



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA DOPRAVNÍ

Bc. Martin Štourač

Navigační systémy pro vnitřní prostory

Diplomová práce

2021

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta dopravní

děkan

Konviktská 20, 110 00 Praha 1



K617..... Ústav logistiky a managementu dopravy

ZADÁNÍ DIPLOMOVÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení studenta (včetně titulů):

Bc. Martin Štourač

Studijní program (obor/specializace) studenta:

navazující magisterský – LA – Logistika a řízení dopravních procesů

Název tématu (česky): **Navigační systémy pro vnitřní prostory**

Název tématu (anglicky): Indoor Navigation Systems

Zásady pro vypracování

Při zpracování diplomové práce se řiďte následujícími pokyny:

- přehled navigačních systémů a jejich využití v logistice
- teoretický základ navigačních technologií, využití teorie grafů, navigační algoritmy a podmínky jejich fungování
- SW nástroje využívané pro tvorbu vlastního indoor navigačního systému (databázové nástroje, open source webové technologie a software)
- vývoj a implementace indoor navigačního systému pro konkrétní veřejně přístupný objekt (stanovení a popis objektu, postup prací při tvorbě systému, popis implementace systému na web)
- vyhodnocení výstupů a závěrečná doporučení





- Rozsah grafických prací: stanoví vedoucí diplomové práce
- Rozsah průvodní zprávy: minimálně 55 stran textu (včetně obrázků, grafů a tabulek, které jsou součástí průvodní zprávy)
- Seznam odborné literatury: Mocková D.: Základy teorie dopravy, ČVUT v Praze 2007
Obe R. and Hsu L.: pgRouting - A Practical Guide, LocatePress 2017
Haverbeke M.: JavaScript - Richtig gut programmieren lernen, Dpunkt.Verlag 2019

Vedoucí diplomové práce: **Ing. Alexandra Dvořáčková, Ph.D.**
Ing. Libor Židek


Datum zadání diplomové práce: **30. června 2020**
(datum prvního zadání této práce, které musí být nejpozději 10 měsíců před datem prvního předpokládaného odevzdání této práce vyplývajícího ze standardní doby studia)

Datum odevzdání diplomové práce: **1. prosince 2021**
a) datum prvního předpokládaného odevzdání práce vyplývající ze standardní doby studia a z doporučeného časového plánu studia
b) v případě odkladu odevzdání práce následující datum odevzdání práce vyplývající z doporučeného časového plánu studia


doc. Ing. Tomáš Horák, Ph.D.
vedoucí
Ústavu logistiky a managementu dopravy


doc. Ing. Pavel Hrubeš, Ph.D.
děkan fakulty

Potvrzuji převzetí zadání diplomové práce.


Bc. Martin Štourač
jméno a podpis studenta

V Praze dne.....20. srpna 2021

Poděkování

Na tomto místě bych rád poděkoval všem, kteří mi poskytli podporu a podklady pro vypracování této práce. Zvláště pak děkuji Ing. Alexandře Dvořáčkové, Ph.D. a Ing. Liboru Žídkovi, za odborné vedení a konzultování diplomové práce a za rady, které mi poskytovali po celou dobu mého studia. Dále bych chtěl poděkovat firmě Intellmaps s.r.o. za možnost spolupráce. V neposlední řadě je mou milou povinností poděkovat svým blízkým a rodičům za podporu, které se mi od nich dostávalo po celou dobu studia.

Prohlášení

Předkládám tímto k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na univerzitě ČVUT fakultě dopravní.

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Nemám závažný důvod proti užívání tohoto školního díla ve smyslu § 60 Zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Praze dne 30. listopadu 2021

.....

Podpis

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta dopravní

Navigační systémy pro vnitřní prostory

Diplomová práce
prosinec 2021
Bc. Martin Štourač

Abstrakt

Tato diplomová práce prezentuje projekt vývoje vnitřní navigace. Projekt byl zahájen v roce 2020 a zaměřoval se na vývoj a implementaci vnitřní navigace pro zařízení Poliklinika Prosek v Praze. Projekt byl realizován ve spolupráci se start-upovou společností Intellmaps.

První část práce popisuje teoretické poznatky o navigaci a o teorii grafů, které jsou později v práci aplikovány. Hlavní část detailně popisuje návrh, vývoj a samotnou implementaci vnitřní navigace. Závěrečná část hodnotí dosažené výsledky samotného projektu vnitřní navigace.

Abstract

This master thesis presents the project indoor navigation. Started in 2020, the project focused on development and implementation of an indoor – navigation for premises of the hospital Prosek in Prague. The project was facilitated under “the umbrella” of the startup company Intellmaps.

This paper begins with an introduction to the navigation landscape, and then it outlines the fundamental theory applied in the project. The main part describes in detail the project development, the indoor – navigation design and its implementation for the specified premises. The final part summarizes and evaluates the project and its results.

Klíčová slova

vnitřní navigace, QGIS, PostgreSQL, pgRouting, Dijkstrův algoritmus, navigování, Poliklinika Prosek, web technologie

Key words

indoor navigation, QGIS, PostgreSQL, pgRouting, Dijkstra algorithm, navigation, Health Center Prosek, web technologies

Obsah

1	Úvod.....	8
1.1	Téma diplomové práce: projekt vnitřní navigace.....	8
1.2	Osobní motivace: proč a jak vznikl projekt 'indoor' navigace	9
2	Navigace	11
2.1	Navigace a její vývoj.....	11
2.2	Vnitřní navigace	12
2.3	Vnitřní navigace a její využití v logistice.....	16
2.4	Porovnání nabídky vnitřních navigací na současném trhu	19
3	Teoretické základy vnitřní navigace.....	21
3.1	Základní pojmy.....	22
3.2	Cesty na grafech	26
3.2.1	Dijkstrův algoritmus	28
3.3	Softwarová podpora výpočtu algoritmů.....	30
3.3.1	PgRouting	31
3.3.2	Historie.....	31
3.3.3	Funkce pgRoutingu	32
3.3.4	Výpočet Dijkstrova algoritmu v pgRouting	34
4	Projekt vnitřní navigace	37
4.1	SDLC	39
4.1.1	Spirálová metoda	41
4.2	Zahájení projektu.....	42
4.2.1	Hlavní cíl projektu vnitřní navigace:	42
4.2.2	Objekt vnitřní navigace	46
4.2.3	Poliklinika Prosek – Objekt vnitřní navigace	46
4.2.4	Hlavní body projektu.....	48
4.3	Plánování projektu – První etapa projektu	51
4.3.1	Software QGIS	54
4.4	Realizace Projektu – druhá etapa.....	55
4.4.1	Databáze PostgreSQL	57
4.5	Vývojový postup vnitřní navigace – druhá etapa.....	58
4.6	Vývoj druhé verze vnitřní navigace – Třetí etapa.....	60
4.6.1	Datové zmapování objektu	61
4.7	Vývojový postup vnitřní navigace – Třetí etapa	63

4.8	Vnitřní navigace – Výsledek a vyhodnocení projektu, zpětná vazba	67
4.8.1	Souhrnné vyhodnocení splnění požadavků projektu.....	67
4.8.2	Shrnutí výsledků.....	68
4.8.3	Zpětná vazba na aplikaci vnitřní navigace	70
4.8.4	Vyhodnocení zpětné vazby na aplikaci vnitřní navigace	71
5	Závěr.....	74
6	Citovaná literatura	77
7	Seznam obrázků	79
8	Seznam tabulek	80
9	Seznam příloh	81

Seznam zkratk

GPS – Global Positioning Systém

SDLC – Cyklus vývoje počítačového softwaru

QGIS – Quantum Geographic Information System

Alg – Algoritmus

JS – skriptovací jazyk JavaScript

Pgr_Dijkstra – Počítačová funkce Dijkstrova algoritmu

1 Úvod

1.1 Téma diplomové práce: projekt vnitřní navigace

Ve své diplomové práci popisuji, analyzuji a vyhodnocuji projekt vývoje a implementace vnitřního 'indoor' navigačního systému pro konkrétní vybraný objekt, v jehož rámci jsem vytvořil návrh architektury systému založeného na praktickém využití teorie grafů a realizoval ji v praxi.

Mým původním záměrem bylo vytvořit pouze návrh architektury systému založeného na praktickém využití teorie grafů, který by byl základem pro vývoj navigace ve vnitřních prostorách, tzv. 'indoor' navigace. Protože jsem na návrhu spolupracoval s technologickou start-upovou společností Intellmaps s.r.o., která se zaměřuje na vývoj interaktivních mapových aplikací a navigačních informačních systémů, vyvíjených prostřednictvím moderních technologií a jejich následnou implementací, rozšířili jsme původní záměr i o následné praktické využití návrhu této architektury při realizaci konkrétní 'indoor' navigace. K realizaci rozšířeného záměru jsme přistoupili formou aplikace základních prvků projektového řízení, protože už na počátku bylo zřejmé, že bude nutná koordinace a řízení více činnosti i osob v delším časovém období. Tak vznikl základ projektu 'indoor' navigace, jehož předpokládaným výstupem bylo vytvoření pilotní verze vnitřní navigace, aplikované na konkrétní objekt polikliniky.

Svoji práci jsem proto pojal jako popis, analýzu, a vyhodnocení jednotlivých fází tohoto projektu vývoje a implementace vnitřního 'indoor' navigačního systému pro konkrétní vybraný objekt. Cílem projektu bylo nejen dodat výše předpokládaný výstup pilotní verze vnitřní navigace v dostatečné kvalitě a stanoveném termínu, ale zejména vytvořit architekturu a řešení, které jsou obecně využitelné pro vývoj aplikací pro navigování ve vnitřních prostorách objektů, včetně vícepodlažních. Aby vývoj těchto aplikací probíhal za přijatelných nákladů, je řešení postaveno na použití dostupných technologií, zejména open-source softwaru.

Projekt probíhal v období od října 2020 do září 2021 ve výše uvedené společnosti Intellmaps a kromě mne se ho účastnili kolegové a spolupracovníci z útvarů webového programování a geografického plánování společností Intellmaps. Konkrétním vnitřním objektem, pro který byla vytvářena vnitřní navigace, byla Poliklinika Prosek, jež byla vybrána po vzájemné dohodě o spolupráci na pilotním projektu mezi společností Intellmaps a Poliklinikou Prosek.

Struktura diplomové práce je vzhledem k jejímu cíli co nejvěrněji zobrazit genezi, průběh a výsledky celého projektu následující:

- V kapitole 2 je popsán evoluční vývoj navigace jako takové, principy fungování vnitřní navigace a způsoby jejího využití, včetně porovnání nabídky vnitřních 'indoor' navigací na současném trhu, aby v projektu byla zřejmá pozice, z níž projekt může vycházet.
- V kapitole 3 jsou obsaženy zejména teoretické předpoklady projektu reprezentované teorií grafů, která byla v projektu využita jako teoretický základ vnitřní navigace, včetně popisu operací na grafech a algoritmů, které v projektu při vývoji vnitřní navigace našly uplatnění, a dále pak i praktické předpoklady spočívající v popisu použité softwarové podpory pro výpočet algoritmů s konkrétními odkazy na aplikaci těchto předpokladů v projektu.
- Kapitola 4 začíná praktická část mé diplomové práce, která je věnována vlastnímu průběhu projektu. Kromě hlavních cílů projektu, popisu objektu řešeného projektem a očekávaných přínosů řešení vnitřní navigace pro tento objekt, obsahuje zejména popis jednotlivých fází projektu, v nichž je zvláštní pozornost věnována výběru softwarového řešení a technologickým možnostem, které byly v projektu zvoleny. V závěru kapitoly 4 jsou prezentovány výstupy projektu, jeho vyhodnocení z pohledu splnění stanovených cílů, vyhodnocení zpětné vazby uživatelů na výsledný produkt projektu, kterým je návrh vnitřní navigace, i možnosti jeho dalšího rozvoje.
- Kapitola 5 obsahuje závěr diplomové práce se shrnutím jejich jednotlivých částí i nastíněním dalšího potenciálu vnitřních navigací a příkladů jejich možného využití.

1.2 Osobní motivace: proč a jak vznikl projekt 'indoor' navigace

Když mě společnost Intellmaps oslovila s nabídkou spolupráce v oblasti tvorby vnitřních navigačních systémů, měla již značnou zkušenost v oblasti technologií pro interaktivní zobrazování územních plánů a lokačních analýz. Jejich dalším záměrem bylo rozšířit svoji nabídku o řešení navigace pro vnitřní prostory, 'indoor' navigace. Důvodů, proč mě tato nabídka oslovila, bylo několik. V první řadě to byla zjevná souvislost tématu s mým studijním zaměřením doprava/logistika a možnost aplikovat studiem získané teoretické vědomosti, zejména z oblasti teorie grafů, v konkrétním projektu. Dále pak můj zájem o moderní technologie, možnost pracovat v dynamickém týmu mladé start-upové společnosti a vyzkoušet si vedení projektové aktivity. A v neposlední řadě i to, že v současném světě stále více založeném na moderních technologiích vidím velký potenciál využití vnitřních navigací, a to jak pro objekty veřejných, dopravních i soukromých budov, tak v průmyslu i logistice, kde svým uživatelům ušetří nejen cenný čas, ale mohou přispět i ke snížení výrobních nákladů.

Návrh na spolupráci jsem přijal s velkou motivací vytvořit pilotní produkt řešení vnitřní navigace postavené na aplikaci teorie grafů. Spolupráce pro mě byla i jedinečnou příležitostí vyzkoušet si principy projektového managementu v praxi, samostatně realizovat projekt ve všech jeho fázích, jak je popisuji v kapitole 4., a také uplatnit a rozšířit si své znalosti v oblasti programování.

Při mém rozhodování sehrála významnou roli i potřeba stále zkoušet hledat nové cesty a prostředky, jak jich dosáhnout, čemuž téma vnitřní navigace a možnosti vyvinout návrh potenciálně nového konceptu její architektury velmi dobře odpovídalo.

2 Navigace

2.1 Navigace a její vývoj

Slovo navigace pochází z latiny a spojuje slova „navis“, míněno loď, a slovo „agere“, míněno pohyb nebo směr. Slovo navigace se rozlišuje podle dané aktivity. Například pánové Sonnenberg a Hofmann-Wellenhof definovali navigaci ve spojení s vozidlem nebo objektem, a nikoliv s člověkem. Naproti tomu pánové Prasad a Ruggieri s Kaplan a Hegarty vnímají člověka jako objekt, který je součástí navigačního procesu (1). Definice navigace existuje v odborné literatuře mnoho. Základní princip všech definic je až na drobné odchylky ve výkladu podobný. V navigaci je vždy nutné určit počáteční bod stavu a koncový bod stavu navigované cesty. Kritérii pro určení optimální cesty jsou nejkratší cesta, nejrychlejší cesta nebo nejekonomičtější cesta.

Navigace a vývoj navigačních technik se datují od starověku a souvisí s rozvojem dopravy, zejména pak mořeplavby. Zatímco navigace na pevnině nebo v její blízkosti byla možná pomocí orientačních bodů, často i uměle vytvořených, na volném moři tato možnost chyběla.

Základním způsobem navigace na moři byla astronavigace, tedy orientace na základě pozorování nebeských těles, jako je Slunce, Měsíc a hvězdy (1). První opravdovou navigační pomůckou se stal kompas, který byl vymyšlen již ve 4. století a do mořeplavectví se dostal koncem 12. století. Díky práci s magnetickými deklinacemi dokázal určit směr světových stran.

Nevýznamnější technologický pokrok v oblasti navigací však nastal až v éře dobývání vesmíru, a to od doby, kdy byla do vesmíru dopravena první družice.

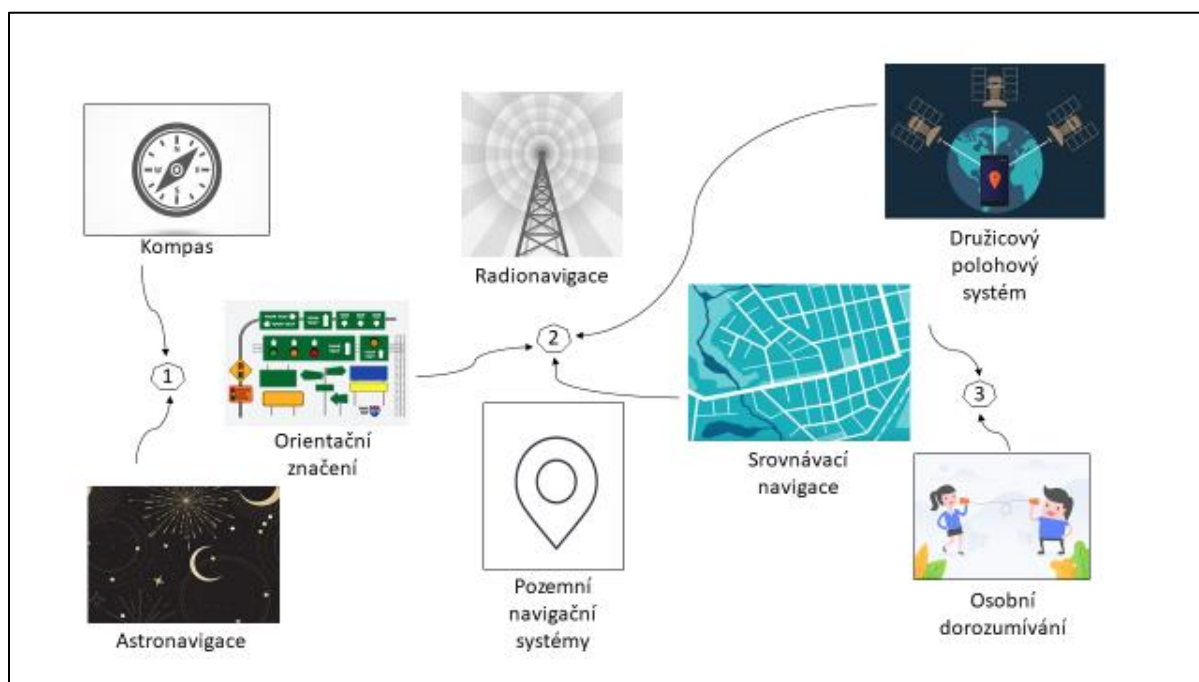
První družicový navigační systém, který byl uveden do provozu Spojenými státy americkými roku 1964 pro potřeby vojenského námořnictva, nesl název TRANSIT. Od roku 1967 byl tento systém uvolněn i pro civilní použití. Byl tvořen šesti družicemi, které obíhaly po polární oběžné dráze ve výšce 1075 km, a třemi pozorovacími stanicemi umístěnými na území USA. Doba oběhu družice byla přibližně 107 minut. Přesnost lokalizace uživatele se postupně zvyšovala (z původních 800 metrů na pozdějších 5 metrů). Data byla přenášena rádiovými vlnami o frekvenci 149,988 a 399,968 MHz a výkonu 2 W. Zpráva měla podobu dvouminutové relace obsahující 6103 bitů uspořádaných do slov. Údaje o drahách družice přenášelo 25 slov, z nichž každé mělo 39 bitů. Z celkových 182 slov jich 157 bylo vyhrazeno pro přenos vojenských informací. Nevýhodou systému bylo to, že získané souřadnice byly pouze dvourozměrné (2).

Navigační techniky, které se vyvíjely od pravěku až po současnost, a jsou využívány i dnes, lze systematicky seřadit do tabulkové posloupnosti podle jejich postupného vývoje:

Osobní dorozumívání	⇒ osobní komunikace (hlasová, vizuální)
Orientační značení	⇒ <u>dopravní značení</u> , <u>turistické značení</u> , <u>značení ulic a domů</u>
Srovnávací navigace	⇒ porovnávání terénu a <u>mapy</u>
Terestrický	⇒ kompas, hloubkoměr
Astronavigace	⇒ pomocí polohy <u>Slunce</u> , <u>Měsíce</u> a <u>hvězd</u>
Radionavigace	⇒ pomocí radiomajáků (<u>NDB</u> , <u>VOR</u>)
Pozemní navigace	⇒ pozemní navigační systémy – např. <u>LORAN</u> , <u>OMEGA</u>
Družicové systémy	⇒ GPS, Galonass

Tabulka 1 - Druhy navigačních technik

Tyto techniky a systémy jsou často využívány spolu, tak, jak je zobrazeno na obrázku 1 dle schématu tabulky 1:



Obrázek 1 - Navigace a jejich kombinace

- ① Námořní navigace v minulosti
- ② Silniční navigace v současnosti
- ③ 'Dětská' navigace

2.2 Vnitřní navigace

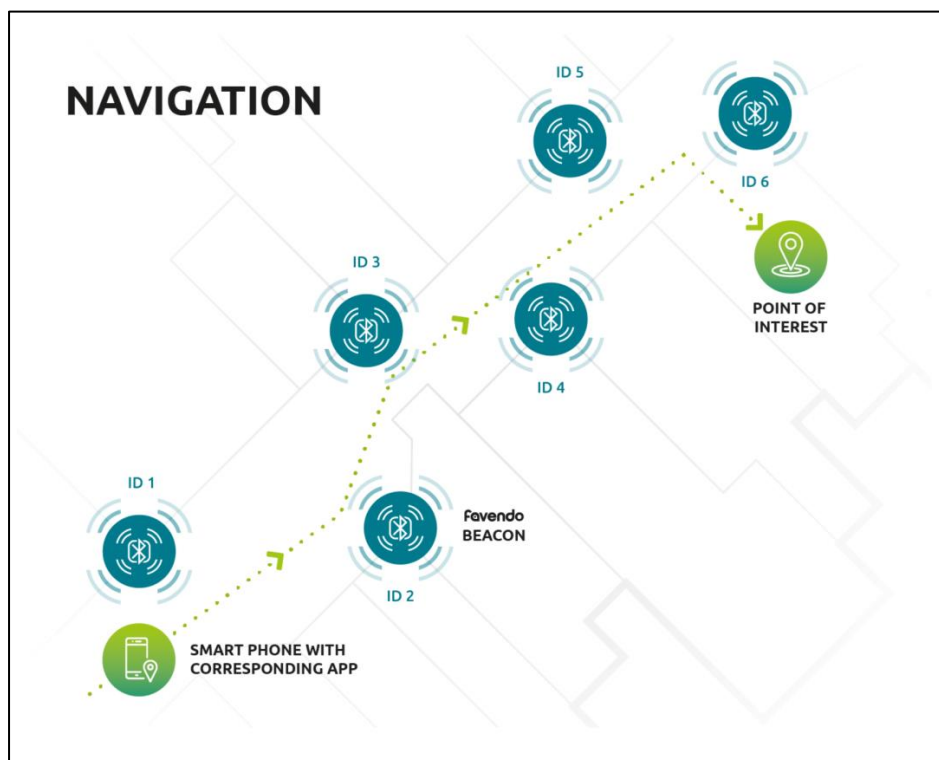
Klasická navigace, tj. navigace v terénu, neboli 'outdoor' navigace, kterou jsou například navigace integrované v automobilech, turistické navigace nebo mapové aplikace v

chytrých zařízení (mobilních telefonech nebo tabletech), je dnes již široce rozšířena a známa. Jejím cílem je dovést uživatele v neznámém prostředí z bodu A do bodu B po předem určené trase. Oproti tomu s vnitřní neboli 'indoor' navigací se nesetkáváme v běžném životě tak často, ale jistě každý z nás byl v situaci, kdy vstoupil do neznámé budovy (objektu) a snažil se v ní dostat do určitého cíle, a ne vždy se mu to podařilo na první pokus, rychle a efektivně.

