trieda komunikaciaServer**

Trieda slúži na komunikáciu so serverom.

send_stav(stav: string) – odošle na server informáciu o aktuálnom stave

send_quest(quest: string) – odošle na server informáciu o aktuálnej úlohe

recv() – prijíma inštrukcie zo serveru

- **Main**

Metódou ***run()*** sa bude spracúvať výstup robota (údaje zo senzorov, otáčkometra...).