

České vysoké učení technické v Praze  
Fakulta elektrotechnická



Bakalářská práce

**Návrh uživatelského rozhraní pro snadnou tvorbu navigačního  
popisu budov pro laiky**

*Petra Marešová*

Vedoucí práce: Ing. Jan Vystrčil

Studijní program: Softwarové technologie a management strukturovaný  
bakalářský

Obor: Web a multimédia

květen 2010



### **Poděkování**

Chtěla bych poděkovat svému vedoucímu ing. Janu Vystrčilovi za jeho obětavou pomoc a odborné vedení. Dále bych chtěla poděkovat všem kolegům dobrovolníkům a dalším účastníkům testů, kteří mi ochotně věnovali svůj čas a podíleli se tak na vývoji této práce.



### **Prohlášení**

Prohlašuji, že jsem svou bakalářskou práci vypracovala samostatně a použila jsem pouze podklady uvedené v příloženém seznamu.

Nemám závažný důvod proti užití 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 25.5.2010 .....



## **Abstract**

Improving of conditions for independent mobility of eye-handicaped people is still topical problem. This work deals with designing of user interface, which helps users to create descriptions of building for NaviTerier. System Naviterier is navigation for visually impaired people in interiors using a cell phone. Requirements for buildings structure and application itself are analyzed in first part of the thesis.

During designing ephasis is placed on simplicity and intuitiveness of user interface. Few prototypes are designed and then tested. Final project is verified with users in usability tests. At the conclusion results of testing are evaluated and direction of further development is recommended.

## **Abstrakt**

Zlepšování podmínek pro samostatný pohyb zrakově handicapovaných osob je stále aktuální problém. Tato práce se zabývá návrhem uživatelského rozhraní aplikace, pomocí které mohou uživatelé tvořit popisy budov pro NaviTerier. Systém NaviTerier je navigace zrakově postižených osob ve vnitřních prostorech pomocí mobilního telefonu. V první části práce jsou zkoumány požadavky na strukturu popisu a na aplikaci samotnou. Při návrhu se klade velký důraz na jednoduchost a intuitivnost rozhraní. Je postupně navrženo několik prototypů a ty jsou otestovány. Výsledný návrh je ověřen s uživateli metodou testů použitelnosti. V závěru práce jsou vyhodnoceny výsledky testování a je doporučen směr dalšího vývoje.





## Obsah

1.	Úvod .....	1
1.1.	Projekt NaviTerier .....	1
1.2.	Tvorba mapových podkladů pro NaviTerier .....	1
2.	Popis řešené problematiky .....	3
2.1.	Problém rozdílných požadavků .....	3
2.2.	Analýza cílové skupiny uživatelů .....	4
2.2.1.	Technická zdatnost .....	4
2.2.2.	Srozumění s prostorovou orientací nevidomých .....	4
2.2.3.	Zodpovědnost .....	4
2.3.	Požadavky na systém .....	4
2.3.1.	Intuitivnost a jednoduchost .....	5
2.3.2.	Nápověda .....	5
2.3.3.	Popis mapy jako struktury elementárních objektů .....	5
2.3.4.	Částečně předdefinovaný formulář na popis elementu .....	5
2.3.5.	Kompromis mezi generováním a slovním popisem .....	6
2.3.6.	Zpětná editovatelnost mapy, uzpůsobení dalším uživatelům .....	6
2.3.7.	Grafické zobrazení objektů a jejich struktury .....	6
2.3.8.	Možnost tzv. „krokování“ .....	6
2.3.9.	Členění na vrstvy podle podlaží .....	6
3.	Popis struktury bakalářské práce .....	8
4.	Analýza a diskuze variant návrhu .....	9
4.1.	Struktura popisu zobrazované mapy .....	9
4.2.	Elementy a Vstupy .....	10
4.3.	Panely a jejich funkce .....	11
4.3.1.	Pane menu .....	11
4.3.2.	Panel nápověda .....	11
4.3.3.	Panel plátno .....	11
4.3.4.	Panel vzorová mapa .....	12
4.3.5.	Panel elementy .....	12
4.3.6.	Panel element .....	12
4.3.7.	Panel makro .....	12
4.3.8.	Panel podlaží .....	12
4.4.	Papírový prototyp .....	13
4.5.	Low fidelity prototyp .....	13
4.6.	Formulář pro popis elementů .....	14
4.6.1.	Data společná pro všechny druhy elementu .....	14
4.6.2.	Obtížnost průchodu .....	15
4.6.3.	Nebezpečí .....	15
4.6.4.	Způsob popisu průchodu elementem .....	16
5.	Výsledný návrh .....	18
5.1.	Hlavní okno .....	18
5.1.1.	Panel menu .....	18
5.1.2.	Panel elementy a Panel makra .....	18
5.1.3.	panel plátno .....	19
5.1.4.	Panel nápověda .....	20