A právě v takových situacích má pomoci vnitřní navigace, jež by jistě usnadnila mnoha uživatelům jejich 'cestování' po budovách a objektech.

Definice vnitřní navigace:

Vnitřní navigace v podstatě znamená flexibilní vedení lidí v nepřehledných či neznámých budovách a komplexech budov. Uživatel vnitřní navigace se zobrazí jeho vlastní poloha na mapě budovy na jeho vlastním chytrém zařízení. Po výběru cíle nebo bodu zájmu se na mapě zobrazí trasa do zvoleného cíle. Poloha uživatele v budově a jejím okolí je trvale přepočítávána pomocí vnitřní lokalizace, takže na zařízení uživatele je vždy zobrazena aktuální poloha na trase (3).



Obrázek 2 – Vnitřní navigace (3)

Funkce vnitřní navigace

Fungování vnitřní navigace je založené na stejném principu, jako je například fungování turistické navigace, tj. nalézt v neznámém prostředí cílový bod, zadat ho do navigace, která uživateli vykreslí nejkratší trasu z aktuálního umístění do cílového bodu a případně bude uživatele po trase k cílovému bodu navigovat. Pro přístup uživatele k vnitřní navigaci z hlediska možnosti jejího využití se zpravidla používají dva následující způsoby:

1) Přístup z displeje pevně umístěného monitoru ve formě navigačního kiosku,

2) Přístup z aplikace nainstalované na chytrém zařízení (mobilním telefonu nebo tabletu) uživatele.

Možnosti způsobu uvedeného pod bodem 1) mají nevýhodu, protože uživateli nemůže být průběžně zobrazována jeho aktuální poloha, a navigování má tedy charakter pouze jednorázového zobrazení. Využívá se v navigačních kioscích umístěných při vstupech do budov (např. nákupních center), kdy uživatel u kiosku v navigaci vyhledá svůj cíl a zobrazí se mu nejkratší cesta, jak se k němu dostat. Pokud se uživatel dostatečně neorientuje nebo si trasu dostatečně nezapamatuje a cestou k cíli se z trasy odchýlí a cíle nedosáhne, plně se projeví hlavní nedostatek tohoto přístupu, kterým je, jak už bylo uvedeno výše, nemožnost zobrazování aktuální polohy. V takovém případě nemá uživatel, který chce navigaci použít, jinou možnost než se vrátit k navigačnímu kiosku a zadat svůj cíl znovu.

Další možností uvedenou pod bodem 2) je instalace aplikace vnitřní navigace do chytrého zařízení uživatele (mobilního telefonu nebo tabletu), které disponuje technologií pro určení polohy (lokalizaci) zařízení, kdy navigace využije určení aktuální polohy zařízení jako polohy uživatele a z této polohy vykreslí nejkratší cestu do uživatelem vybraného cílového bodu v budově. Uživatel může průběžně sledovat svoji polohu na displeji zařízení a kopírovat směr navigační čáry až do cíle. Tento způsob je přesnější a pro uživatele komfortnější, ale realizačně obtížnější, protože určování polohy zařízení v uzavřeném objektu je mnohem složitější než ve volném terénu. Velmi omezujícím faktorem je skutečnost, že uvnitř budov není použitelná služba GPS. Určování polohy v objektu bývá řešeno různými způsoby, zejména umístěním bezdrátových směrových senzorů, které komunikují se zařízením, nebo připojením k signálu Wi-Fi a využitím Wi-Fi triangulace, tedy způsobu zjišťování souřadnic a vzdáleností založeném na trigonometrickém výpočtu ze známé polohy umístění známých Wi-Fi bodů nebo metodou Finger-prints, což je mapa tzv. otisků a určování polohy podle předem vyměřené mapy.

Finanční kalkulace vnitřních navigací:

V části funkce vnitřní navigace jsou definována dvě aplikovatelná řešení pro použití vnitřního navigování po budovách. První řešení aplikuje navigování z pevně umístěného monitoru ve formě navigačního kiosku. Druhé řešení aplikuje navigování z chytrého zařízení uživatele, které probíhá dynamickým určování polohy uživatele. Určování dynamické polohy se provádí u druhého řešení způsobem získávání polohy z bezdrátových směrových senzorů nebo využitím Wi-Fi triangulace. Jedná se o tři různá technická řešení, která se liší i pořizovacími náklady při implementaci do objektů budov.

Na příkladu kalkulace si vypočteme pořizovací cenu dvou, respektive tří aplikovaných řešení pro vzorový jednopodlažní obdélníkový objekt o orientační rozloze 230 m² s hlavní chodbou uprostřed a šesti bočními chodbami. Při implementaci druhého, respektive třetího řešení kalkulujeme s umístěním 10 bezdrátových směrových senzorů a pěti přístupovými Wi-Fi body, které byly odhadnuty pro vzorový objekt. Další kalkulační položky z kalkulace pořizovacích nákladů jsou uvedeny v tabulce 2.

Číslo	Položky kalkulace	Navigační kiosek	Navigování z chytrého zařízení	
			Pomocí senzorů	Pomocí Wi-Fi
1	Vytvoření web aplikace	150.000 Kč	X	X
2	Vytvoření mobilní aplikace	X	300.000 Kč	300.000 Kč
3	Pořízení počítače + dotykového monitoru	35.000 Kč	X	X
4	Pořízení 10 bezdrátových senzorů (Cena pořízení 1000 Kč/Ks)	X	10.000 Kč	X
5	Wi-Fi Router	X	X	5.000 Kč
6	5 přístupových Wi-Fi bodů (Cena 2500 Kč/Ks)	X	X	12.500 Kč
7	Přístup k internetu (1 rok)	X	X	12.000 Kč
8	Instalace	7.000 Kč	5.000 Kč	10.000 Kč
9	Údržba	3.000 Kč	5.000 Kč	7.000 Kč
	Celkem	195.000 Kč	320.000 Kč	339.507 Kč

Tabulka 2 - Kalkulace pořizovacích nákladů indoor navigací

Nákladové položky jsou pouze odhadem skutečných pořizovacích cen. Kalkulace pořizovacích nákladů má znázornit finanční rozdílnost mezi prvním a druhým, respektive třetím aplikovaným řešením pro vnitřní navigování v budovách. První aplikované řešení (navigační kiosek) je technicky jednodušší ve srovnání s druhým aplikovaným řešením (navigování z chytrého zařízení – pomocí senzorů), respektive třetím aplikovaným řešením

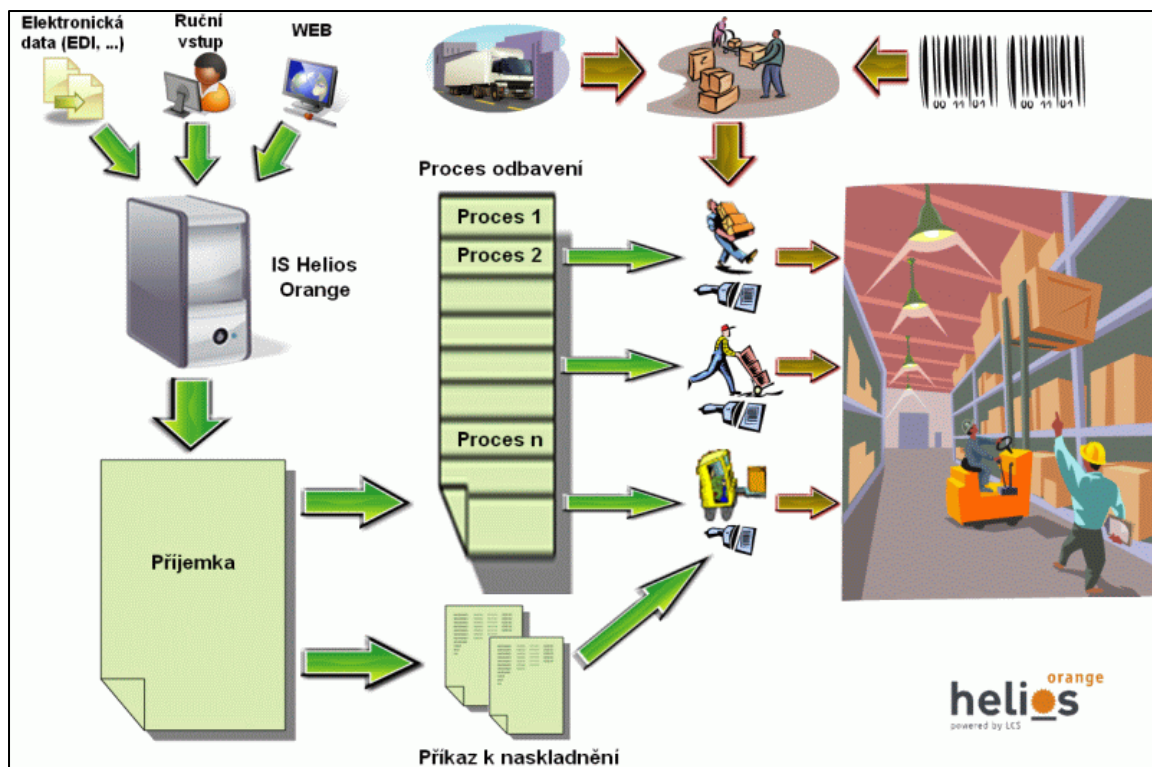
(navigování z chytrého zařízení – pomocí Wi-Fi), ale finančně výhodnější na pořízení, viz tabulka 2.

2.3 Vnitřní navigace a její využití v logistice

Využití vnitřních navigací teprve pomalu naplňuje svůj potenciál na trhu s technologiemi. V logistice nebo skladovém hospodářství se s navigačními systémy setkáme často ve formě integrovaných systémů, které využívají zaměstnanci ke své práci. Možné grafické rozhraní bude u navigačního systému využívat jiné principy, než je dále uváděno v dalších částech této práce. V současnosti se lze s vnitřní navigací setkat poměrně zřídka, pokud není člověk zaměstnancem velkého distribučního nebo logistického centra. Často se vyskytuje v obchodních centrech ve formě kiosků s dotykovými panely zobrazujícími cestu od kiosku do uživatelem zadaného obchodu. S tímto typem vnitřní navigace se občas lze setkat i v nemocničním prostředí. Výskyt vnitřních navigací pro externího návštěvníka objektu je velmi nízký. Např. v literatuře věnované oblasti obchodu panuje tento názor:

„Největší potenciál má vnitřní navigace v retailu a prostorách obchodních center. Díky inteligentním funkcím, jako je třeba nákupní seznam, zákazník ušetří čas a nakupovat bude touto cestou mnohem častěji. Navigace mu naplánuje nejrychlejší trasu od položky k položce z jeho nákupního seznamu“ (4).

Možnost využití vnitřní navigace se tedy nabízí například v supermarketu, kdy by si uživatel předem vytvořil seznam zboží, které potřebuje nakoupit, a navigace by mu vytvořila optimální trasu k jednotlivým položkám jeho seznamu. Na stejném principu pracují navigační systémy ve skladové logistice.

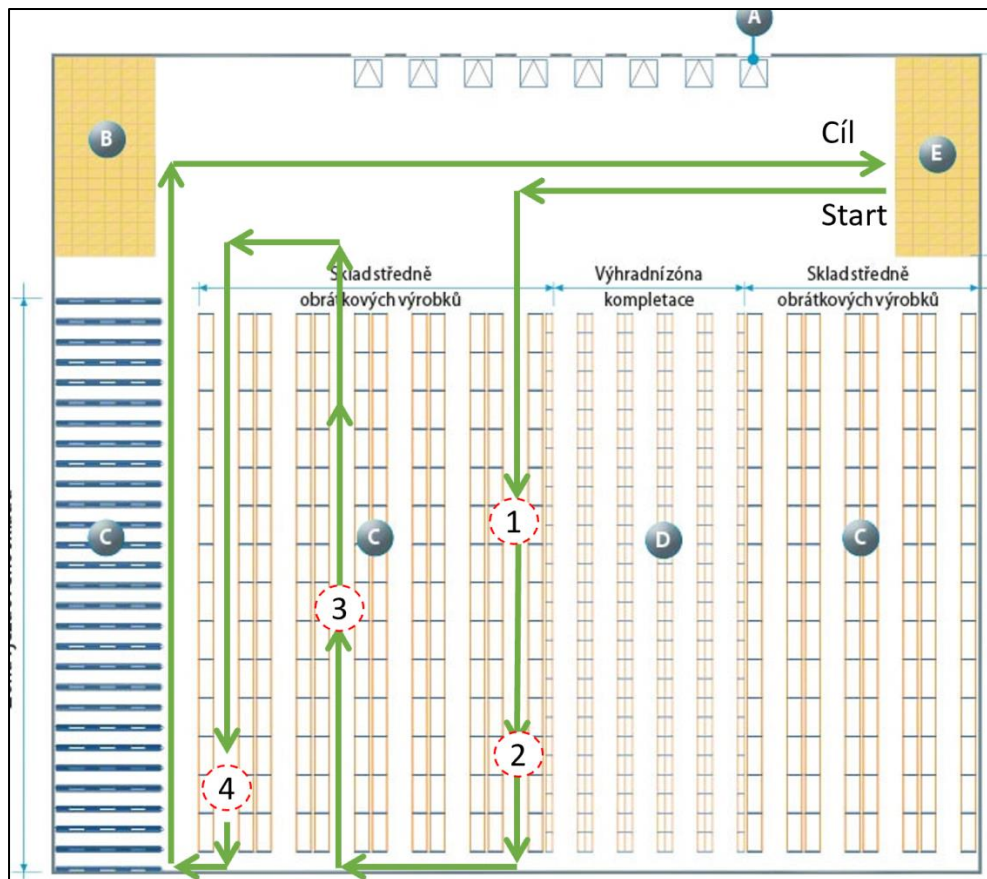


Obrázek 3 - Přejem zboží na sklad (5).

Co se týče využití vnitřní navigace v logistice, níže jsou uvedeny dva vzorové příklady využití:

a) Proces naskladnění s vnitřní navigací:

- Skladník obdrží příkaz k naskladnění obsahující seznam zboží, které má být naskladněno na již přesně stanovená regálová místa ve skladu.
- Skladník příkaz nahraje jako seznam položek, které musí být rozvezeny na konkrétní předem známá místa ve skladu, do aplikace vnitřní navigace.
- Aplikace vnitřní navigace, v níž je pro každý druh zboží předem přiřazen cílový bod – konkrétní regálové místo ve skladu, vygeneruje skladníkovi optimální trasu rozvezení jednotlivých balení zboží podle jejich druhu po skladu.



Obrázek 4 - Vnitřní navigace ve skladu (6)

b) Proces rozvozu dodávek (Batchs) ze skladu do výrobní haly:

- Pracovník skladové logistiky obdrží seznam dodávek (tzv. Batchs), které má vyzvednout ze skladu a rozvést po výrobní hale.
- Seznam nahraje do aplikace vnitřní navigace.
- Navigace mu vytvoří plán rozvozu a na displeji přesně ukáže, z kterého místa ve skladu, na které místo ve výrobní hale má přemístit určenou dodávku, a to včetně určení optimální trasy rozvozu.

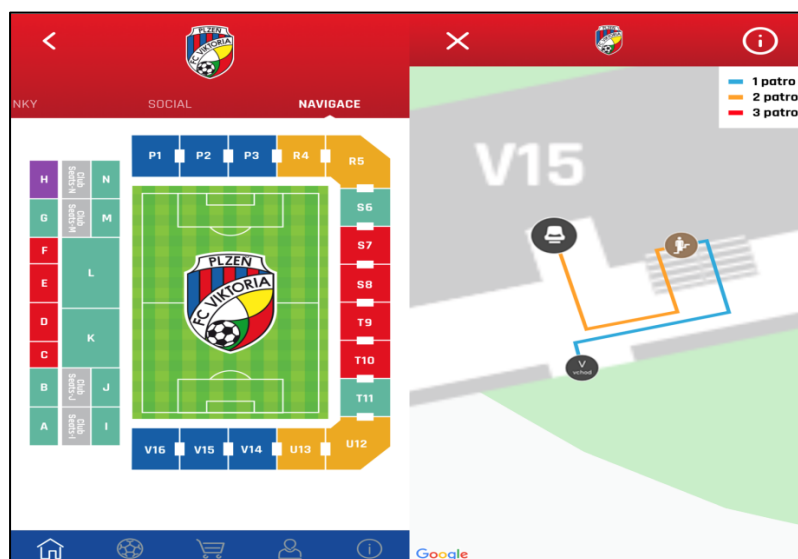
Podobných příkladů, jak by bylo možné využít vnitřní navigaci v logistice, je mnoho. Nabízející se řešení však zatím nejsou dostatečně převedena do praxe a lze konstatovat, že ve skladovém a výrobním hospodářství se masově nevyužívají, samozřejmě s výjimkou velkých logistických center, která mají svá vlastní řešení vyvinutá přesně na míru jejich potřebám. Zejména u malých a středních podniků bude v tom, že podobná řešení nevyužívají v takovém měřítku, jak by bylo možné, a mnohdy o jejich využití ani neuvažují, hrát roli stávající finanční náročnost řešení šitých na míru konkrétnímu podniku.

2.4 Nabídky vnitřních navigací na současném trhu pro externí uživatele

Vnitřní navigace jsou již na trhu dostupné a první řešení jsou již nabízena technologickými společnostmi i na českém trhu. Možná ne v tak vysoké míře jako například ve Spojených státech amerických, kde vývoj vnitřních navigací podporují velké korporátní společnosti, jako jsou Google nebo Here Maps, nicméně i v České republice lze získat nabídku na vytvoření vnitřní indoor navigace například od firem:

- Sophis
- Alternetivo
- Symbion
- Artin
- Správa Informačních technologií města Plzně

Nabídka však ještě nemusí znamenat úspěšnou realizaci v praxi. Ale například společnost Správa Informačních technologií města Plzně na svých internetových stránkách prezentuje konkrétní příklad již realizovaného řešení, kterým je aplikace s názvem 'FCViktoriaPlzen', která je aplikací vnitřní navigace pro fotbalový stadion FC Viktoria Plzeň. Tato navigace pomáhá návštěvníkům s orientací a nasměrováním do jednotlivých sektorů stadionu a lze v ní vyhledat cestu k nejbližším toaletám nebo stánkům s občerstvením, u kterých je v navigaci možné zjistit i nabídku jednotlivých prodejců (7).



Obrázek 5 - Navigace aplikace FCViktoriaPlzen (8).

Aplikaci 'FCViktoriaPlzeň' lze volně stáhnout v obchodech s aplikacemi, Apple Storu pro zařízení iPhone nebo z Google play pro zařízení Android. Část navigace je volně dostupná a vyžaduje připojení na Wi-Fi Viktoria Plzeň, ale navigování po objektu si lze v aplikaci vyzkoušet i bez doporučeného připojení. Na obrázku 5 v jeho levé části můžeme vidět uživatelské rozhraní pro vnitřní navigaci a na pravé straně obrázku samotné navigování do sektoru V15.

Výhody a nevýhody navigace 'FCViktoriaPlzeň':

Výhody navigace	Nevýhody navigace
Integrace do aplikace	Chybějící navigování mezi patry
Spojení s Wi-Fi	Chybějící navádění k sedačkám
Přehledný koncept	Nemožnost sdílení polohy

Tabulka 3 - Výhody a nevýhody aplikace

Příklad vnitřní navigace 'FCViktoriaPlzeň' názorně ukazuje povedený koncept vnitřní navigace. Koncept je přesně postaven a upraven pro potřeby sportovního stadionu. Koncept je téměř bezchybný, jako malé negativum lze možná zmínit chybějící navigování mezi patry nebo chybějící možnost vytvořit navigační cílové body samostatně pro každou sedačku nebo alespoň řadu v určitém sektoru.

3 Teoretické základy vnitřní navigace

Teoretickým základem pro moderní řešení navigačních systémů může být teorie grafů. V teorii grafů se můžeme setkat s řešením různých problémů, které lze aplikovat na grafy. V případě teorie grafů je graf reprezentován určitou množinou vrcholů a množinou hran, které mohou být vzájemně propojeny.

Graf je možné si představit jako zjednodušení reálného světa, kdy daný problém znázorníme pomocí bodů a čar, které je spojují. Tyto body se nazývají uzly (vrcholy) grafu a spojnice mezi nimi se nazývají hrany grafu (9). Příkladem reálného využití grafů může být například železniční síť, kde jednotlivé vrcholy reprezentují města a hrany železniční trasy mezi vrcholy, městy. Dalším příkladem grafu z reálného světa může být znázornění turistických tras, kde vrcholy grafu znázorňují křižovatky neboli rozcestníky a hrany přímé cesty mezi těmito rozcestníky.

Graf může reprezentovat jakýkoliv problém, síť, uspořádání prvků, algoritmus a podobně. Grafy mají využití nejen v informatice a matematice, ale také např. ve fyzice, v chemii, v elektrotechnice, v ekonomii nebo v sociologii. Grafy můžeme využít k hledání nejkratší cesty mezi dvěma místy, k řešení úloh umělé inteligence při hraní šachů nebo k řešení mnohých dalších úloh.

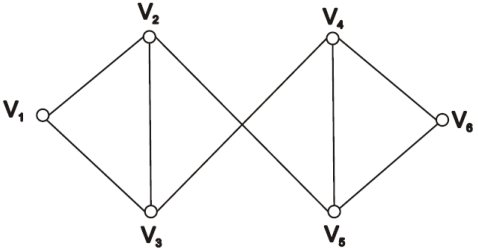
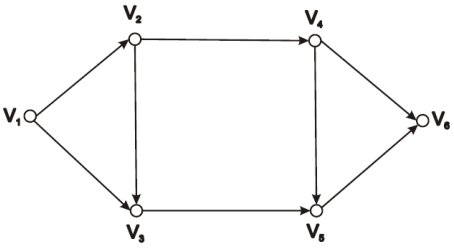
3.1 Základní pojmy

V odborné literatuře se lze setkat s různými pojmy a terminologií pro jednotlivé soubory množiny v teorii grafů. Proto je vhodné si vydefinovat jasně určené pojmy, které budou i v dalších částech práce využity, tak aby byla zachována jednotnost terminologie v rozsahu celé práce. Jako student Fakulty dopravní ČVUT v Praze bych rád využil stejnou terminologii pro teorii grafů, která je uvedena v učebních plánech předmětu Teorie grafů na fakultě dopravní, vyučovaného doc. Ing. Denisou Mockovou, Ph.D. Pojmy jsou přehledně seřazeny ve strukturované tabulce podle skript Teorie grafů (10).

Tabulka pojmů:

Sled	Je posloupnost po sobě následujících vrcholů a hran, která začíná a končí ve vrcholu. Uzavřený sled začíná a končí ve stejném vrcholu. Otevřený sled začíná v jednom vrcholu a v jiném končí. Druhy sledů: a) Neorientovaný sled b) Orientovaný sled c) Uzavřený neorientovaný sled d) Uzavřený orientovaný sled e) Otevřený neorientovaný sled f) Otevřený orientovaný sled
Tah	Je sled, ve kterém se neopakuje žádná hrana.
Cesta	Je sled, ve kterém se neopakuje žádný vrchol.
Dráha	Je orientovaný sled, ve kterém se neopakuje žádný vrchol ani hrana, je zároveň orientovanou cestou a tahem.
Délka cesty	Mezi dvěma vrcholy $u, v \in V$ v hranově ohodnoceném grafu G platí: $ m(u, v) = \sum_{h \in m(u, v)} o(h)$ kde $m(u, v)$ je označení cesty, která začíná ve vrcholu u a končí ve vrcholu v , $o(h)$ je ohodnocení hrany $h \in E$.
Vzdálenost	Pro vzdálenost dvou vrcholů $u, v \in V$ grafu G platí: $d(u, v) = \min \left\{ \sum_{h \in m(u, v)} o(h) \right\},$ kde M je množina všech cest mezi vrcholy u a v .
Most	Mostem je hrana, jejímž odstraněním z grafu G se graf rozpadne na dvě komponenty, resp. počet komponent se zvětší právě o jednu.
Artikulace	Artikulací je vrchol, jehož odstraněním z grafu G , včetně iniciujících hran, vzniknou dvě a více komponent. Počet komponent se zvýší právě o jednu.

Tabulka 4 - Teorie grafů (10)

Neorientovaný graf	Orientovaný graf (digraf)
<p>Definice neorientovaného grafu: Rozumíme uspořádanou trojici $G = (V, X \text{ resp. } H, p)$, kde V a X/H jsou množiny, přičemž V je konečná neprázdná množina a p je prosté zobrazení množiny X do množiny všech neuspořádaných (u, v), $u, v \in V, u \neq v$. Prvky množiny V nazýváme vrcholy grafu G, prvky množiny X hranami grafu G a zobrazení p incidencí grafu G. Incidence p přiřazuje každé hraně grafu neuspořádanou dvojici vrcholů. Platí-li pro incidenci hrany $h \in X$, $p(h) = (u, v)$, říkáme, že hrana h indikuje s vrcholem u i s vrcholem v. Vrcholy u, v nazýváme krajními vrcholy hrany h. Někdy budeme říkat, že tyto vrcholy jsou sousední, nebo také, že spolu sousedí. Protože každá hrana může být vyjádřena jako neuspořádaná dvojice sousedních vrcholů, je zápis $p(h) = (u, v)$ ekvivalentní se zápisem $p(h) = (v, u)$. Neorientovaný graf: $G = (V, X, p)$ resp. $G = (V, H, p)$ nebo zkráceně $G = (V, X)$ resp. $G = (V, H)$</p>	<p>$D = (V, Y, p)$, kde prvky množiny Y jsou orientované hrany a p přiřazuje každé hraně uspořádanou dvojici vrcholů $p(h) = (v_i, v_j)$. Mohutnost množiny vrcholů = počet vrcholů grafu budeme značit n. $n = V$ Mohutnost množiny hran = počet hran grafu budeme značit q. $q = X$</p>
<p>Příklad grafu:</p> 	<p>Příklad grafu:</p> 
<p>Stupeň vrcholů (neorientovaný graf)</p>	<p>Stupeň vrcholů (orientovaný graf)</p>
<p>Nazýváme počet hran indikujících s vrcholem $v \in V$.</p>	<p>Stupeň vrcholu v orientovaném grafu je uspořádaná dvojice přirozených čísel (a, b), kde a je počet hran vycházejících z vrcholu v_i a b je počet hran vcházejících do vrcholu v_i.</p>
<p>V neorientovaném grafu $G = (V, X)$ platí:</p> $\sum_{v \in V} st(v_i) = 2q$	<p>V orientovaném grafu $D = (V, Y)$ platí:</p> $\sum_{v \in V} (a_{v_i} + b_{v_i}) = 2q$

Tabulka 5 - Typy grafů (10)

Ze základních pojmů pro cestování na grafech je v rámci další práce důležité se zaměřit na pojmy most a artikulace. Až se budeme v kapitole 4 zaměřovat na vytváření grafu pro vnitřní navigaci, budeme čelit problému grafu spojeného mostem a artikulací. Pro představu náš prostorový graf bude skrz jednotlivé výškové úrovně propojen skrze hrany tzv. mostu (horizontální propojení skrze výtahovou šachtu) nebo propojen skrze hrany a vrcholy, které tvoří artikulaci v prostorovém grafu (propojení skrze schodiště). A podle definice těchto dvou pojmů, pokud dojde k odstranění těchto propojení, graf se rozpadne na dvě a více komponent. Z tohoto důvodu je důležité se zaměřit na funkčnost a správnost určení předmětných hran a vrcholů v našem prostorovém grafu vnitřní navigace v projektu vnitřní navigace v kapitole 4.

3.1.1 Aparát teorie grafů a typy grafů.

Základní rozdělení grafů je na neorientovaný graf a orientovaný graf, které se dále kategorizují podle komponent přidaných do grafu, jako je smyčka nebo násobná hrana. Základní rozdělení grafů je znázorněno v tabulce 5 (10).

Kombinace obou grafů, obsahující jak hrany neorientované, tak orientované, se nazývá smíšený graf.

Podle výskytu orientovaných a neorientovaných hran nebo jejich kombinace jsou grafy kategorizovány jako 1) neorientovaný, 2) orientovaný – digraf, 3) smíšený – migraf.

Dále se v různých typech grafů mohou objevovat komponenty typu smyčka nebo násobná hrana, které mohou sloužit pro další operace na grafu. V případě projektu vnitřní navigace nebudou tyto komponenty využity. Z obecné definice grafů budeme využívat pouze neorientovaný graf, tzv. obyčejný graf.

Kategorizace grafů:

Tabulka 6 - Typy grafů

obyčejný	prostý	multigraf	pseudograf
digraf	prostý digraf	multidigraf	pseudodigraf
migraf	prostý migraf	multimigraf	pseudomigraf

Kde:

- Obyčejný – neobsahuje ani smyčky ani násobné hrany
- Prostý – jen smyčky (nenásobné)
- Multigraf – jen násobné hrany

- Pseudograf – násobné hrany a smyčky (i násobné smyčky)

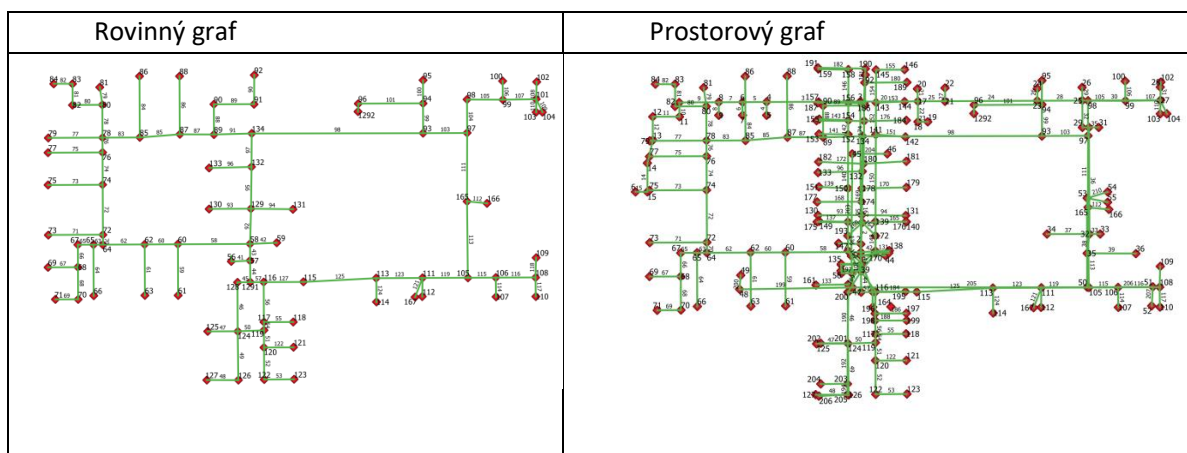
Vrcholové a hranové ohodnocení hran

Graf $G = (V, X, p)$ nazveme vrcholově (hranově) ohodnoceným grafem, pokud existuje funkce $o(v)$, resp. $o(h)$, která přiřadí každému vrcholu $v \in V$ (hraně $h \in X$) nezáporné číslo vyjadřující určitou kvantitativní nebo kvalitativní vlastnost vrcholu (hrany). Grafy mohou být vrcholově i hranově ohodnocené. Na graf, který není hranově ohodnocen, můžeme nahlížet jako na hranově ohodnocený graf, ve kterém je každé hraně h přiřazeno ohodnocení $o(h) = 1$. Podobně na graf, který není vrcholově ohodnocen, můžeme nahlížet jako na vrcholově ohodnocený graf, ve kterém je každému vrcholu přiřazeno ohodnocení $o(v) = 1$ (10).

V definici projektu budeme pracovat i s důležitým rozdělením mezi rovinným (planárním) a prostorovým grafem, kde dle definice je rozdělení definováno takto:

Graf nazveme rovinným, je-li možné jej zakreslit způsobem tak, aby se žádné z jeho hran neprotínaly jinde než ve vrcholech. V opačném případě se jedná o graf prostorový (10).

Podle definice lze přiblížit rovinný a prostorový graf na konkrétním příkladu z projektu vnitřní navigace, kde graf pro jedno podlaží představuje rovinný graf a graf pro všech 5 podlaží konkrétní budovy uvedené v kapitole 4 představuje prostorový graf.



Tabulka 7 - Rovinný a prostorový graf projektu vnitřní navigace

Ze základních pojmů byly v možné míře vybrány všechny důležité pojmy, které úzce souvisejí s kapitolou 4 a je nutné je mít jednotně vydefinovány pro celý projekt. V rámci projektu, kde dva a více lidí kooperuje na jedné věci v rámci vnitřní navigace, pomáhá definice klíčových pojmů, jako je hrana nebo vrchol či neorientovaný nebo prostorový graf, v projektových činnostech jasně definovat cíle a úkoly.

3.2 Cesty na grafech

V rámci operací na grafech existuje z pohledu teorie grafů mnoho matematických aplikací, které lze využít při výpočtech na grafech. Příkladem může být výpočet minimální kostry grafu, výpočet cesty na grafech, výpočet nejspolehlivější cesty, či výpočet cesty s maximální kapacitou. V úvahu lze vzít další úlohy na grafech, což může být třeba obsluha vrcholů sítě. V rámci teoretické části se zaměříme pouze na určitou část optimalizačních úloh, kterými jsou výpočty cest na grafu.

Část teorie grafů, cesty na grafech, se pro nás stává velice zajímavou pro řešení nalezení nejkratší cesty v projektu vnitřní navigace. Cesty na grafech hledají optimální trasy v síti znázorněné schematickým modelem pomocí neorientovaného, souvislého, hranově ohodnoceného grafu, kde optimální trasa vyžaduje minimalizaci nákladů nutných pro její vykonání (10).

Cesty na grafech se dělí do několika kategorií:

- 1) Nejkratší cesty
- 2) Nejspolehlivější cesty
- 3) Cesty s maximální kapacitou

Úlohy spolehlivosti cest a cesty s maximální kapacitou v současném projektu nebudou řešeny. Poslední kategorií jsou vyhledávání nejkratších cest na grafech, kdy se daný algoritmus snaží nalézt nejkratší cestu na grafu z bodu A do bodu B, pokud možno v co nejkratším čase. Protože existuje mnoho různých algoritmů pro tento výpočet, které se používají pro výpočty v různých softwarech, jako je například Google Maps nebo jiné navigační aplikace, je vhodné si základní algoritmy přestavit níže v tabulce 7.

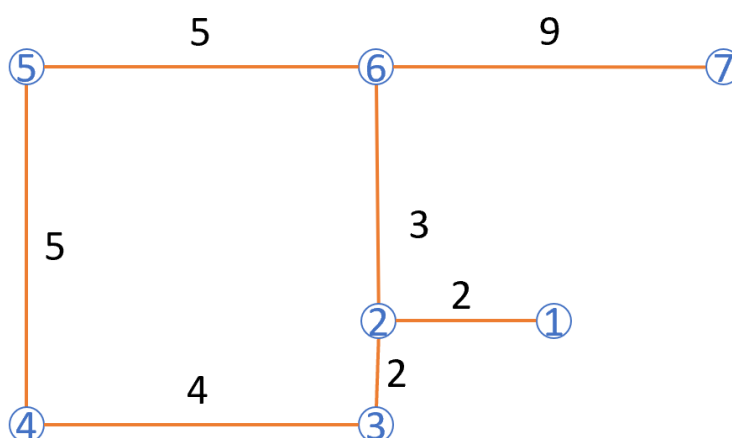
Dijkstrův algoritmus	Bellman-Fordův algoritmus
<p>Dijkstrův algoritmus se odlišuje od ostatních díky své schopnosti najít nejkratší cestu z jednoho uzlu ke každému jinému uzlu ve stejné grafové datové struktuře. To znamená, že namísto hledání nejkratší cesty od počátečního uzlu k jinému specifickému uzlu algoritmus pracuje na nalezení nejkratší cesty ke každému jednotlivému dosažitelnému uzlu – za předpokladu, že se graf nezmění.</p> <p>Algoritmus běží, dokud nejsou navštíveny všechny dostupné uzly. Proto byste Dijkstrův algoritmus museli spustit pouze jednou a uložit výsledky, abyste je mohli použít znovu a znovu, aniž byste museli algoritmus znovu spouštět, pokud se jakkoli nezměnila struktura dat grafu.</p>	<p>Podobně jako Dijkstrův algoritmus pracuje Bellman-Fordův algoritmus na nalezení nejkratší cesty mezi daným uzlem a všemi ostatními uzly v grafu. Ačkoli je pomalejší než předchozí, Bellman-Ford kompenzuje jeho nevýhodu svou všestranností. Na rozdíl od Dijkstrova algoritmu je Bellman-Ford schopen pracovat s grafy, ve kterých jsou některé váhy hran záporné.</p>
Floyd-Warshall algoritmus	Johnsonův algoritmus
<p>Floyd-Warshall vyniká tím, že na rozdíl od předchozích dvou algoritmů se nejedná o jednozdrojový algoritmus. To znamená, že vypočítá nejkratší vzdálenost mezi každou dvojicí uzlů v grafu, spíše než pouze počítání z jednoho uzlu. Funguje tak, že hlavní problém rozdělí na menší a poté zkombinuje odpovědi k vyřešení problému s hlavní nejkratší cestou.</p>	<p>Johnsonův algoritmus funguje nejlépe s řídkými grafy – grafy s méně hranami, protože jeho běh závisí na počtu hran. Takže čím méně hran, tím rychleji vygeneruje trasu.</p> <p>Tento algoritmus se liší od ostatních, protože při určování nejkratší cesty spoléhá na dva další algoritmy. Nejprve používá Bellman-Fordův algoritmus k detekci negativních cyklů a eliminaci jakýchkoli negativních hran. Pak se s tímto novým grafem spoléhá na Dijkstrův algoritmus pro výpočet nejkratších cest v původním grafu, který byl vložen.</p>

Tabulka 8 - Základní typy algoritmů pro hledání nejkratší cesty (11)

3.2.1 Dijkstrův algoritmus

I z definic jednotlivých algoritmů pro výpočet nejkratší cesty je patrné, že nejvhodnější bude využít Dijkstrův algoritmus pro vyhledání nejkratší cesty. Až si představíme projekt vnitřní navigace, tak ten bude založen na jednom souvislém grafu a vyhledávání cest z předem určeného místa. Definice Dijkstrova algoritmu z tabulky 7 říká, že algoritmus hledá nejkratší cestu z jednoho vrcholu ke každému jinému vrcholu ve stejné datové struktuře. To znamená, že pokud se graf nemění, algoritmus má již vypočtené výsledky uložené v 'mezipaměti', protože funguje na principu navštívení všech vrcholů. Tento předpoklad značně zlepšuje i výpočetní dobu algoritmu, a zvyšuje se tím i jeho rychlost při každém jednotlivém výpočtu. Pro jednoduchost je vytvořen výpočet Dijkstrova algoritmu na rovinném obyčejném grafu podle obrázku 6. Aby byl výpočet jednoduchý, je graf pro příklad algoritmu zjednodušen, ale odpovídá orientačně grafu z tabulky 6.

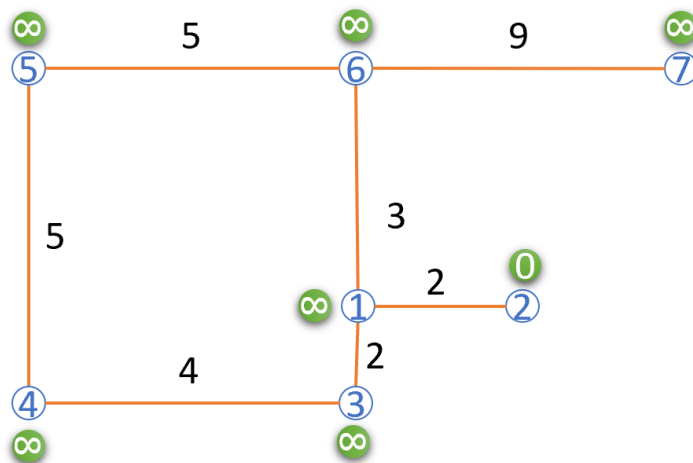
Výpočet:



Obrázek 6 - Obyčejný neorientovaný graf (vstup)

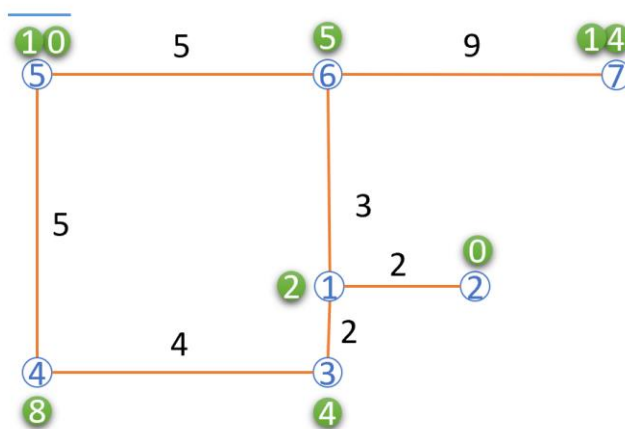
Postup výpočtu Dijkstrova algoritmu:

1. V grafu P na obrázku 7 zvolím počáteční bod k ($k = V_0$), kterým bude vrchol 1 a koncový vrchol cesty 7, značený ($v = v_n$).
2. Všem vrcholům v_i náležícím množině vrcholů V přiřadím hodnotu t_i , kde hodnota počátečního vrcholu je ($t_0 = 0$), a pro ostatní $t_j = \infty$. Hodnotami jsou čísla v zelených buňkách.



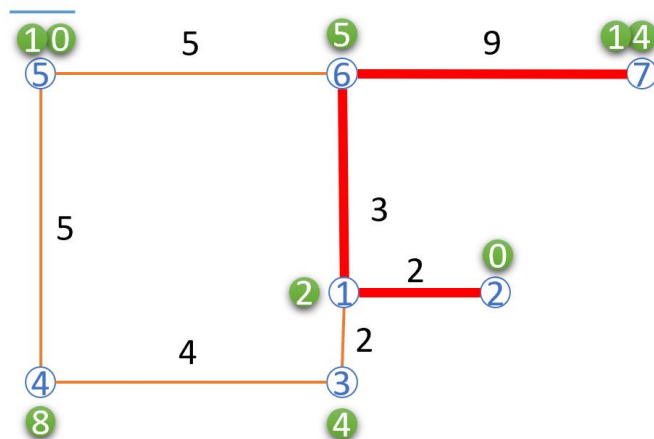
Obrázek 7 - Výpočet Dijkstrova algoritmu

3. V celém grafu P hledáme hranu, pro kterou platí: $t_j - t_i > o(v_i, v_j)$. Pokud taková hrana existuje, potom ohodnocení t_j nahradíme ohodnocením: $t_j' = t_j + o(v_i, v_j)$, což ukazuje další obrázek 6, hodnoty přiřazené vrcholům jsou v zelených buňkách.



Obrázek 8 - Postup výpočtu Dijkstrova algoritmu

Zpětný postup z vrcholu B (Vrchol 7) do vrcholu A (Vrchol 1) je vypočtený podmínkou $t_j - t_i = o(v_i, v_j)$.



Obrázek 9 - Výsledek výpočtu Dijkstrova algoritmu

Pomocí zpětného výpočtu lze jednoduše dopočítat výslednou nejkratší cestu na grafu, která je znázorněná na obrázku 9 červenou čarou. Algoritmus tedy systematicky postupuje od výchozího uzlu k cíli a aktualizuje vzdálenosti. Výstupem algoritmu je pak délka nejkratší cesty z výchozího uzlu do všech ostatních uzlů (12).

3.3 Softwarová podpora výpočtu algoritmů

Výše zmíněné algoritmy je možné v současné době počítat s využitím počítačových technologií. Výpočet algoritmů lze naprogramovat jedním z objektově orientovaných jazyků, jako je například programovací jazyk C++. Možností volby programovacího jazyka je mnoho. K psaní kódu může uživatel zvolit například programovací jazyk Python. Ovšem volba programovacího jazyka může značně ovlivnit rychlost spuštění / rychlost exekuce kódu. V případě, kdy je předmětem kódu výpočet algoritmu pro hledání nejkratší cesty, je nutné, aby výpočet pobíhal co nejrychleji. Důvodem požadavku na rychlost výpočtu algoritmu je nutnost počítat s rostoucím počtem hran na grafech, které jsou předmětem výpočtu algoritmu.

Hlavním úkolem kapitoly Softwarová podpora výpočtu algoritmů je představit volně dostupný software pgRouting, který nabízí sadu základních algoritmů pro výpočet na grafech. Jeho velkou výhodou je, že je aplikovatelný do databázového prostředí PostgreSQL, jako přídatný modul. Vkládání vstupních dat do počítačového kódu vytvořeného v objektově orientovaných jazycích je složitý proces. Mnohdy je potřeba data vkládat pouze číselně pomocí vstupních matic grafů a jiné způsoby mohou být již složité a vyžadují pokročilé programování. V druhé řadě problém těchto postupů je náročný pro ty, kteří neovládají objektově orientované programování. Z vyjmenovaných důvodů se práce zaměřuje na již

vyvinuté řešení, kterým je software pgRouting. Jedná se o software jednoduchý na obsluhu, který disponuje velkou knihovnou funkcí, které při správně vytvořených vstupech uložených v databázi, dokáží efektivně počítat s grafy.

3.3.1 PgRouting

PgRouting je doplňkový software pro databázi PostgreSQL, který slouží k vývoji a využívání navigačních a síťových analýz. PgRouting vychází z modulu PostGIS, který je delší dobu součástí databáze PostgreSQL. PostGIS je důležitým doplňkem, který dokáže ukládat prostorová data a dále je zpracovávat. Ačkoliv PostGIS disponuje mnoha různými funkcemi jak pro analýzu na grafech, tak pro síťové analýzy, v žádném případě nedokáže vypočítat v optimálním čase nejkratší cestu mezi bodem A a B. Dokáže jej pouze geometricky analyzovat.

Software pgRouting je implementací problému nejkratší cesty a jiných síťových analýz na grafu. Využitím pgRoutingu, se uživatel vyhýbá složitému programování a náročnému propojování při řešení problému, kterým může být například nalezení nejkratší cesty. Software pgRouting se prezentuje jako balíček funkcí, které stačí spustit k obdržení výsledku, který lze přenést dále (13).

3.3.2 Historie

Software pgRouting je v současnosti v třetí verzi 3+ a poslední stabilní verze funguje na verzi 3. 2., přičemž nižší verze již nemusejí být kompatibilní s kmenovým softwarem. Příkladem může být dodržení kompatibility verzí, např. stará verze softwaru pgRouting 2.3 bude fungovat pouze s modulem PostGIS 2.3 a s databází PostgreSQL 9,5+. Kompatibilitu verzí je nutné dodržet, protože použití neaktuálních verzí nebo odlišných verzí by mohlo v mnoha projektech způsobit velké vývojové problémy, a to i v rámci budoucího využití.

V rámci našeho projektu budeme využívat nejnovější verze všech tří komponent (PostgreSQL, PostGIS, pgRouting). Manuály nebo knihy k volně dostupným softwarům jsou mnohdy zastaralé a konečný uživatel by měl vždy dbát na aktuálnost a komptabilitu těchto tří komponent.



Obrázek 10 – Kompatibilita verzí modulů a softwarů

3.3.3 Funkce pgRoutingu

Software pgRouting je vyvíjen na bázi opensource, což znamená, že vývoj probíhá vytvářením softwaru za podpory jednotlivců nebo organizací. Z toho důvodu software nedisponuje stovkami funkcí jako jiné placené softwary pro síťovou analýzu. V současnosti nabízí jedenáct různých skupin funkcí síťové analýzy.

Skupiny síťových analýz pgRouting (14):

- All Pairs Shortest Path, Johnson's Algorithm
- All Pairs Shortest Path, Floyd-Warshall Algorithm
- Shortest Path A*
- Bi-directional Dijkstra Shortest Path
- Bi-directional A* Shortest Path
- Shortest Path Dijkstra
- Driving Distance
- K-Shortest Path, Multiple Alternative Paths
- K-Dijkstra, One to Many Shortest Path
- Traveling Sales Person
- Turn Restriction Shortest Path (TRSP)

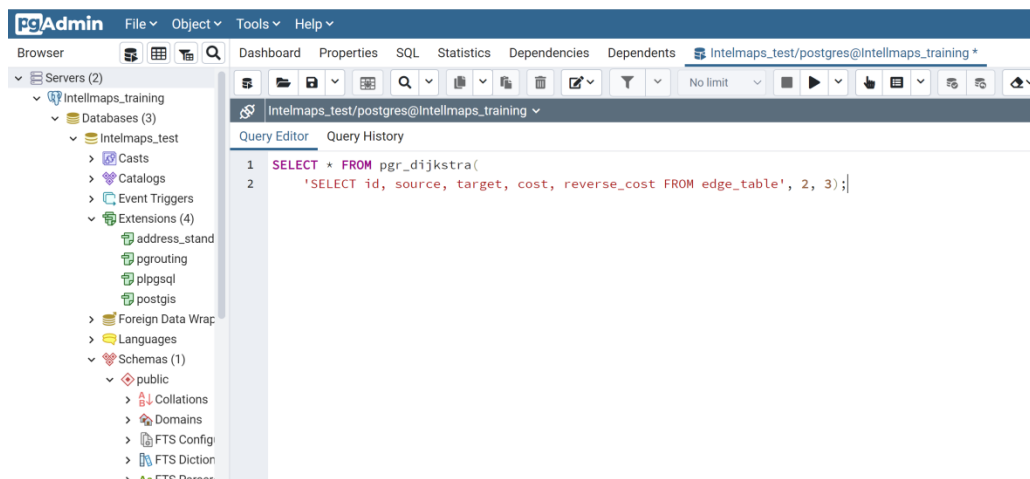
V jednotlivých skupinách knihovny softwaru pgRouting lze nalézt i tři ze čtyř základních algoritmů pro vyhledávání nejkratší cesty - Johnsonův algoritmus, Floyd-Warshallův algoritmus a Dijkstrův algoritmus.

Zajímavé je se také zaměřit na skupinu síťových analýz Travel Sales Person neboli problém obchodního cestujícího, který k řešení nabízí knihovna softwaru pgRouting. Problém obchodního cestujícího obecně řeší problém navštívení všech vrcholů v grafu s minimalizací dopravní práce.

V rámci práce se zaměřím hlavně na Dijkstrův algoritmus, který software pgRouting umí vypočítat. Z historického hlediska jde o nejspolehlivější algoritmus v této knihovně, protože v rámci vývoje byl navržen mezi prvními, což dokládá i historický název softwaru pgDijkstra, který byl později změněn na pgRouting.

3.3.4 Výpočet Dijkstrova algoritmu v pgRouting

Fungování algoritmu je založeno na psaní SQL kódu do konzole databáze PostgreSQL, kde je příkaz k výpočtu spouštěn a kde se i ukazují výsledky výpočtu. Výpočet funkce je založen na volání funkce `pgr_dijkstra`, která celý výpočet provede. Výpočtová funkce potřebuje vstupní parametry. V případě funkce `pgr_dijkstra` je vstupním parametrem obyčejný neorientovaný graf:



Obrázek 11 - Prostředí výpočtu pgRouting v PostgreSQL

Skript volání funkce z obrázku 11 postupně identifikuji:

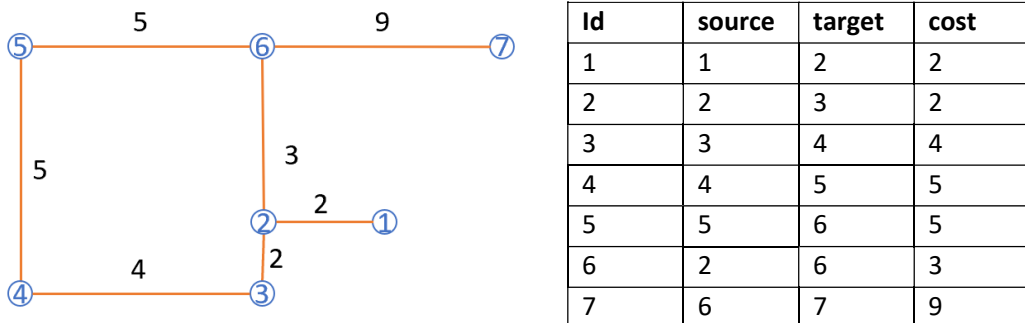
```
SELECT * FROM pgr_dijkstra(  
    'SELECT id, source, target, cost, cost as reverse_cost FROM priklad_tabulk  
a',  
    1, 7  
);
```

Funkce `SELECT * FROM` je klasická syntaxe jazyka SQL, která nám říká: Vyber všechny sloupce a jejich hodnoty z (tabulky).

`Pgr_dijkstra` je samotná funkce, kterou voláme do knihovny `pgRouting`. V kombinaci s první funkcí "*" nám funkce říká: „Vyber všechny hodnoty, které vrátí funkce `pgr_dijkstra`“. V případě této funkce se jedná o sloupce (`seq`, `path_seq`, `node`, `edge`, `cost`, `agg_cost`). Popis, co jednotlivé sloupce znamenají, bude uveden v posledním kroku kapitoly u výsledku výpočtu.

Druhý řádek skriptu je vnořená funkce, která říká: „Vyber sloupce se jménem `id`, `target`, `cost`, `reverse_cost` z tabulky `priklad_table`“. To znamená, že tento řádek vybere vstupy z tabulky, která reprezentuje graf pro funkci `pgr_dijkstra`, a ta vypočte nejkratší cestu z tabulky `priklad_table`.

Abychom pochopili samotný vstup do funkce, využijeme příklad z kapitoly 3.2.1, kde příklad neorientovaného obyčejného grafu převedeme do strukturované tabulky (příklad_table), pro výpočet funkce.



Obrázek 12 - Převedení grafu do strukturované tabulky (příklad_table)

Na obrázku 12 vidíme, jak lze prezentovat neorientovaný graf v SQL tabulce. Důležité je graf zapsat správně a dávat si pozor při zapisování vrcholů, kdy při přepsání či změně dojde k rozkladu grafu. Z naformátované tabulky si dokáže skript SQL funkce vzít potřebná data pro výpočet *pgr_dijkstra* funkce. Důležité je dodržet pojmenování sloupců a jejich datový typ:

- Id – ANY-INTEGER
- Source – ANY-INTEGER
- Target – ANY-INTEGER
- Cost – ANY-NUMERICAL

Tabulka v tomto příkladu je zjednodušena pro vysvětlení, jak funguje SQL skript. Ve vývojové části tabulka obsahuje i sloupec *the_geom*, kterou vytváří PostGIS doplněk a která reprezentuje fyzicky nakreslenou čáru, aby výsledek mohl být graficky vykreslen.

V příkladu našeho SQL skriptu voláme požadavek o nalezení nejkratší cesty z vrcholu 1 do vrcholu 7 na grafu z obrázku 12.

seq	path_seq	node	edge	cost	agg_cost
1	1	1	1	2	0
2	2	2	6	3	2
3	3	6	7	9	5
4	4	7	-1	0	14

Výsledkem je tabulka s šesti sloupci, které nám popisují sekvenci nalezené cesty. Vysvětlení sloupců je následující:

- Seq – sekvence vrcholů tzv. sloupce node
- Path_seq – sekvence hran tzv. sloupec edge
- Node – Vrcholy, jak jdou po sobě z vrcholu 1 do vrcholu 7.
- Edge – Hrany, jak jdou po sobě, označeny podle id sloupce z tabulky prikklad_table.
- Cost – ohodnocení hrany
- Agg_cost – suma ohodnocení hran, čtvrtý řádek celková délka trasy z vrcholu 1 do vrcholu 7

Následujícím způsobem je možné provádět výpočty z libovolného vrcholu do libovolného cílového vrcholu. Funkce pgr_dijkstra nabízí tři způsoby výpočtů nejkratší cesty.

- Z jednoho vrcholu do jednoho vrcholu
- Z jednoho vrcholu do více vrcholů
- Z více vrcholů do jednoho vrcholu
- Z více vrcholů do více vrcholů

Využití softwaru pgRoutingu je jednoduché, pouze vyžaduje dobrou znalost teorie grafů a způsobů, jak převést daný graf do atributové tabulky, kterou lze dále zpracovat v databázi SQL. Převedení grafu do SQL atributové tabulky může mít velkou výhodu v možnosti párování dat pomocí SQL propojení. Zmíněným způsobem lze jednotlivým vrcholům nebo hranám přidělit i další dílčí informace nezávisle na grafu.

4 Projekt vnitřní navigace

V rámci své diplomové práce jsem byl společností Intellmaps s.r.o. osloven s nabídkou k vytvoření vnitřní navigace pro uzavřený objekt. Pojmem vnitřní navigace v rámci projektu rozumějme vytvoření aplikace pro navigování klientů v objektu Polikliniky Prosek pomocí pevně zabudovaného navigačního kiosku, z něhož bude možné vyhledávat nejkratší trasy k lékařům v objektu.

Bylo nutné vzít v úvahu, že společností byl projeven zájem na vytvoření vnitřní navigace, do které bude možné integrovat i její budoucí vyšší verze nebo propojení s ostatními systémy, jako je například lokace uživatele v budově pomocí bezdrátových snímačů nebo integrace navigace do třírozměrných pohledů. Cílem tohoto projektu nebylo řešit propojení s výše zmíněnými systémy, ale je nutné tyto požadavky na architekturu a logiku aplikace zahrnout již do její první verze. Požadavek na možnost integrace vyšších budoucích verzí musel být zahrnut do plánu a zvyšoval časovou náročnost vytvoření cílového produktu vnitřní navigace. Ucelené řešení ovšem umožní jednoduchou integraci dalších systémů a technicky dostupnější vypracování aplikace vnitřní navigace i pro další budoucí objekty.

Efektivní zpracování projektu vyžadovalo, aby byl projekt rozdělen do čtyř fází. Projekt prochází scénářem všech čtyř fází, aby bylo zřejmé, jak byl produkt vnitřní aplikace vyvíjen a jak projekt postupoval v čase.

První fáze: Zahájení projektu
Druhá fáze: Plánování projektu
Třetí fáze: Realizace projektu
Čtvrtá fáze: Ukončení projektu

Tabulka 9 - Fáze projektu

Dále je projekt rozdělen do jednotlivých etap, které rozdělují projekt a znázorňují prototypy jednotlivých verzí vnitřní navigace v jednotlivých fázích projektu. Fáze rozdělují projekt z časového pohledu a etapy rozdělují projekt podle jednotlivých verzí.

Popis jednotlivých etap:

První etapa:

První etapa obsahuje zpracování záměrů společnosti Intellmaps, projednání jednotlivých aspektů a analýzu vstupních dat, diskusi nad zpracováním a použitím vnitřní navigace. Do této etapy byla zahrnuta i schůzka se správcem budovy Polikliniky Prosek, pro přesné vydefinování cílových bodů zájmu. Jako příklad lze uvést seznam lékařských pracovníků pro určení cíle navigování nebo požadavek na určení částí objektu, které do navigace zahrnuty být nemají.

Druhá etapa:

Druhá etapa obsahuje vývoj a vytvoření první beta verze vnitřní navigace. Etapa zahrnuje i vytvoření základní logické struktury vnitřní navigace, kterou lze vytvořit dle technických požadavků na základě definice z první fáze projektu. Beta verze by měla zajistit otestování funkčnosti, spolehlivosti, ale i rychlosti použitého řešení. Beta verze je důležitý bod projektu z pohledu fungování aplikace, její integrace do rozdílných prostředí a možných budoucích úprav. Beta verze vnitřní navigace se zaměřuje primárně pouze na funkčnost a představuje podporu pro následující etapy.

Třetí etapa:

Třetí etapa obsahuje vytvoření první verze vnitřní navigace verze 1.0, která představuje implementaci beta verze navigace do webového prostředí společnosti Intellmaps. Cílem této etapy je tedy přenést beta verzi aplikace vnitřní navigace v plné funkčnosti do nového prostředí. Verze 1.0 počítá i s vytvořením základního uživatelského rozhraní pro obsluhu vnitřní navigace. Výsledkem má být plně funkční webová aplikace vnitřní navigace verze 1.0, kterou již bude možné uživatelsky obsluhovat.

Čtvrtá etapa:

Čtvrtá etapa počítá s představením vize pro budoucí vývoj verzí vnitřní navigace 2.0 a 3.0. V této etapě je představeno optimální směřování aplikace, její možná vylepšení a možnosti propojení s jinými systémy nebo zařízeními, které dokáží funkcionalitu vnitřní navigace vylepšit a konečnému uživateli dopřát větší komfort při jejím využívání.



Obrázek 13 - Stupně etap a fází projektu

Jednotlivé verze vnitřní navigace jsou rozděleny do jednotlivých etap, aby byl znázorněn vývoj a prototypování jednotlivých verzí vnitřní navigace. Etapa projektu, či spíše etapa realizace projektu, znamená realizaci určité části fáze projektu, specifickou vzhledem k obsahu projektu a jeho konkrétnímu pojetí (15). Aby byl projekt přehledný a hlavně vývojový postup etap detailnější, je kladen velký důraz na zařazení jednotlivých etap do jednotlivých fází projektu. V obvyklém pojetí mívá projekt čtyři fáze řízení: zahájení, plánování, realizace a ukončení. Pojem „fáze projektu“ je rozdílný od pojmu „etapa projektu“, jejíž význam je důležitý z hlediska obsahu konkrétního projektu. Fáze řízení projektu jsou pro všechny projekty stejné a univerzálně platné (16). Například na obrázku 13 je vyznačeno pomocí barevného rozlišení, do jaké fáze je jednotlivá etapa projektu zařazena, a věnuje se jí nejvíce. Aby byla orientace v projektu usnadněna, byla vytvořena tabulka 9, která obsahuje rozlišení jednotlivých fází projektu různými barvami. Další části projektu budou odpovídat těmto fázím.

První fáze: Zahájení Projektu	(Zelená barva)
Druhá fáze: Plánování Projektu	(Oražová barva)
Třetí fáze projektu: Realizace Projektu	(Modrá barva)
Čtvrtá fáze projektu: Ukončení projektu:	(Žlutá barva)

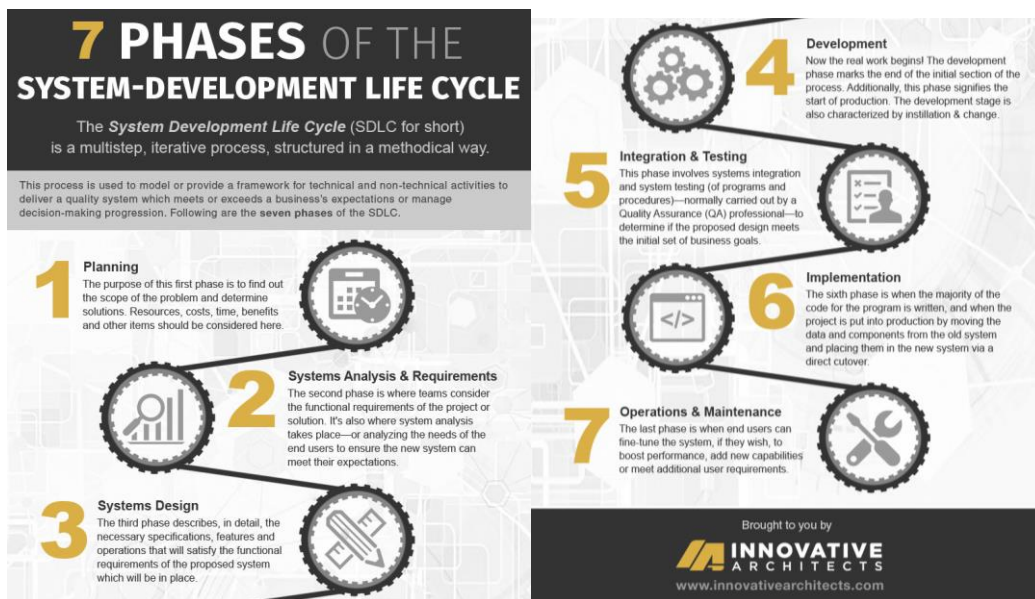
Tabulka 10 - Kategorizace fází projektu

Stejné značení bude použito i v dalších kapitolách, například při vytváření Ganttova diagramu nebo později ve fázi realizace, v níž budou jednotlivé etapy do detailu popisovány pomocí SDLC cyklů.

4.1 SDLC

SDLC neboli softwarový vývojový životní cyklus je proces, který produkuje software s nejvyšší kvalitou a nejnižšími náklady v co nejkratším čase. SDLC obsahuje podrobný plán pro vývoj, změnu, udržování a nahrazení softwarového systému (17). Při vytváření aplikace vnitřní navigace byly použity stejné postupy jako pro vývoj velkých firemních softwarů pomocí metody SDLC, která zahrnuje šest kroků cyklu, podle kterého se postupovalo i v případě naší aplikace vnitřní navigace:

- 1) Plánování
- 2) Analýza
- 3) Design
- 4) Vývoj
- 5) Testování
- 6) Implementace
- 7) Údržba



Obrázek 14 - SDLC diagram (18)

SDLC metod, které se využívají k vývoji je více, a každá SDLC metoda přistupuje k vývoji aplikace nebo softwaru jiným způsobem.

Druhy metod SDLC:

1. Vodopád

- Zahnuje pevnou strukturu, tak aby byly všechny systémové požadavky definovány na samém začátku projektu. Teprve poté mohou začít fáze návrhu a vývoje. Taková metoda funguje, když software je definován již na začátku a nepotřebuje nutné úpravy (18)

2. Iterativní

- Iterativní metodologie vezme model vodopádu a několikrát jej projde v malých krocích. Spíše než procházet celý projekt napříč fázemi SDLC je každý krok přeměněn na několik mini projektů, které mohou při vývoji produktu přinášet přidanou hodnotu (18).

3. DevOps

- DevOps je jednou z nejnovějších metodologií SDLC a je přijímána mnoha softwarovými společnostmi a IT organizacemi. Jak název napovídá, předpokladem DevOps je spojit vývojové týmy s operačními týmy za účelem zefektivnění poskytování a podpory (18).

4. V-Model

- Evoluce klasické vodopádové metodologie, v-model SDLC procesní kroky jsou otočeny nahoru po fázi kódování. V-model má velmi striktní přístup, přičemž další fáze začíná až po dokončení předchozí fáze (18).

5. Spirála

- Spirálová metodika umožňuje týmům převzít více SDLC modelů na základě rizikových vzorců daného projektu kombinací iterativních a vodopádových přístupů. Výzvou pro spirálový model je určit ten správný okamžik, kdy přejít do další fáze (18).

6. Lean

- Pokud jde o software a projekty, „lean“ metodika SDLC se zaměřuje na snižování plýtvání v každé fázi, včetně plánování, nákladů a rozsahu. Tento přístup je nejpřesvědčivější pro organizace s přísnými hardwarovými požadavky a dalšími potřebami nákupu (18).

7. Agilní

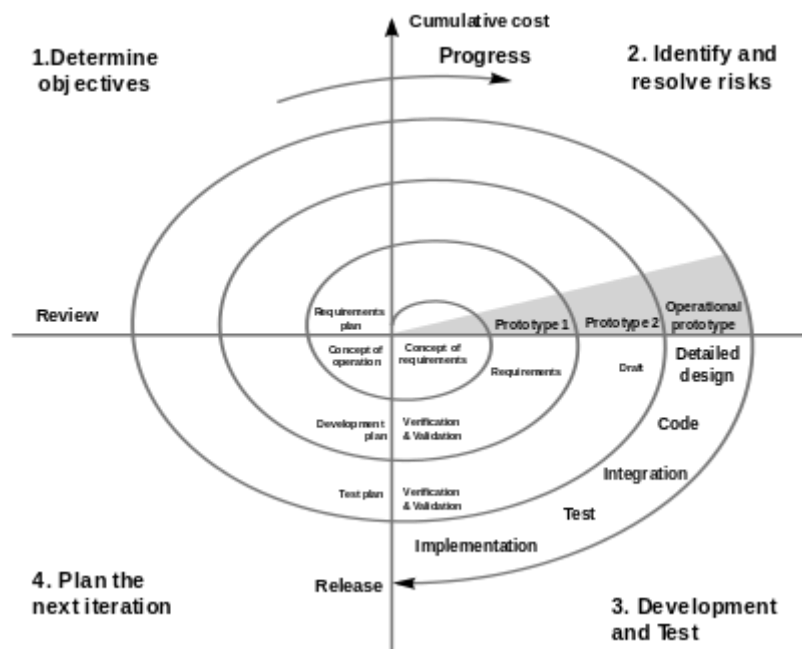
- Agilní metodika je opakem vodopádového přístupu. Namísto toho, aby požadavky, návrh a testování byly považovány za velké sekvenční kroky, agilní model z nich dělá probíhající procesy, které vyžadují zapojení vývojářů, managementu a zákazníků (18).

8. Prototypování

- V metodologii prototypování se designérský tým zaměřuje na vytvoření raného modelu nového systému, softwaru nebo aplikace. Tento prototyp nebude mít plnou funkčnost ani nebude důkladně testován, ale dá externím zákazníkům představu o tom, co přijde. Poté lze shromáždit zpětnou vazbu a implementovat ji ve zbývajících fázích SDLC (18).

4.1.1 Spirálová metoda

Po prozkoumání všech dostupných metod byla pro vývoj vnitřní navigace zvolena spirálová metoda (viz výše Spirála). Jedná se o nejpřesnější metodu pro vytvoření aplikace, kterou je vnitřní navigace. Metoda kombinuje vodopádový i iterativní přístup, ale zároveň dokáže pracovat s implementací nových nebo nečekaných požadavků a dokáže vstříbat střední až velké úpravy projektu. Zároveň tato metoda vývoje postupných prototypů dokáže eliminovat riziko spojené s využitím nevhodné technologie, ztrátou času nebo nedostatkem finančních prostředků.



Obrázek 15 - Grafické znázornění spirálové metody

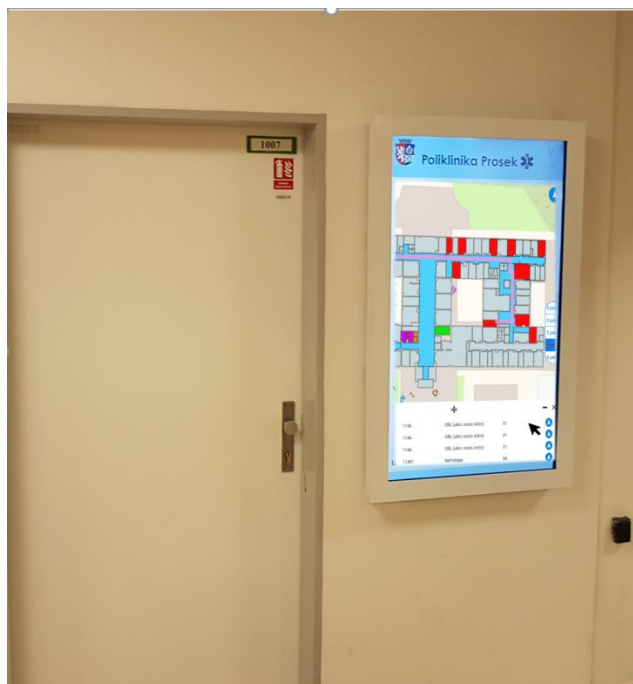
V projektu budeme moci demonstrovat použití spirálové metody. Podle obrázku 15 lze výsledek etapy dva, tři i čtyři znázornit jako jednotlivé křivky spirály, kdy každý prototyp, v našem případě jednotlivá verze vnitřní navigace, je podkladem další verze. S jistotou lze dodat, že nejsou přesně naplněny postupy, které jsou znázorněny na schématu spirálové metody, ale značně se k nim přibližují a snaží se je splňovat v dílčích krocích.

4.2 Zahájení projektu

Projekt vnitřní navigace byl zahájen v listopadu roku 2020 po domluvě s jednatelem a zakladatelem mladé technologické společnosti Intellmaps s.r.o., Ing. Ladislav Čapkem. Požadavky na vnitřní navigaci měly různé specifikace a musely splňovat funkční, systémová, ale i uživatelská kritéria.

4.2.1 Hlavní cíl projektu vnitřní navigace:

Hlavním cílem projektu bylo vytvořit vnitřní navigaci pro veřejný zdravotnický objekt Poliklinika Prosek formou webové aplikace, která bude mít za úkol navigovat uživatele nejkratší trasou od pevného startovního bodu do cílového bodu zájmu. Projekt je primárně zaměřen na vytvoření navigace pro pevně umístěný počítač u vchodu objektu (Info-kiosek).



Obrázek 16 - Cíl projektu vnitřní navigace

Na obrázku 16 je vizualizován cílový stav vnitřní navigace, který má za úkol uživateli na monitoru pevně umístěného počítače kiosku ukázat nejkratší cestu do cílových bodů v pěti různých poschodích v jednoduchém 2D pohledu.

Když je vydefinován výsledný produkt, je nutné již v této fázi projektu přesně zaznamenat požadavky, určit jejich kategorii a prioritu. Přesně vydefinované požadavky jsou základem každého projektu a jejich přesná znalost nám pomůže nejen v druhé fázi projektu, kterou je plánování, ale i se sestavením časového harmonogramu a rozdělením úloh jednotlivým členům, kteří budou na projektu participovat.

Na architektuře systému lze demonstrovat nutnost vstupních požadavků, kdy jeden z požadavků již počítá se zařazením vnitřní navigace do budoucích projektů, jako je například určování polohy pomocí bezdrátových senzorů ve vnitřních prostorách budovy. Architektura vytvoření cesty v tomto případě musí být navržena dynamicky a musí již umět vytvářet navigovanou cestu, pokud možno od jakéhokoliv vrcholu uvnitř budovy.

Jednotlivé požadavky na vnitřní navigaci jsou uvedeny v tabulce 10.

Požadavek	Detailní popis požadavku	Priorita	Zařazeno do současného projektu	Využití v budoucích projektech
1. Vytvořit vnitřní navigaci pro webový prohlížeč	Cílem požadavku je vytvořit vnitřní navigaci, která bude fungovat na webovém prohlížeči, nikoliv jako desktopová aplikace nebo jako aplikace stažitelná v obchodech s aplikacemi Apple Store nebo Google play pro smartphones. Aplikace počítá v budoucích projektech s funkcí i pro mobilní zařízení, ale pouze ve webovém prohlížeči díky responsivnímu řešení použitých technologií.	Vysoká	Ano	Ano
2. Použít dostupné technologie / opensource řešení	V 99 % případů využít dostupných řešení tzv. open-source nástrojů nebo knihoven, které jsou dostupné bezplatně na internetu. Příkladem může být software QGIS nebo databáze PostgreSQL. Pro webové řešení pro design stránky například framework Quasar.js nebo Vue.js.	Vysoká	Ano	Ano
3. Použít mapové technologie k navádění	Tento požadavek vyplývá z hluboké propojenosti navigování v mapě a vysoké dostupnosti řešení pro mapové podklady, i z faktu, že vstupní podklady již byly zpracovány ve formátech Shapefile, které se nejlépe kombinují s mapovými technologiemi.	Vysoká	Ano	Ano
4. Navádění zákazníků z předem určeného vrcholu (Info-kiosek)	Navádění pacientů od předem určeného místa na grafu k bodům zájmu. Požadavek počítal původně s manuálním vytvořením jednotlivých čar do každého bodu zájmu. Po vznesení požadavku 5 bylo dohodnuto, že navigování od Info-kiosku bude možné, ale navigovatelné čáry budou muset být vytvořeny alternativním způsobem.	Vysoká	Ano	Ne
5. Navádět z různých vrcholů na grafu vnitřní navigace	Tento požadavek již počítá s integrací vnitřní navigace do budoucího projektu, kde se bude uživatel objevovat na mobilním zařízení jeho aktuální poloha a od této polohy bude navigován do bodu zájmu.	Vysoká	Ne	Ano

6. Navádět po schodech nebo výtahem	Požadavek má dát možnost volby uživatelům vnitřní navigace zvolit si, jestli jejich cesta povede do vyšších pater či patra po schodech nebo výtahem. Požadavek má za cíl snížit vytížení výtahů.	Nízká	Ne	Ano
7. Navádět klienty po budově k lékařům	Požadavek navádět pouze k lékařům polikliniky vznesla společnost Poliklinika Prosek. V objektu se nachází i jiné organizace, ale hlavní prioritou je navigovat k odborným a praktickým lékařům	Střední	Ano	Ano
8. Navádění na více míst najednou	Požadavek na navádění na více míst najednou je zahrnut pro budoucí modifikace vnitřní navigace. Vnitřní navigace zajistí funkcionalitu navigovat uživatele na více míst najednou.	Nízká	Ne	Ano
9. Navádění v režimu outdoor / indoor	Na jednoduchém příkladu zařadit do navigace možnost navádění na venkovní místo. Například navigovat nejkratší trasu od Info-kiosku na venkovní parkoviště. Možnost duálního navádění ukazuje na komplexnost aplikace.	Střední	Ano	Ano

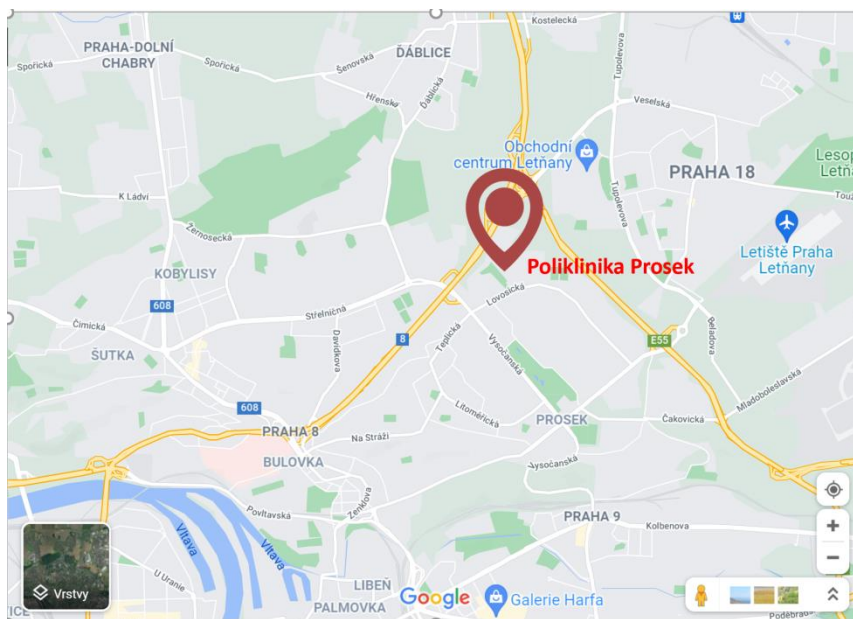
Tabulka 11 - Požadavky projektu vnitřní navigace

4.2.2 Objekt vnitřní navigace

Již vstupní požadavky projektu ukázaly na mnoho pohledů a funkcionalit, které bude muset vnitřní navigace splňovat, nebo aspoň svojí logikou a architekturou být plně připravena na to, že v budoucích projektech budou tyto funkcionality využity, například logiku navádění. V rámci první fáze projektu se seznámíme s objektem Polikliniky Prosek, pro který bude projekt vnitřní navigace vytvořen.

4.2.3 Poliklinika Prosek – Objekt vnitřní navigace

Zdravotnické zařízení Polikliniky Prosek a.s. sídlí na adrese Lovosická 440, 190 00 Praha 9, Česká republika. Objekt se nachází na severu Prahy u Městského okruhu u sjezdu na Vysočany. Zdravotnické zařízení Poliklinika Prosek je největší zdravotnické zařízení na Praze 9 a jedno z největších v hlavním městě Praze; je zde poskytována ambulantní zdravotní péče ve všech základních i specializovaných lékařských oborech. Současně zabezpečuje i provoz lůžkového oddělení následné péče a provoz lékárenského zařízení.



Obrázek 17 - Poloha Polikliniky Prosek

Současně zabezpečuje i provoz lůžkového oddělení, oddělení následné péče a provoz lékárenského zařízení. Zdravotní péče je poskytována jak provozovatelem zdravotnického zařízení Poliklinika Prosek a.s., tak i dalšími soukromými nestátními zdravotnickými zařízeními, a to občanům a návštěvníkům Prahy 9 a přilehlých spádových obvodů včetně okresu Praha - východ (19).



Obrázek 18 - Poliklinika Prosek

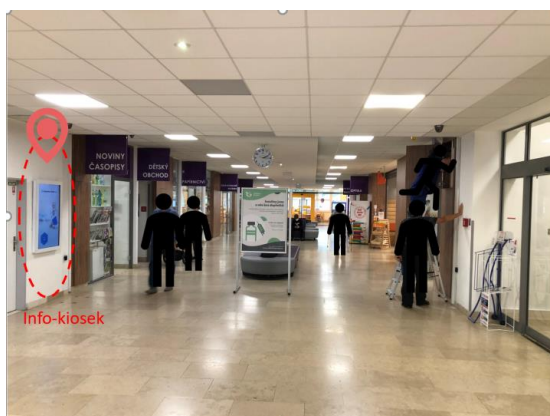
0. podlaží:

První podzemní podlaží označené pro účely projektu jako 0. podlaží svojí vybaveností slouží hlavně jako zázemí pro zdravotnické firmy, nachází se zde sportovní místnosti a pouze pět lékařských ordinací, které jsou aktivně využívány. Podlaží 0 nemá pro tento projekt tak velký význam.

S detailnějším zmapováním podlaží se počítá v budoucích projektech, kdy se bude například vytvářet navigace i pro technické pracovníky a údržbáře.

1. nadzemní podlaží:

První nadzemní podlaží (přízemí), je místo, kde se pacient ocitne při vstupu do budovy Polikliniky Prosek. Nachází se v něm hlavně administrativní zázemí polikliniky, soukromé subjekty (občerstvení, trafika, lékárna) a pouhých 15 lékařských ordinací. Patro zahrnuje i vstup do lůžkového přístavku. V prvním nadzemním podlaží se bude hned u vchodu nacházet Info-kiosek, který bude umístěn na viditelném místě na zdi - viz obrázek 18.



Obrázek 19 - 1. patro s polohou kiosku

2. nadzemní podlaží:

V druhém nadzemním podlaží (1. patře) se nachází převážná část všech ordinujících lékařů. Je zde celkem 36 lékařských ordinací praktických lékařů nebo specializovaných oddělení odborných lékařů. Druhé nadzemní podlaží je, co do obsazenosti, ze všech podlaží nejvíce využitě. V druhém nadzemním podlaží se předpokládá nejvyšší počet bodů zájmů k vyhledávání.

3. nadzemní podlaží:

Třetí nadzemní podlaží (2. patro) je již svými fyzickými rozměry jednodušší i pro orientaci pacienta / klienta. Na třetím nadzemním podlaží se nacházejí pouze specializovaná oddělení (Cardio, ORL, EMG, Psychiatrie, Neurologie, Gastro)

4. nadzemní podlaží:

Čtvrté nadzemní podlaží (3. patro) je fyzickými rozměry stejné jako třetí nadzemní podlaží. Ve čtvrtém nadzemním podlaží se nachází pouze zubní lékařské ordinace a psychiatrická ordinace.

4.2.4 Hlavní body projektu

Již na začátku projektu po vydefinování požadavků bylo nutné vytvoření a schválení předběžného programu hlavních činností, které bude nutné vykonat pro plynulý průběh projektu. Vytvoření hlavních bodů činnosti v projektu muselo být předmětem plánovací fáze projektu. Jelikož projekt vnitřní navigace přinášel mnoho neznámých, tak je pouze orientační a detekuje pouze hlavní body práce na projektu. Díky takto sestavenému seznamu bodů je možné dále sestavit Ganttův diagram a určit časový odhad jednotlivých úloh a určit přibližný časový odhad projektu.

Legenda k tabulce činností:

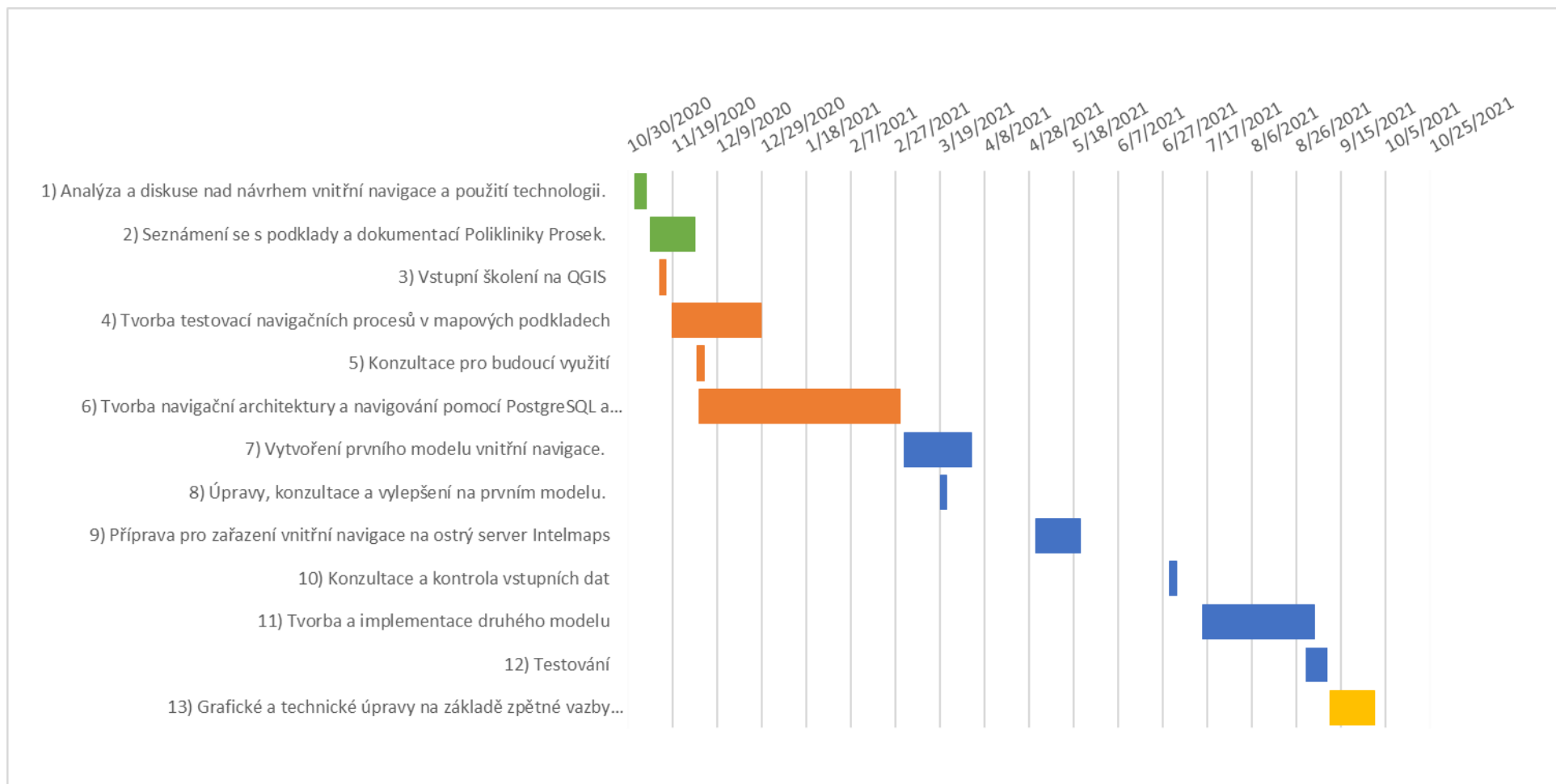
- MŠ: Martin Štourač (Student Fakulty dopravní ČVUT)
- LČ: Ing. Ladislav Čapek, MBA (Jednatel společnosti Intellmaps)
- JJ: Jan Jandourek (Project leader, společnost Intellmaps)
- PS: Pavel Srovnal (GIS Specialista, společnost Intellmaps)
- RI: Ing. Radoslav Irha (Software developer, společnost Intellmaps)
- RF: Robert Fischer (Vedoucí provozně-technického úseku, Poliklinika Prosek)

ID činnosti	Činnost	Účast na činnosti
1.	Analýza a diskuse nad návrhem vnitřní navigace a použití technologií	MŠ, JJ, LČ
2.	Seznámení se s podklady a dokumentací Polikliniky Prosek	MŠ
3.	Vstupní školení na QGIS a na práci s mapovými podklady včetně mapového podkladu Polikliniky Prosek	MŠ, PS
4.	Tvorba testovacích navigačních procesů v mapových podkladech	MŠ
5.	Konzultace pro budoucí využití	MŠ, LČ, RI
6.	Tvorba navigační architektury a navigování pomocí PostgreSQL a pgRouting	MŠ
7.	Vytvoření prvního modelu vnitřní navigace	MŠ
8.	Úpravy, konzultace a vylepšení na prvním modelu	MŠ, LČ, RI
9.	Příprava pro zařazení vnitřní navigace na ostrý server společnosti Intellmaps	MŠ, RI
10.	Konzultace a kontrola vstupních dat	MŠ, LČ, RF
11.	Tvorba a implementace druhého modelu	MŠ, RI
12.	Testování	MŠ
13.	Grafické a technické úpravy na základě zpětné vazby z testovací části	MŠ

Tabulka 12 - Plán činností na projektu

Plán projektu je pouze předběžný a je časově nadhodnocený; nadhodnocení je běžná praktika v projektovém řízení. Ve vývojovém projektu, jako je vnitřní navigace, není jednoduché odhadnout náročnost jednotlivých úkolů, zejména technicky vývojových úkolů, jako je například práce s PostgreSQL a pgRoutingem. Jednotlivé úkoly z tabulky 11 jsou přiřazeny svým nositelům a jsou vypsány ve vytvořeném Ganttově diagramu.

Ganttův diagram graficky zobrazuje časovou náročnost projektu vnitřní navigace. Ganttův diagram zobrazuje i jednotlivé fáze přiřazené jednotlivým činnostem v projektu podle tabulky 11. Z Ganttova diagramu lze vyčíst celkový odhad časové náročnosti, který činí 297 dnů a celkové trvání projektu v délce jednoho roku.

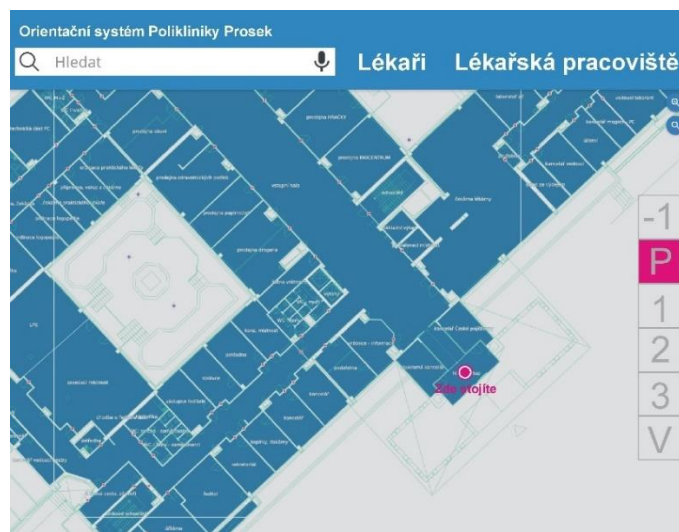


Obrázek 20 - Ganttův diagram projektu vnitřní navigace

4.3 Plánování projektu – První etapa projektu

V rámci plánovací fáze jsem byl seznámen se vstupními podklady a prvotními grafickým návrhem, jak by měla vnitřní navigace vypadat. Grafický návrh zpracoval pan Jan Jandourek, projektový leader společnosti Intellmaps, návrh byl vytvořen pomocí nástroje Figma určeného k vytváření grafických vizuálů pro návrh webové stránky.

Grafické návrhy řešení vnitřní navigace:



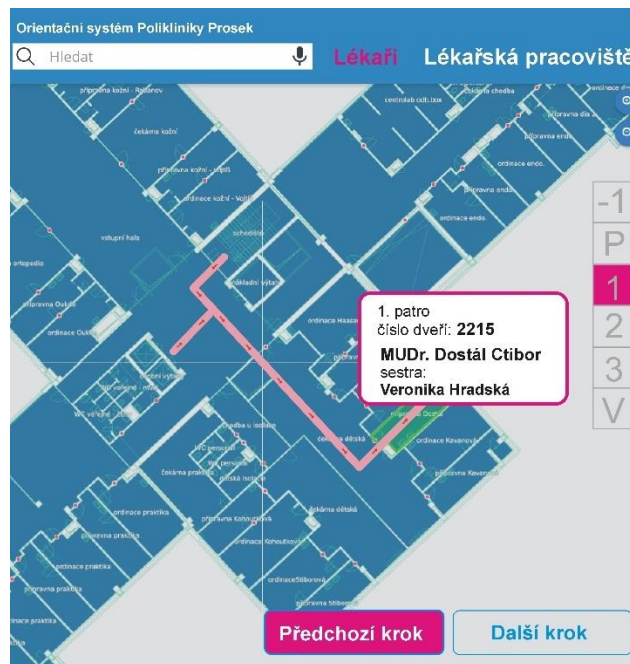
Obrázek 21 - Grafický návrh 1

Obrázek 21 ukazuje vstupní obrazovku pro uživatele, který bude navigaci využívat, hlavním bodem tohoto grafického návrhu je zobrazení vnitřní navigace (Info-kiosku) v plánu budovy, aby se uživatel vnitřní navigace rychle zorientoval určením své polohy v prostoru.



Obrázek 22 - Grafický návrh 2

Obrázek 22 ukazuje, jak bude probíhat vykreslení nejkratší trasy, a jak bude uživatel navigován do cíle.



Obrázek 23 - Grafický návrh 3

Obrázek 23, zobrazuje překleknutí na 1. patro v navigaci a dále ukazuje uživateli navigační cesty v 1. patře. V návrhu je vyznačeno, že v navigaci pro Info-kiosek budou vyhledávání pouze jednotliví lékaři, kteří mají přiřazenu místnost. Základem pro vytvoření seznamu lékařů Polikliniky Prosek jsou volně dostupné webové stránky lekari.poliklinikaprosek.cz. Struktura seznamu lékařů na uvedeném webu je zřejmá z obrázku 23 a bude využita pro vytvoření seznamu lékařů pro účely navigace.

Příjmení	Jméno	Telefon	Místnost	Pracoviště
MUDr. Adamcová	Katarína	608 093 430	3032	Psychiatrie
MUDr. Adámek	Tomáš	266 010 319	2059	Chirurgie
MUDr. Antončíková	Marcela	266 010 313	2-12	Psychiatrie
MUDr. Bačkovský	Petr	266 010 153	2062	Chirurgie
MUDr. Bašný	Zdeněk	266 010 321	2119	Psychiatrie
Mgr. Běhounek	Jan	266 010 145	0083	Psychiatrie
Běhouňková	Iveta	266 010 145	0083	Psychiatrie
MUDr. Biegel	Martin	266 010 114		Ortopedie
MUDr. Bláhová	Jana	266 010 220	1074	Gynekologie
MUDr. Bludovský	Jaroslav	266 010 235	3006	ORL (ušní, nosní, krční)
MUDr. Bálková	Dita	266 010 132	2055	Chirurgie
Doktor sto Bondar	Oleksandr	266 010 241	4005	Stomatologie
MUDr. Brabec	Jiří	266 010 172	0149	Mamologie
Mgr. Brokešová	Aneta	266 010 340	2119	Psychiatrie
MUDr. Bzobyhatá	Nikola	266 010 291	1146	ORL (ušní, nosní, krční)
Mgr. Cmarová	Máňa	724 517 773	4053	Psychologie
MUDr. Černá	Irena		2135	RTG
MUDr. Dolejší	Jana	266 010 143	4021	Ortodoncie
MUDr. Dostál	Ctibor	737 056 301 - mobil	2215	Praktičtí lékaři pro děti a dorost
Doudová	Zuzana	266 010 145	0083	Psychiatrie
MUDr. Ďurovcová	Vítěslava, Ph.D.	734 421 520	2140	Endokrinologie
Doktor sto Elezović	Branislava	266 010 146	4008	Stomatologie

Obrázek 24 - Seznam lékařů

U zmíněného návrhu vnitřní navigace se původně počítalo s tím, že se jednotlivé navigační čáry v podkladu budou vytvářet pouze manuálně, vždy od místa (zde stojíte) k lékaři vybranému ze seznamu. Navrhované řešení bylo zamítnuto v samém počátku projektu, a to ze dvou důvodů.

1) Poliklinika Prosek poskytuje své prostory více než 150 lékařům. Manuální vytvoření více než 150 cest by bylo časově náročné a vysoce neefektivní.

2) Vnitřní navigace má být v dalším projektu součástí projektu navigování uvnitř budovy pomocí bezdrátových senzorů a uživatel bude navigován na základě aktuální dynamické polohy. To by ve výsledku s původním návrhem vyžadovalo manuálně vytvořit pro dvacet bezdrátových senzorů celkem $150 \cdot 20 = 3000$ navigovatelných cest pro vnitřní navigaci.

V rámci plánování byly také vydefinovány základní nástroje a typy souborů pro zpracování budoucích podkladů k vnitřní navigaci. Vstupní podklady ke zpracování byly poskytnuty ve formátu shapefile.

Formát Shapefile:

Ukládá netopologické informace a informace o atributech i jako prostorový doplněk v data setech. Zároveň ukládá do zvláštního data setu i jednotlivé geometrické údaje a jejich doplňky, jako jsou vektorové souřadnice. Shapefile podporuje vytváření bodů, linií a polygonů. K těmto geometrickým reprezentacím dovoluje zapisování i jednotlivých atributů jako databázového formátu, kde každý atributový záznam má vztah jedna k jedné k přidělenému shapefilu (20).

Již z definice shapefilu je patrné, že se jedná o multifunkční formát souborů pro mapový podklad, ve kterém lze vytvářet mnoho parametrů a přiřazovat jim určité vlastnosti. V rámci plánovací fáze a první etapy pro vnitřní navigaci jsem obdržel zpracované shapefilu jednotlivých podlaží Polikliniky Prosek.

Číslo	Název	Typ souboru	Popis
1.	Patro_p_mistnosti	Shp.	Polygon místností přízemního patra + přiřazené atributy
2.	Patro_p_dvere	Shp.	Body značící dveře v přízemním patře + přiřazené atributy
3.	Patro_p_linie	Shp.	Linie ze stavebního náčrtu
4.	Patro_1_mistnosti	Shp.	Polygon místností prvního patra + přiřazené atributy
5.	Patro_1_dvere	Shp.	Body značící dveře prvního patra + přiřazené atributy
6.	Patro_1_linie	Shp.	Linie ze stavebního náčrtu
7.	Patro_2_mistnosti	Shp.	Polygon místností druhého patra + přiřazené atributy
8.	Patro_2_dvere	Shp.	Body značící dveře druhého patra + přiřazené atributy
9.	Patro_2_linie	Shp.	Linie ze stavebního náčrtu
10.	Patro_3_mistnosti	Shp.	Polygon místností třetího patra + přiřazené atributy
11.	Patro_3_dvere	Shp.	Body značící dveře třetího patra + přiřazené atributy
12.	Patro_3_linie	Shp.	Linie ze stavebního náčrtu
13.	Patro_4_mistnosti	Shp.	Polygon místností čtvrtého patra + přiřazené atributy
14.	Patro_4_dvere	Shp.	Body značící dveře čtvrtého patra + přiřazené atributy
15.	Patro_4_linie	Shp.	Linie ze stavebního náčrtu

Tabulka 13 - Seznam vstupních dat

4.3.1 Software QGIS

Pro zpracování tak komplexních souborů byl využit open-source software QGIS. Jedná se o volně dostupný software na zpracování 2D mapových podkladů v různých souřadnicových systémech. Software QGIS je neplacenou alternativou k softwaru Arcgis, který je placený a vlastnění jeho licence stojí tisíce dolarů ročně.

QGIS je geografický informační systém (GIS) publikovaný pod všeobecnou licencí GNU GPL. Projekt QGIS vznikl v roce 2002, verze s označením 1.0 vyšla později v roce 2009. Mezi hlavní výhody patří zejména rychlost vývoje a rozšiřování jeho funkcionality. Licence GNU GPL umožňuje používání softwaru i pro komerční účely. Podstatné je, že umožňuje i modifikaci zdrojového kódu a jeho následné šíření (21).

Software QGIS využívá ve velké míře i společnost Intellmaps a byl mi doporučen pro fázi analýzy projektu a jeho technického zpracování. Do jeho používání jsem byl zaškolen jedním z profesionálních uživatelů, kterým jsem byl upozorněn na důležité faktory, které je potřeba dodržovat, aby nebyla ztracena kompatibilita.

Jedná se o souřadnicový systém ESPG, ve kterém se nachází každá vrstva vytvořená v softwaru QGIS nebo importovaný 'shapefile' soubor, viz tabulka 12.

Zeměpisné souřadnice (šířka φ a délka λ) je nutno vztáhnout ke konkrétnímu referenčnímu tělesu – elipsoidu (spheroid) nebo kouli (sphere). Daný souřadnicový systém je tedy určen:

- referenčním tělesem,
- trigonometrickou sítí bodů, které mají pevně určené nebo jednoznačně odvoditelné hodnoty souřadnic (tato síť tvoří spolu s referenčním tělesem tzv. geodetické datum),
- polohou nultého poledníku (v současnosti v drtivé většině případů mezinárodní nultý poledník procházející nedaleko londýnského Greenwiche),
- jednotkou úhlové míry (nejčastěji stupeň) (22)

Shapefily, které budou zpracovány, jsou v referenčním systému 5514 S-JTSK / Krovak East North – SJTSK. Detailní informace podle definice souřadnicového systému najdeme na stránkách <https://epsg.io/5514>. Důležitým poznatkem je, že jeho střed souřadnic se nachází na -544006.67 a -1144002.81 a jeho hlavní využití je pro oblast Čech a Slovenska. Důvodem použití tohoto referenčního systému je jeho velké zastoupení v České republice a jednoduché zpracování pro převedení do jiných souřadnicových systémů, jako je například systém ESPG: 4326, který je v současnosti světově nejrozšířenějším souřadnicovým systémem pro webové mapové platformy.

4.4 Realizace Projektu – druhá etapa

Realizace projektu začala důkladnou analýzou vstupních dat a vymýšlením vhodných schémat pro vnitřní navigaci. Díky možnostem, které nabízí software QGIS, jako jsou funkce *Create New vector layer*, typ linie, typ bod, vznikla možnost vytvořit graficky neorientovaný obyčejný graf podle povinných náležitostí, kde hrany tvoří jednotlivé linie (typ *Vector Lines*) a vrcholy body (typ *Vector points*). Díky těmto funkcím dokážeme vytvořit neorientovaný graf pro každé podlaží, viz obrázek 25, kde lze vidět vytvořený neorientovaný graf pro druhé nadzemní podlaží Polikliniky Prosek.



Obrázek 25 - Neorientovaný graf pro Polikliniku Prosek

Takto dokážeme pokrýt sítí vrcholů a hran možné dostupné plochy, kam chceme vést všechny potřebné navigovatelné cesty. Z důvodu optimalizace a úspory času, a abychom splnili podmínku cestování po hranách mezi vrcholy na grafu, musíme v grafu znát ohodnocení hran, které bude sloužit jako hlavní parametr při určení nejkratší cesty. Takovým parametrem ohodnocení hrany může být například čas, který potřebujeme k vykonání cesty mezi vrcholem A a vrcholem B na grafu, druhým parametrem může být ohodnocení hrany vzdálenosti mezi vrcholem A a vrcholem B na grafu.

Ohodnocením hrany v případě vnitřní navigace bude vzdálenost mezi jednotlivými vrcholy, informace o vzdálenosti ale nebudeme nikam zapisovat a ani nemusíme provádět reálná měření v objektu. Ohodnocením hran bude vzdálenost mezi jednotlivými vrcholy podle skutečné vzdálenosti. Využijeme skutečnost, že neorientovaný graf kreslíme na mapovém podkladu souřadnicového systému 5514, který používá jako své metrické jednotky metry. Když se zaměříme na obrázek 25, vidíme v pravém dolním rohu výše zmíněný souřadnicový systém a měřítko, ve kterém je mapa vyobrazena, v našem případě 1:405, kdy 1 centimetr na mapě odpovídá přibližně 4 metrům reálných parametrů.

Shapefile dokáže tuhle informaci uchovat v paměti a uloží ji pod speciálním kódováním geom (geometrie), odkud může být dále použita pro výpočty algoritmů.

QGIS není software určený pro tvorbu síťových grafů, používá se spíš jako mapový editor a zapisovač údajů. Z tohoto důvodů je nutné vyzkoušet faktickou propojenost všech hran mezi sebou. Aby nebylo testování náročné, bylo toto testování provedeno pouze skrze vrcholy tzv. artikulace, které jsou pro testování kritickým místem na grafu, protože propojují více hran najednou. Jejich odstraněním se graf rozpadne na podgraf grafu.

4.4.1 Databáze PostgreSQL

PostgreSQL je výkonný objektově relační databázový systém s otevřeným zdrojovým kódem, který využívá a rozšiřuje jazyk SQL v kombinaci s mnoha funkcemi, které bezpečně ukládají a škálují ty nejsložitější datové zátěže. Počátky PostgreSQL se datují do roku 1986 jako součást projektu POSTGRES na University of California v Berkeley, systém má za sebou více než 30 let aktivního vývoje na základní platformě.

PostgreSQL si získal silnou reputaci pro svou osvědčenou architekturu, spolehlivost, integritu dat, robustní sadu funkcí, rozšiřitelnost a odhodlání komunity open-source softwaru trvale poskytovat výkonná a inovativní řešení. PostgreSQL běží na všech hlavních operačních systémech, je kompatibilní s ACID od roku 2001 a má výkonné doplňky, jako je populární rozšiřovač geoprostorových databází PostGIS (23).

Díky kombinaci zapojení nadstavby pgRouting do databáze PostgreSQL jsme schopni vykonávat na grafu síťové analýzy, které jsou detailně popsány v kapitole 3. Pro výpočet nejkratší cesty na grafu bude využit Dijkstrův algoritmus, který optimálně splní požadavek navigovat zákazníky nejkratší cestou po objektu.

Dijkstrův algoritmus, včetně jeho fungování v PostgreSQL databázi je popsán podrobně ve třetí kapitole této práce. Vstup do funkce Dijkstrova algoritmu je proveden pouze jednoduchým importem souborů shapefile, ze software QGIS pomocí funkce *import to database*. Ve třetí etapě bude využito jiných nástrojů, jak vkládat vstupní data do databáze. Spouštění algoritmu funguje stejným způsobem, jak je popsáno v kapitole 3. Pro znázornění zde přikládáme použitý kód z databáze:

```
SELECT * FROM pgr_dijkstra(
... 'SELECT id, source, target, cost, reverse_cost FROM edge_table',
... start, cil,
... FALSE
);
```

Obrázek 26 - Kód pgRouting

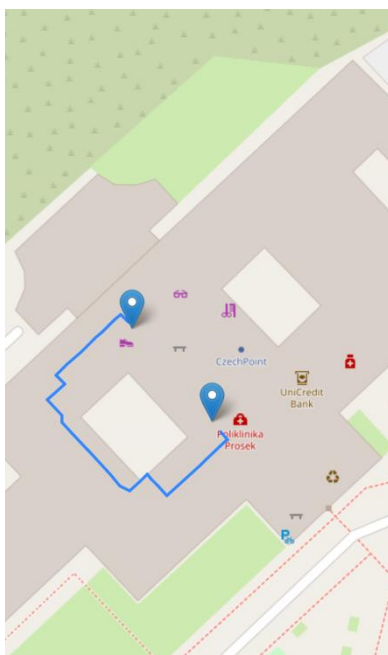
Výstup z databáze je vždy ve formátu atributové tabulky a nese informace o nejkratší cestě na grafu. Tímto způsobem je zajištěno vypočtení nejkratší cesty ve vnitřní navigaci. Výpočtem i vytvořením struktury grafu je vydefinováno i fungování, jak se bude prostřednictvím neorientovaného obyčejného grafu navigovat. Podle SDLC schématu jsme

oranžovým rámečkem, který značí využití technologie pouze statickým způsobem, a který již později dál nevstupuje do datového přenosu dat ve vnitřní navigaci, pouze vkládá data do databáze.

V zeleném rámečku je popsán přenos technologií, které jsou využity pro vykreslování nejkratší cílové cesty na grafu. Funkcionalita Geoserveru je pouze dočasná a i její fungování je pouze dočasné a pouze pro první verzi aplikace.

Ve stručnosti je Geoserver klient, který komunikuje mezi databází a Javascriptem a pouze transformuje data ve formátu sql. z databáze do formátu GeoJSON pro skriptovací (programovací) jazyk JavaScript, který formát GeoJSON vykreslí jako nejkratší cestu formou linie.

Scriptovací jazyk JavaScript v této verzi plní funkci takzvaných rukou vnitřní navigace a pomáhá vykreslovat nejkratší cestu ve vnitřní navigaci první verze. Použití JavaScriptu umožní zobrazovat dynamická data z databáze ve webovém prohlížeči.



Obrázek 28 - První verze vnitřní navigace

Ukázka první verze navigace je zobrazena na obrázku 27 a ukazuje popsané principy ze schématu na obrázku 26. Zobrazují se pouze dva koncové body, které lze umístit na mapě, kde další funkce zobrazí nejkratší cestu. Všechny tyto funkce vykonává skriptovací jazyk JavaScript. Funkce použité v první verzi navigace jsou popsány níže:

- 1) Inicializace mapy – Leaflet
- 2) Přidání podkladové mapy Openstreetmap Leaflet
- 3) Uchycení startovního bodu
- 4) Uchycení koncového bodu
- 5) Funkce pro nalezení nejbližšího bodu
- 6) Funkce pro update koncových bodů
- 7) Funkce pro získání nejkratší cesty z GEOserveru

Celý kód je uveden v příloze 1 práce. Jako příklad toho, jak funguje, si můžeme ukázat 7. funkci v JavaScript kódu.

```
// Funkce pro získání nejkratší cesty z GEOserveru
function getRoute() {
  var url =
    `${geoserverUrl}/wfs?service=WFS&version=1.0.0&request=GetFeature&typeName=
    routing:shortest_path&outputformat=application/json&viewparams=source:
    ${source};target:${target}`;
  $.getJSON(url, function(data) {
    map.removeLayer(pathLayer);
    pathLayer = L.geoJSON(data);
    map.addLayer(pathLayer);
  });
}
```

Obrázek 29 - Kód JavaScript

V příkladu máme funkci Javascriptu getRoute, která volá do softwaru Geoserver, kdy hodnotu kódu `${source}` a `${target}` obdržíme z funkcí 3 a 4. Kód slouží pouze jako předloha pro druhou verzi navigace. Pro tvoření kódu jsem použil zdrojový kód pro volné použití a možnosti úpravy dle vlastní potřeby (24).

4.6 Vývoj druhé verze vnitřní navigace – třetí etapa

V třetí etapě projektu vnitřní navigace bude využito všech poznatků z první verze vnitřní navigace a velká část znalostí z této verze pro vytvoření druhé verze vnitřní navigace. Pro vytvoření druhé verze jsem využil technologie nabídnuté společností Intellmaps, pod kterou jsem celý projekt vnitřní navigace vedl. Tento krok mi dovoluje úpravy v architektuře systému, jakou může být nahrazení technologie Geoserver technologií node.js pro zjednodušení přenosu dat.

4.6.1 Datové zmapování objektu

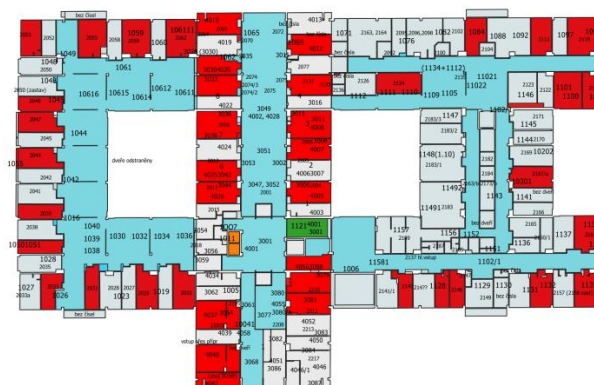
V prvním kroku jsem musel vytvořit neorientovaný graf pro celý objekt Polikliniky Prosek, který bude zahrnut ve vnitřní navigaci. Samotný objekt disponuje pěti podlažními, z toho jsou čtyři nadzemní a jedno podzemní.

- 1. podzemní podlaží: 135 místností zaměřených v polygonu
- 1. nadzemní podlaží: 138 místností zaměřených v polygonu
- 2. nadzemní podlaží: 154 místností zaměřených v polygonu
- 3. nadzemní podlaží: 61 místností zaměřených v polygonu
- 4. nadzemní podlaží: 48 místností zaměřených v polygonu

Celkem se nachází v objektu 536 potencionálních míst, kam může být uživatel navigován.

Fyzický stav budovy disponuje celkem 536 místnostmi. Požadavek k navigování ovšem počítá pouze s navigováním k lékařům. Seznam aktivních lékařů je volně dostupný na webové stránce Polikliniky Prosek <https://lekari.poliklinikaprosek.cz/>.

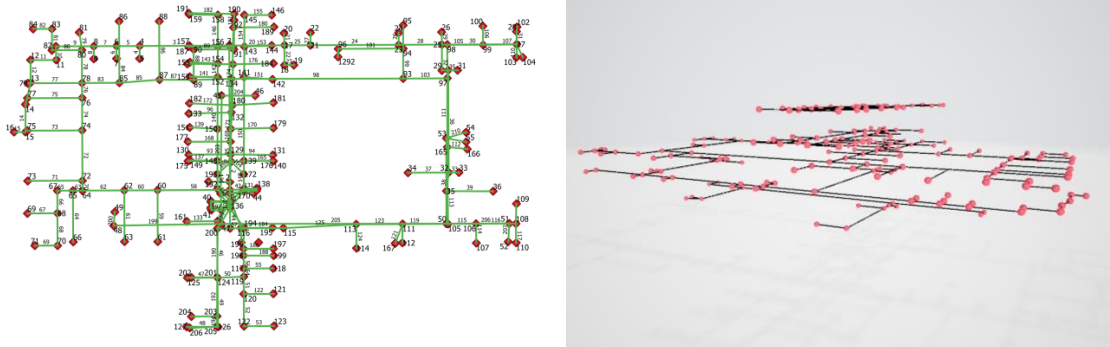
Ze seznamu obdržím hodnotu 140 aktivních lékařů v objektu. Každému lékaři je přiřazeno číslo místnosti/dveří. Podle čísel dveří si můžeme vydefinovat místnosti v jednotlivých podlažích, jak je zobrazeno na obrázku 28, kde červeně označená místa znamenají obsazení místnosti konkrétním lékařem.



Obrázek 30 - Zmapované 2. patro Polikliniky Prosek

Stejným způsobem zmapuji i ostatní čtyři patra a do jednotlivých nákrešů zakreslím aktivní místnosti. Pokud znám aktivní místnosti a jejich unikátní číslo, kterým je číslo místnosti/dveří, tak postupuji s tvorbou neorientovaného obyčejného grafu pro celý objekt.

Výsledný neorientovaný graf obsahuje celkem 210 neorientovaných hran a 208 vrcholů. Z pohledu teorie grafů se nejedná o velký graf. V případě 'indoor' navigování již počet cílových vrcholů násobně převyšuje počet vrcholů pro vnitřní navigace umístěné v obchodních centrech.



Obrázek 31 - Neorientovaný graf Polikliniky Prosek ve 2D a 3D pohledu

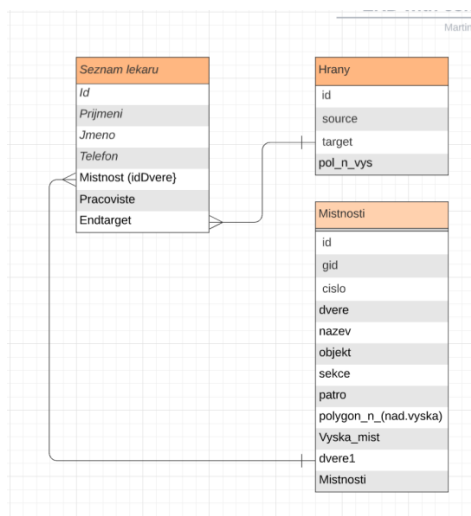
V naší vnitřní navigaci chceme navigovat uživatele po objektu podle jména lékaře. Momentálně disponujeme pouze údaji o grafu, který má svojí vizuální podobu zapsanou i v atributové tabulce *Hrany* a nenese žádné údaje o lékařích.

- Tabulka *Hrany* (Id – id hrany, source – startovní vrchol, target – cílový vrchol, pol_n_vys – nadmořská výška hrany / patro)

Druhou tabulku, kterou potřebujeme disponovat, abychom mohli navigovat uživatele podle jmen lékařů, je tabulka *Seznam lékařů*. Tabulku *Seznam lékařů* si vytvořím z webové stránky <https://lekari.poliklinikaprosek.cz/>. Data z webu jsem převedl do formátu csv. Aby bylo možné vytvořit propojení dvou tabulek pomocí pravidel SQL, tak ke každému záznamu lékaře, který má přiřazenu místnost, přiřadím hodnotu našeho koncového vrcholu z tabulky hrany.

- Tabulka *Seznam lékařů* (Id – id doktora, Příjmení, Jméno, Telefon, Místnost (iddveře), Pracoviště, Endtarget = hodnota sloupce target (tabulka hrany))

Propojení tabulky *Seznam lékařů* a tabulky *Hrany* je provedeno vztahem N:1. To znamená, že více lékařů může být napárováno pouze na jeden vrchol z neorientovaného grafu.



Obrázek 32 - Schéma propojení tabulek

V druhé verzi je propojení tabulek jednoduché, ale pro budoucí verze vnitřní navigace se propojování tabulek jeví jako efektivní způsob, jak zpracovat vyhledávání v aplikaci a umožnit rozšířené vyhledávání.

4.7 Vývojový postup vnitřní navigace – třetí etapa

Vývojová část druhé verze vnitřní navigace zahrnuje mnoho poznatků z první a druhé etapy projektu a z první verze vnitřní navigace. Vychází se ze stejných principů pro navigování. Podstatných změn doznala architektura systému po zaimplementování do technologií / prostředí společnosti Intellmaps. Pro spravování technologie společnosti Intellmaps musela být domluvena spolupráce s programátorem společnosti Intellmaps, Ing. Radoslavem Inhrou, aby provedl úpravy v kódu při implementaci vnitřní navigace, například pro technologii node.js.

Vývojová část vnitřní navigace je rozdělena do tří částí, kde jednotlivé části podrobně popisují postupy vývoje při využití jednotlivých technologií. Vývojová část neobsahuje všechny části programového kódu, ale pouze jeho hlavní části, které vykonávají nejdůležitější operace, na kterých je založena vnitřní navigace. Důvodem zobrazování pouze části kódu (2. část) je zasazení vnitřní navigace do struktur kódu společnosti Intellmaps. Proto nelze uvést celý kód v příloze. Struktura kódu společnosti Intellmaps zahrnuje jiné softwarové produkty a technologie, které využívá i vnitřní navigace. Příkladem sdílené technologie je zobrazování souborů shapefile na webovém prohlížeči pomocí technologie CartoDB. Využití technologií firmy Intellmaps je zahrnuto v požadavcích projektu.

Pro ilustraci je vývoj zobrazen schématicky na obrázku 31, který zobrazuje jednak posun ve vývoji vnitřní aplikace ve srovnání s první etapou, jednak využití technologie v druhé verzi

vnitřní navigace a její vylepšení. Na schématu je zobrazena i kooperace s technologiemi společnosti Intellmaps. Pro jednodušší strukturu a popis částí vývoje je schéma rozděleno do tří částí:

1. část – Operace pro vkládání vytvořených vstupů do databáze přes technologii CartoDB

V této části se vkládají data do webové aplikace Builder společnosti Intellmaps, která umožňuje zobrazovat nahrané podklady v mapě aplikace Intellmaps. Tento krok tak umožní vytvářet vnitřní navigaci pro různé mapové podklady a půdorysy budov.

2. část – Fungování operací v pozadí a databázové operace pro tvorbu nejkratší cesty

V druhé části jsou zpracovány poznatky z první verze vnitřní navigace. Je odebrán software GeoServer, který převáděl data z databáze ve formátu sql do GeoJSON formátu. Tento postup je nahrazen komplexním kódem pro výpočet nejkratší cesty a funkcí vytvoření výstupu nejkratší cesty již ve formátu GeoJSON.

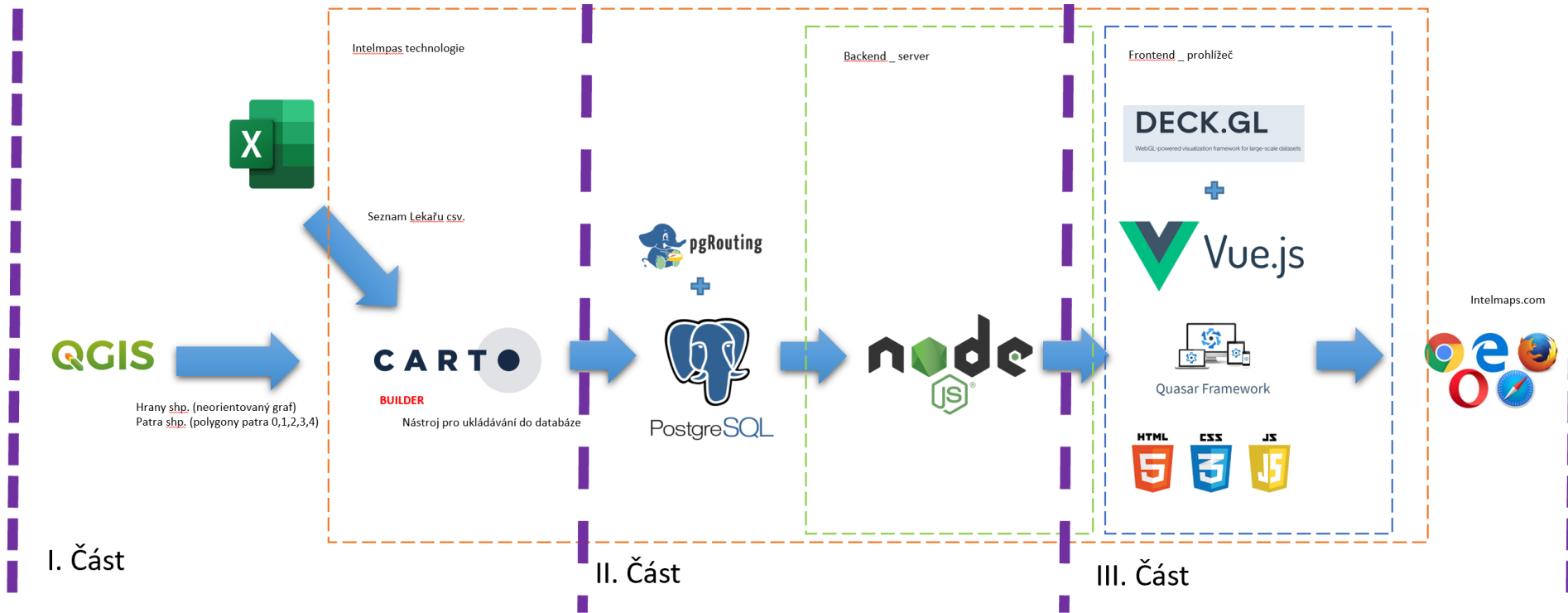
```
SELECT jsonb_build_object(
  'type',      'FeatureCollection',
  'features',  jsonb_agg(features.feature)
) as geoJSON
FROM (
  SELECT jsonb_build_object(
    'type',      'Feature',
    'id',        seq,
    'geometry',  ST_AsGeoJSON(the_geom)::jsonb,
    'properties', to_jsonb(inputs) - 'gid' - 'the_geom'
  ) AS feature
FROM (
  SELECT routing.seq, edge.pol_n_vys, edge.the_geom, edge.patro FROM
pgr_dijkstra(
  'SELECT CAST (id AS INTEGER), CAST (start AS INTEGER) as
source, CAST (cil AS INTEGER) as target, ST_Length(the_geom) as cost,
ST_Length(the_geom) as reverse_cost FROM adminjan.hrany',
  69, 29,
  FALSE) as routing
LEFT JOIN adminjan.hrany AS edge ON routing.edge = edge.id
ORDER BY routing.seq
) AS inputs)
AS features;
```

Tento kód je implementován do prostředí node.js, které při každém volání funkce pošle kód do databáze s vydefinovanými vrcholy (start, cíl). Databáze obratem vrátí výstup nejkratší cesty v GeoJSON formátu, který node.js pošle do Javascriptu / Vue.js.

3. část – Fungování operací pro prohlížeč a další práce spojené se zobrazením dat

Třetí část tvoří programovací část v oblasti webového programování, pro programování jsou použity technologie Vue.js, které tvoří nadstavbu kombinace Javascriptu, html a css. Většina vzhledových komponent byla vytvářena pomocí dostupné knihovny Quasar.js, která poskytuje již mnoho vytvořených webových vizualizací a funkcionalit.

Všechny části kódu z vnitřní navigace využívají dostupnosti z webové stránky <https://quasar.dev/>, která nabízí již mnoho předpřipravených komponent pro webové programování. Celkové fungování aplikace vnitřní navigace zobrazuje již výše zmíněné schéma na obrázku 31, které znázorňuje jednotlivé postupy - od vkládání dat po jejich zpracování jednotlivými nástroji, které jsou pro zpracování využity.



Obrázek 33 - Schéma architektury technologií druhé verze vnitřní navigace

4.8 Vnitřní navigace – Výsledek a vyhodnocení, projektu, zpětná vazba od uživatelů

Tento oddíl obsahuje vyhodnocení projektu z hlediska naplnění jím stanovených cílů a požadavků. Dále se zabývá zpětnou vazbou získanou v rámci testování aplikace vnitřní navigace Polikliniky Prosek od jejích potencionálních uživatelů a možností budoucího vylepšení aplikace na základě této zpětné vazby.

4.8.1 Souhrnné vyhodnocení splnění požadavků projektu

Jednotlivé požadavky, které měl projekt splnit, byly stanoveny již v první fázi projektu a jsou shrnuty v tabulce 10 v pododdíle 4.2.1 této kapitoly s názvem „Hlavní cíle projektu navigace“. Jedná se celkem o 9 jednotlivých cílů / požadavků. Kromě základního cíle, kterým byl požadavek na vytvoření funkční aplikace vnitřní navigace ve vnitřních prostorách Polikliniky Prosek určené pro prostředí webového prohlížeče, obsahuje seznam cílů i požadavek na použití mapových technologií pro navádění v navigaci, požadavek na použití dostupných technologií tzv. open-source řešení, které měly zajistit cenově přijatelnou realizaci, a šest požadavků na různé způsoby navádění uživatelů v budově z pohledu funkčnosti, resp. možnosti rozšíření funkčnosti aplikace. Výsledný produkt aplikace pro vnitřní navigaci vytvořený pro objekt Polikliniky Prosek, kterým je druhá verze navigace, splňuje, jak je vidět z tabulky č. 13 níže, šest z těchto devíti požadavků na 100 % a je možné konstatovat, že tato verze již tvoří plnohodnotný produkt, který lze využít pro testovací fázi nebo zjištění uživatelské spokojenosti. Zbývající požadavky, které byly splněny na méně než 100 % procent, ale více než na 50%, nejsou ještě, stejně jako požadavek na navádění v režimu indoor / outdoor, ve výsledné aplikaci zahrnuty z pohledu její aktuální funkčnosti, nicméně jsou zahrnuty do architektury druhé verze navigace a představují technologický základ pro budoucí verze.

Požadavek	Procentuální splnění požadavku
1. Vytvořit vnitřní navigaci pro webový prohlížeč	100%
2. Použít dostupné technologie / Opensource řešení	100%
3. Použít mapové technologie k navádění	100%
4. Navádění zákazníků z předem určeného vrcholu (info kiosky)	100%
5. Navádět z různých vrcholů na grafu vnitřní navigace	80%
6. Navádět po schodech nebo výtahem	80%
7. Navádět klienty po budově k lékařům	100%
8. Navádění na více míst najednou	60%
9. Navádění v režimu outdoor / indoor	100%

Tabulka 14 - Naplnění požadavků projektu

4.8.2 Shrnutí výsledků

- První čtyři klíčové požadavky, kterým byla v projektu podle tabulky 10 přiřazena vysoká priorita, byly splněny na 100 %. Aplikace vnitřní navigace pro objekt Polikliniky Prosek je vytvořena pro prostředí webového prohlížeče, což ji předurčuje k využití uživateli nejen ve statických navigačních kioscích, ale i přímo z webové stránky poskytovatele.

- Vnitřní navigace plně využívá dostupné technologie, tzv. open-source technologie / zdroje, jejichž využívání je dovoleno i pro komerční účely a nejsou licencované pro specifickou společnost. Stoprocentní využití pouze open-source technologií ve všech částech projektu znamená nižší náklady projektu a snazší budoucí vývoj a cenově dostupné produkty vnitřní navigace.

- Vnitřní navigace využívá plně mapové podklady, její zobrazení lze tedy pevně upevnit v těchto mapových podkladech, které mohly být primárně zpracovány i pro jiné účely.

Použití mapových podkladů může být prospěšné pro budoucí vývoj aplikace, zejména v oblasti 3D vnitřní navigace.

- Navádění z Info-kiosku na předem vybrané místo je splněno. V aplikaci lze provést úpravy výchozího bodu, pokud by se v budoucnu změnila poloha kiosku při jeho fyzickém přesunutí.

- Uživatel bude navigován na přesně stanovené cíle, které bude mít na výběr. Uživatel je do vybraného cíle naveden se 100 % přesností.

Výše uvedené body jsou požadavky, které byly splněny na 100 % a jsou pevně implementovány do současné verze vnitřní navigace.

Níže uvedené požadavky (viz odrážky níže) byly naplněny na méně než 100 % své funkčnosti, proto jejich funkční zapracování do aplikace nebylo možné. Ovšem vnitřní nastavení aplikace a architektura, na níž je postavena, již s těmito požadavky počítají a z určité části disponují možnostmi tyto požadavky do aplikace postavené na druhé verzi vnitřní navigace zavést.

- Navádění z různých vrcholů grafu je pro uživatele aplikace aktuálně nemožné. Ovšem navádění po grafu je již možné z různých vrcholů a připraveno pro chvíli, kdy aplikace bude umět zjišťovat polohu z různých míst objektu.

- Možnost navádět uživatele po schodech nebo výtahem je realizována vložením dvou neorientovaných grafů do databáze. Jeden neorientovaný graf je mezi patry propojen skrze výtahové šachty a druhý neorientovaný graf skrze schodiště. Aby tento krok mohl být aplikován, musí být ještě provedeno vícero změn v kódu.

- Navádění na více míst najednou může být v aplikaci uplatněno v dalších verzích při přesné úpravě Dijkstrova algoritmu a se zavedením možnosti zadávat více bodů navigování najednou.

Popis druhé verze navigace je uveden na obrázku 32 a pod ním je uvedena legenda k jednotlivým funkcionalitám druhé verze navigace označeným číslicemi 1 až 5.



Obrázek 34 - Výsledná vnitřní navigace verze 2

- 1) Vybrat ze seznamu lékařů v tabulce pod navigací.
- 2) Sledovat zobrazení výchozí linie z kiosku v prvním patře směrem k výtahu.
- 3) Přepínáním pater zobrazit pomocí podbarvení uživateli cílové patro a ukázat, kam má pokračovat.
- 4) Přepnout do podbarveného cílového patra.
- 5) Sledovat zobrazení od výtahu do cílového vrcholu / destinace.

4.8.3 Zpětná vazba na aplikaci vnitřní navigace

Protože aplikace vnitřní navigace pro objekt Polikliniky Prosek je primárně službou návštěvníkům objektu, měla by proto, stejně jako ostatní poskytované služby, především splňovat požadavky typu klientů, kteří ji budou využívat (cílové skupiny). Získávání zpětné vazby od uživatelů je běžně používaná metoda pro zdokonalování služeb i produktů. Z tohoto

důvodu jsme i do našeho projektu zařadili získání zpětné vazby alespoň od malého vzorku potencionálních uživatelů. Testování jsme prováděli na přenosném zařízení přímo v prostorách Polikliniky Prosek. Předně jsme obdrželi zpětnou vazbu od správce budovy Polikliniky Prosek, pana Ing. Fischera, který vnitřní navigaci ohodnotil slovně:

„Navigace dokáže navigovat ke všem lékařům po budově, a pokrýt tak velký objekt, jako je Poliklinika Prosek. Velkou výhodou vidím v jednoduché úpravě navigačních cílů při změně lékařů nebo možnost navigování uživatelů mezi patry. Mezi možné budoucí úpravy bych zařadil lepší vyhledávání jednotlivých lékařů v seznamu.“

Aby zhodnocení bylo nestranné, byla vnitřní navigace představena třem náhodným kolemjdoucím, kteří měli krátké slovní ohodnocení:

1. „Je to pěkné, ale takovým technologiím nevěřím, navigace mi přijde málo přehledná.“
2. „Navigaci v podobných budovách bych ocenila. Zlepšit byste měli přehlednost podlaží a vyhledávání lékařů v seznamu.“
3. „Díky navigaci jsem zjistil, kde se nachází kožní ordinace. Nápad vnitřní navigace se mi líbí.“

I na základě krátké zpětné vazby čtyř osob lze již získat cenné informace před uvedením aplikace vnitřní navigace do ostrého provozu. Zaměřit bychom se měli zejména na grafickou úpravu vzhledu, aby aplikace působila již na první pohled přehledněji. Druhou zásadní informací ze zpětné vazby je nutnost zlepšení vyhledávání jednotlivých lékařů ve fázi zadávání cíle v navigaci. Vyhledávání lékařů musí působit přehledně a jednoduše.

4.8.4 Vyhodnocení zpětné vazby na aplikaci vnitřní navigace

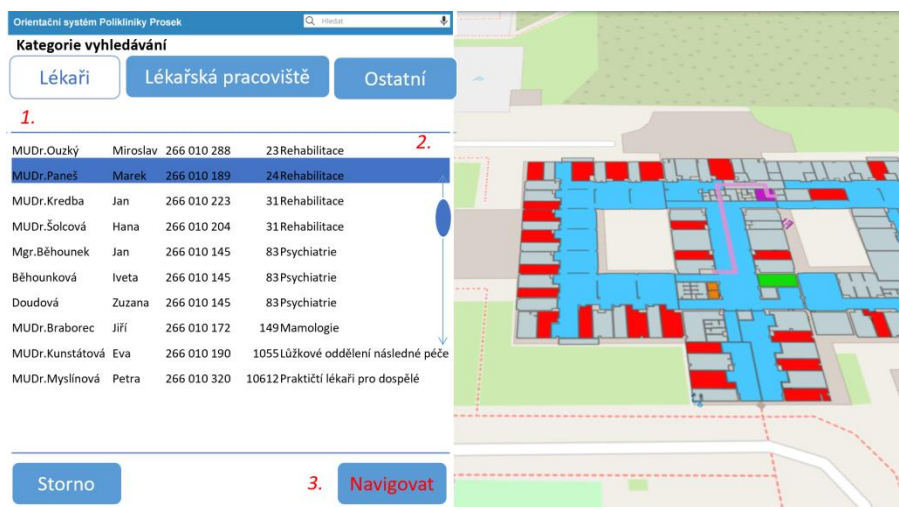
Zpětná vazba, ačkoliv krátká, přinesla mnoho podnětů, na které je nutné se v budoucnosti zaměřit a věnovat jim zvýšenou pozornost. Jedním takovým podnětem je vytvoření přehledného vyhledávání / zadávání cílových bodů, tj. lékařů, k nimž má být uživatel navigován na startu vnitřní navigace. V aplikaci postavené na druhé verzi vnitřní navigace, která byla použita pro testování na Poliklinice Prosek, byl pro vyhledání lékaře na startu navigace použit abecedně řazený rolovací seznam jednotlivých lékařů. Tento postup je zejména při tak velkém množství cílových bodů / lékařů nepřehledný a časově náročný. Uživateli by mělo být nabídnuto více možností volby, jak zadat, resp. vyhledat v navigaci svůj cíl / požadovaného lékaře.

a) Vyhledáním podle jména lékaře

b) Výběrem ze seznamu lékařů

c) Volbou/výběrem ze seznamu podle kategorie pracoviště (např. oční, kožní, rehabilitace apod.)

Budoucí vzhled vnitřní navigace by mohl být upraven, jak je navrženo na obrázku 33, což zlepšší možnosti vyhledávání.



Obrázek 35 - Budoucí vylepšení

Nejen na základě poskytnuté zpětné vazby, kdy byl dotazovaný vzorek respondentů poměrně malý, se můžeme domnívat, že vnitřní navigaci budou spíše využívat mladí lidé nebo lidé, kteří mají větší důvěru v informační technologie, naopak starší lidé budou volit jiné metody pro nalezení cílového místa / lékaře. Nicméně dotazovaným respondentům za jejich zpětnou vazbu tímto děkujeme a doufáme, že postupem času se tento nástroj vnitřní navigace stane pro klienty Polikliniky Prosek běžnou součástí informačního systému klientů a rozšíří se postupně i do dalších podobných zařízení.

4.8.5 Shrnutí

Na závěr oddílu 4.8 je nutné konstatovat, že cíle projektu byly splněny, byl dodržen i stanovený časový harmonogram projektu a produktem projektu byla funkční aplikace vnitřní navigace.

Z pohledu projektového řízení je nutné si uvědomit, že vytvoření aplikace vnitřní navigace s využitím řešení (vnitřní architektury), které ještě nebylo aplikováno nebo tato aplikace není veřejně známá, představovalo i riziko toho, že projekt nebude dokončen a skončí neúspěšně. Projekt tedy v sobě nesl velkou míru rizika, protože si dal za cíl vytvořit funkční aplikaci

postavenou na vnitřní architektuře představující potencionálně nové, dosud neaplikované řešení. Při zahájení projektu tak bylo nutné počítat i s možností ukončení projektu v jeho průběhu. Díky dobrému managementu a perfektní spolupráci i nadšení celého projektového týmu se podařilo projekt dovést do vítězného konce.

5 Závěr

Diplomová práce popisuje, analyzuje a vyhodnocuje projekt vývoje a implementace vnitřního 'indoor' navigačního systému pro konkrétní vybraný objekt, v jehož rámci autor práce vytvořil návrh architektury systému pro vnitřní navigaci, založené na praktickém využití teorie grafů, a v rámci projektu ji dovedl až k praktické realizaci ve formě uživatelské aplikace.

Projekt byl realizován ve spolupráci s technologickou start-upovou společností Intellmaps s.r.o., která se zaměřuje na vývoj interaktivních mapových aplikací a navigačních informačních systémů vyvíjených prostřednictvím moderních technologií. Cílem práce i vlastního projektu bylo vytvořit navigační architekturu pro vnitřní objekty, včetně vícepodlažních, založenou na dostupných teoretických poznatcích z již zmíněné oblasti teorie grafů, dále na použití dostupných mapových technologií navádění a realizovanou prostřednictvím open-source software, tedy technologií, které dokážou nahradit placený software. Předpokládaným cílovým výstupem projektu bylo vytvoření aplikace vnitřní navigace postavené na webovém rozhraní pro konkrétní objekt Polikliniky Prosek, s tím že architektura této „pilotní verze“ bude snadno aplikovatelná a využitelná při vytváření vnitřních navigací i pro jiné vnitřní objekty, tedy bude tvořit základ a platformu budoucích realizací.

Pro naplnění cíle práce i projektu jako takového autor postupoval následovně. V teoretické části v kapitole 2 nejprve popsal evoluční vývoj navigace jako takové, principy fungování vnitřní navigace a způsoby jejího využití, včetně porovnání nabídky vnitřních 'indoor' navigací na současném trhu. V kapitole 3, která také spadá do teoretické části práce, představil teoretické předpoklady projektu reprezentované teorií grafů, která byla v projektu využita jako teoretický základ vnitřní navigace, včetně popisu operací na grafech a popisu algoritmů, které v projektu při vývoji vnitřní navigace uplatnil. Rovněž představil použitou softwarovou podporu pro výpočet algoritmů uplatněnou následně v praktické části práce při vlastním vývoji vnitřní navigace. Praktickou část práce, která je obsažena v kapitole 4 a zaměřuje se na samotný vývoj vnitřní navigace, zahrnující jak vývoj vnitřní architektury této navigace, tak vývoj webové uživatelské aplikace postavené na této platformě, realizoval autor práce ve spolupráci se společností Intellmaps formou projektové aktivity, popisuje v ní tedy průběh projektu. V úvodu kapitoly stručně představuje teoretické základy projektového řízení, následuje stanovení hlavních cílů projektem, popis objektu řešeného projektem a očekávaných přínosů řešení vnitřní navigace pro tento objekt. Dále představil a popsal jednotlivé fáze projektu, v nichž zvláštní pozornost věnoval výběru softwarového řešení a technologickým možnostem, které v projektu zvolil. V závěru kapitoly 4 poté prezentoval

výstupy projektu, jeho vyhodnocení z pohledu splnění stanovených cílů, vyhodnocení zpětné vazby uživatelů na výsledný produkt projektu, kterým je aplikace vnitřní navigace pro projekt Polikliniky Prosek.

Z vyhodnocení projektu v závěru kapitoly 4 vyplývá, že cíl práce i projektu byl splněn, přestože projekt v sobě nesl velkou míru rizika, protože si dal za cíl vytvořit funkční aplikaci postavenou na vnitřní architektuře představující potencionálně nové, dosud neaplikované řešení a na jeho počátku nemohlo být garantováno, že bude úspěšně dokončen, a bylo tedy nutné počítat i s možností předčasného ukončení projektu v některé z jeho fází. Nicméně webová aplikace vnitřní navigace pro Polikliniku Prosek byla projektem úspěšně vytvořena, zástupcům polikliniky představena a mobilně testována na přenosném zařízení. Splněn nebyl jenom tento hlavní cíl – vytvořit uživatelskou aplikaci, ale i dílčí cíle projektu, z nichž mnohé jsou důležité pro další rozvoj aplikace směrem k širší nabídce funkcí i uživatelskému komfortu.

Důležitým prvkem je, že navigační architektura pro vnitřní objekty, kterou autor práce vytvořil, a na které je výše uvedená aplikace postavena, není izolovaným řešením, ale představuje platformu pro vytváření analogických aplikací vnitřní navigace pro další objekty a umožňuje vývoj dalších funkcionalit i propojení na jiné systémy, přičemž je vystavěna na dostupných technologiích, které jsou předpokladem cenové přijatelnosti aplikací na nich postavených.

Další využití webové aplikace vnitřní navigace je možné například pro objekty, jako jsou veřejné budovy, pro státní správu, zdravotnictví a školství, nebo objekty služeb a nákupní centra, případně výpravní budovy železniční dopravy, příletové a odletové haly letišť s přilehlými parkovacími domy, mohou to ale být i průmyslové nebo skladové areály. Vnitřní navigace může být užitečná pro veřejnost vyhledávající určitý úřad nebo službu, pacienty vyhledávající ordinaci lékaře, studenty vyhledávající učebny nebo místa zkoušek, cestující hledající místa parkování a odjezdu nebo odletu. Současně vnitřní navigace může být určená také pracovníkům externího servisu nebo pracovníkům provádějícím prohlídky a revize všech možných technických zařízení budov, jako jsou například dveřní a vratové systémy, systémy klimatizace a vzduchotechniky, osvětlení budov, zabezpečovací systémy. Všem těmto potencionálním uživatelům pomůže ke zkrácení času potřebného pro hledání určitého místa v určitém objektu na minimum.

Z příkladů možných objektů a důvodů vhodných pro aplikaci vnitřní navigace vyplývá, že nestačí jen poptávka všech návštěvníků, pacientů, zákazníků, cestujících, pracovníků provádějících servis nebo revize atd., tedy jedním termínem uživatelů vnitřní navigace, ale proti tomu musí být nabídka všech správců budov, majitelů, úřadů, poskytovatelů služeb a

dopraců, tedy jedním termínem provozovatelů vnitřní navigace. Těm může přinést výhody z toho, že svým zákazníkům poskytnou kvalitnější služby, zejména profitují-li z jejich počtu, a provozovatelům servisních i jiných služeb i finanční výhody ve formě snížení nákladů při lepším využití fondu pracovní doby svých zaměstnanců. Tyto benefity lze očekávat i při využití vnitřní navigace v průmyslu, zejména ve skladovém hospodářství.

Autor práce je přesvědčen, že oblast vnitřní navigace má v současném technologicky zaměřeném světě velký potenciál, a doufá, že i jeho práce přispěje alespoň malým dílem k jejímu budoucímu rozšíření.

6 Citovaná literatura

1. Wikipedia. *Navigace*. [Online] <https://cs.wikipedia.org/wiki/Navigace>.
2. Bouma, Ondřej. fi.muni.cz. *HISTORIE A VÝVOJ SATELITNÍCH NAVIGAČNÍCH SYSTÉMŮ*. [Online] 15. 04 2003. <https://www.fi.muni.cz/usr/jkucera/pv109/2003/xbouma.htm>.
3. Was ist Indoor Navigation ? . *Favendo.com*. [Online] Favendo, 07. 01 2020. <https://www.favendo.com/de/indoor-navigation-de/was-ist-indoor-navigation>.
4. WDF. Mistoprodeje. *Mistoprodeje*. [Online] <https://www.mistoprodeje.cz/obsah/pos-aktuality/technologicky-trend-v-retailu-indoor-navigace/>.
5. Helios.eu. *Helios.eu*. [Online] https://forum.helios.eu/orange/doc/cs/Soubor:Logistika_-_prijem.gif.
6. Mecalux. *Mecalux*. [Online] <https://www.mecalux.cz/technicka-prirucka-pro-skladovani/navrh-skladu/usporadani-skladu>.
7. Plzen.eu. *Sport*. [Online] <https://www.plzen.eu/o-meste/aktuality/aktuality-z-mesta/fanousci-fc-viktoria-plzen-mohou-vyuzit-novou-aplikaci-ma-radu-uzitecnych-funkci.aspx>.
8. Plzně, Správa Informačních technologií města. *Aplikace FC ViktoriaPlzeň*. [AppleStore] Plzeň : Správa Informačních technologií města Plzně.
9. Motivace. *Matematická biologie*. [Online] <https://portal.matematickabiologie.cz/index.php?pg=zaklady-informatiky-pro-biology--teoreticke-zaklady-informatiky--teorie-grafu--motivace>.
10. Doc. Ing. Denisa Mocková, Ph.D. *Základy teorie dopravy*. místo neznámé : ČVUT v Praze, 2007.
11. What is the best shortest path algorithm. *MyRouteOnline*. [Online] <https://www.myrouteonline.com/blog/what-is-the-best-shortest-path-algorithm>.
12. Matematická biologie. *Dijkstrův algoritmus*. [Online] <https://portal.matematickabiologie.cz/index.php?pg=zaklady-informatiky-pro-biology--teoreticke-zaklady-informatiky--teorie-grafu--optimalizacni-ulohy-nad-grafy--dijkstruv-algoritmus>.
13. Hsu, Regina O. Obe a Leo S. *pgRouting A practical Guide*. 978-0989421737.
14. pgRouting. *pgRouting*. *pgRouting*. [Online] <https://pgrouting.org/index.html>.
15. pmconsulting. *Etapa projektu*. [Online] <https://www.pmconsulting.cz/slovníkovy-pojem/etapa-projektu/>.
16. pmconsulting. *Faze řízení projektu*. [Online] <https://www.pmconsulting.cz/slovníkovy-pojem/faze-řízení-projektu/>.
17. Tx.cz. *SDLC*. [Online] <https://www.tx.cz/slovník/isms/sdlc>.
18. innovativearchitects. *SDLC models*. [Online] innovativearchitects. <https://www.innovativearchitects.com/KnowledgeCenter/basic-IT-systems/8-SDLC-models.aspx>.

19. Poliklinika Prosek. [Online] <https://www.poliklinikaprosek.cz/stranka-poliklinika-prosek-179>.
20. ESRI. *ShapeFile*. [Online] <https://www.esri.com/Library/Whitepapers/Pdfs/Shapefile.pdf>.
21. Gisementor. *QGIS zacatecnik*. [Online] <https://training.gismentors.eu/qgis-zacatecnik/>.
22. [Online] <https://www.natur.cuni.cz/geografie/geoinformatika-kartografie/ke-stazeni/projekty/moderni-geoinformacni-metody-ve-vyuce-gis-kartografie-a-dpz/souradnicove-systemy/>.
23. PostgreSQL. *About PostgreSQL*. [Online] <https://www.postgresql.org/about/>.
24. GitHub. [Online] <https://github.com/thegisdev/leaflet-pgrouting-geoserver>.

7 Seznam obrázků

Obrázek 1 - Navigace a jejich kombinace	12
Obrázek 2 – Vnitřní navigace (Favendo.com, 2020)	13
Obrázek 3 - Příjem zboží na sklad (Helios.eu, n. d.)	17
Obrázek 4 - Vnitřní navigace ve skladu (Mecalux, n. d.)	18
Obrázek 5 - Navigace aplikace FCViktoriaPlzen (Plzně, n. d.)	19
Obrázek 6 - Obyčejný neorientovaný graf (vstup)	28
Obrázek 7 - Výpočet Dijkstrova algoritmu	29
Obrázek 8 - Postup výpočtu Dijkstrova algoritmu	29
Obrázek 9 - Výsledek výpočtu Dijkstrova algoritmu	30
Obrázek 10 – Komptabilita verzí modulů a softwarů	32
Obrázek 11 - Prostředí výpočtu pgRouting v PostgreSQL.....	34
Obrázek 12 - Převedení grafu do strukturované tabulky (priklad_table).....	35
Obrázek 13 - Stupně etap a fází projektu	38
Obrázek 14 - SDLC diagram (SDLC models, n. d.)	40
Obrázek 15 - Grafické znázornění spirálové metody	42
Obrázek 16 - Cíl projektu vnitřní navigace.....	43
Obrázek 17 - Poloha Polikliniky Prosek.....	46
Obrázek 18 - Poliklinika Prosek.....	47
Obrázek 19 - 1. patro s polohou kiosku	47
Obrázek 20 - Ganttův diagram projektu vnitřní navigace	50
Obrázek 21 - Grafický návrh 1.....	51
Obrázek 22 - Grafický návrh 2.....	51
Obrázek 23 - Grafický návrh 3.....	52
Obrázek 24 - Seznam lékařů	52
Obrázek 25 - Neorientovaný graf pro Polikliniku Prosek.....	56
Obrázek 26 - Schéma první verze vnitřní navigace	58
Obrázek 27 - První verze vnitřní navigace.....	59
Obrázek 28 - Zmapované 2. patro Polikliniky Prosek	61
Obrázek 29 - Neorientovaný graf Polikliniky Prosek ve 2D a 3D pohledu	62
Obrázek 30 - Schéma propojení tabulek.....	63
Obrázek 31 - Schéma architektury technologií druhé verze vnitřní navigace	66
Obrázek 32 - Výsledná vnitřní navigace verze 2	70
Obrázek 33 - Budoucí vylepšení.....	72

8 Seznam tabulek

Tabulka 1 - Druhy navigačních technik	12
Tabulka 2 - Kalkulace pořizovacích nákladů indoor navigací	15
Tabulka 3 - Výhody a nevýhody aplikace	20
Tabulka 4 - Teorie grafů (Mocková, 2007)	22
Tabulka 5 - Typy grafů (Mocková, 2007).....	23
Tabulka 6 - Typy grafů.....	24
Tabulka 7 - Rovinný a prostorový graf projektu vnitřní navigace	26
Tabulka 8 - Základní typy algoritmů pro hledání nejkratší cesty (MyRouteOnline, n. d.)	27
Tabulka 9 - Fáze projektu.....	37
Tabulka 10 - Kategorizace fází projektu.....	39
Tabulka 11 - Požadavky projektu vnitřní navigace	45
Tabulka 12 - Plán činností na projektu.....	49
Tabulka 13 - Seznam vstupních dat	54
Tabulka 14 - Naplnění požadavků projektu	68

9 Seznam příloh

Počítačový kód, první verze vnitřní navigace JavaScript:

```
var geoserverUrl ='http://127.0.0.1:8080/geoserver'
var selectedPoint = null;

var source = null;
var target = null;

// Inicializace mapy - Leaflet
var map = L.map("map", {
  center: [50.1294445275036, 14.490294284731982],
  zoom: 16 //set the zoom level
});

//Přidání podkladové mapy Openstreetmap Leaflet
var OpenStreetMap = L.tileLayer(
  "http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png",
  {
    maxZoom: 19,
    attribution:
      '&copy; <a
href="http://www.openstreetmap.org/copyright">OpenStreetMap</a>'
  }
).addTo(map);

// Nultá hodnota při prvním zobrazení
var pathLayer = L.geoJSON(null);

// Uchycení pro startovní bod
var sourceMarker = L.marker([50.1291211173488, 14.48969006538391], {
  draggable: true
})
.on("dragend", function(e) {
  selectedPoint = e.target.getLatLng();
  getVertex(selectedPoint);
  getRoute();
})
.addTo(map);

// Uchycení pro koncový bod
var targetMarker = L.marker([50.12903969, 14.49011935], {
  draggable: true
```

```

    })
    .on("dragend", function(e) {
        selectedPoint = e.target.getLatLng();
        getVertex(selectedPoint);
        getRoute();
    })
    .addTo(map);

// Funkce pro nalezení nejbližšího bodu
function getVertex(selectedPoint) {
    var url =
`${geoserverUrl}/wfs?service=WFS&version=1.0.0&request=GetFeature&typeName=routing:Nearest_vartex&outputformat=application/json&viewparams=x:${selectedPoint.lng};y:${selectedPoint.lat}`;
    $.ajax({
        url: url,
        async: false,
        success: function(data) {
            loadVertex(
                data,
                selectedPoint.toString() === sourceMarker.getLatLng().toString()
            );
        }
    });
}

// Funkce pro update koncových bodů
function loadVertex(response, isSource) {
    var features = response.features;
    map.removeLayer(pathLayer);
    if (isSource) {
        source = features[0].properties.id;
    } else {
        target = features[0].properties.id;
    }
}

// Funkce pro získání nejkratší cesty z GEOservru
function getRoute() {
    var url =
`${geoserverUrl}/wfs?service=WFS&version=1.0.0&request=GetFeature&typeName=routing:shortest_path&outputformat=application/json&viewparams=source:${source};target:${target}`;

    $.getJSON(url, function(data) {

```

```
        map.removeLayer(pathLayer);
        pathLayer = L.geoJSON(data);
        map.addLayer(pathLayer);
    });
}
getVertex(sourceMarker.getLatLng());
getVertex(targetMarker.getLatLng());
getRoute();
```

Počítačový kód, část HTML:

```
<head>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="initial-scale=1,user-scalable=no,maximum-
scale=1,width=device-width">
  <meta name="mobile-web-app-capable" content="yes">
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="theme-color" content="#000000">
  <meta name="description" content="">
  <meta name="author" content="">
  <title> Routing App</title>

  <!-- link the css files -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.0.3/leaflet.css">
  <link rel="stylesheet" href="css/app.css">
</head>
<body>
  <div class="navbar navbar-fixed-top" role="navigation">
    <div class="container-fluid">
      <div class="navbar-header">
        <div class="navbar-icon-container">
          <a href="#" class="navbar-icon pull-right visible-xs" id="nav-btn">
            <i class="fa fa-bars fa-lg white"></i>
          </a>
        </div>
        <a class="navbar-brand" href="#"> Routing App</a>
      </div>
    </div>
  </div>
  <div id="container">
    <!-- the map will go inside this div below -->
    <div id="map"></div>
    <!-- link to the js files -->
    <script src="https://code.jquery.com/jquery-3.2.0.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/leaflet/1.0.3/leaflet.js"></script>
    <script src="js/app.js"></script>
  </div>
</body>

</html>
```