5.1.5.	Panel formulář .....	21
5.2.	Formulář pro popis chodby .....	21
5.2.1.	Název elementu .....	21
5.2.2.	Přibližné rozměry .....	21
5.2.3.	Počet vstupů .....	22
5.2.4.	Povrch podlahy .....	22
5.2.5.	Doplňující informace .....	22
5.2.6.	Index obtížnosti průchodu chodbou.....	23
5.2.7.	Hned při vstupu hrozí nebezpečí .....	24
5.2.8.	Popisy průchodů chodbou.....	24
5.3.	Formulář pro popis místnosti .....	25
5.4.	Formulář pro popis schodiště .....	26
5.4.1.	Název elementu a délka .....	26
5.4.2.	Typ schodiště a sklon .....	26
5.4.3.	Povrch a doplňující informace .....	27
5.4.4.	Průchod schodištěm a popis průchodu .....	27
5.4.5.	Schodiště navazuje na vstup .....	27
5.5.	Formulář pro popis dveří .....	28
5.5.1.	Všeobecný popis .....	28
5.5.2.	Panel průchod dveřmi .....	29
5.5.3.	Popis průchodu dveřmi .....	29
5.6.	Formulář pro popis výtahu .....	30
5.6.1.	Název elementu .....	30
5.6.2.	Tlačítko na vyvolání výtahu a otvírání dveří.....	30
5.6.3.	Ovládání výtahu a doplňující informace .....	30
5.6.4.	Výtah spojuje tato patra.....	31
5.7.	Závěrečné zpracování dat algoritmem.....	32
5.8.	Menu u elementu vyvolané stiskem pravého tlačítka myši.....	32
5.9.	Klávesové zkratky.....	32
6.	Tvorba a implementace prototypů .....	33
6.1.	High fidelity prototyp.....	33
6.1.1.	Formuláře.....	33
6.1.2.	Panel plátno.....	33
6.1.3.	Omezení funkcionality prototypu.....	33
7.	Ověření návrhu uživatelského rozhraní .....	34
7.1.	Výsledky ověření low fidelity prototypu .....	34
7.2.	Testování high fidelity prototypu .....	34
7.2.1.	Metoda testování .....	34
7.2.2.	Tvorba prototypu.....	34
7.2.3.	Výběr účastníků testu .....	34
7.2.4.	Materiály pro testování .....	35
7.2.5.	Popisovaný objekt .....	35
7.2.6.	Zadání testu.....	36
7.2.7.	Průběh testování.....	36
7.3.	Výsledky testů .....	37
7.3.1.	Časové výsledky testu .....	37

7.3.2.	Problémy odhalené testováním .....	38
7.3.3.	Připomínky k formulářům elementů .....	39
7.3.4.	Zajímavé postřehy uživatelů.....	39
8.	Závěr.....	40
8.1.1.	Zhodnocení .....	40
8.1.2.	Přístupnost aplikace přímo z internetu .....	40
8.1.3.	Další testování .....	40
8.1.4.	Datový výstup .....	41
8.1.5.	Propagace NaviTerieru .....	41
9.	Reference .....	42
A	Seznam použitých zkratk.....	44
B	Varianty rozložení panelů .....	45
C	Low fidelity prototyp .....	46
D	High fidelity prototyp .....	47
E	zadání testu s uživateli .....	48
F	Obsah přiloženého CD .....	52



## Seznam obrázků

Obr. 1 Vrstvy podlaží .....	7
Obr. 2 Stávající spojení elementů .....	9
Obr. 3 Výběr podle tvaru .....	9
Obr. 4 Elementy a vstupy.....	10
Obr. 5 Spojování vstupů.....	11
Obr. 6 Varianta A.....	13
Obr. 7 Vybrané rozložení panelů.....	14
Obr. 8 Nebezpečí po vstupu.....	16
Obr. 9 Varianty průchodu v elementu .....	17
Obr. 10 Volba podlaží.....	18
Obr. 11 Volba elementu .....	19
Obr. 12 Panel makro a krokování.....	19
Obr. 13 Rozměry elementu .....	22
Obr. 14 Manipulace se vstupem .....	22
Obr. 15 Formulář popisu chodby .....	23
Obr. 16 Panel obtížnosti průchodu.....	24
Obr. 17 Panel popisy průchodů .....	25
Obr. 18 Vlastnosti schodiště.....	26
Obr. 19 Panel průchodu schodištěm.....	27
Obr. 20 Panel pro napojení schodiště .....	28
Obr. 21 Formulář pro popis dveří .....	29
Obr. 22 Panel popisu průchodů .....	30
Obr. 23 Část panelu pro popis výtahu .....	31
Obr. 24 Panel propojení pater pomocí výtahu.....	32
Obr. 25 Model budovy .....	36
Obr. 26 Graf časového průběhu testování.....	37
Obr. 28 Varianta B .....	45
Obr. 29 Varianta C .....	45



## 1. Úvod

Možnost samostatného pohybu je jedním ze základních požadavků na kvalitní a plnohodnotný život. Osoby s poruchou zraku mají v tomto ohledu velmi ztížené podmínky. I přes to, jak moderní technologie usnadňují běžné lidské činnosti, problém samostatného pohybu a snadné orientace nevidomých v prostoru zůstává stále aktuální. Není zatím vyřešen do takové míry, aby se při orientaci bylo možné plně spolehnout na automatický navigační systém. V exteriérech se nabízí řešení pomocí automatických navigací s použitím technologie GPS, v interiérech je tedy zapotřebí náhradního řešení. Je těžko pochopitelné, že nejsou legislativně ošetřeny pomůcky pro navigaci nevidomých ve veřejných institucích, stanicích MHD a jiných dopravních prostředků, stejně jako jsou ošetřeny bezbariérové přístupy, určené vyhláškou 398/2009 Sb..

Nabízí se tu mnoho variant v podobě navigačních map a popisných tabulek v braillově písmu, interaktivních informačních tabulí a lokálních telekomunikačních zařízení, umožňujících nevidomému spojení s právě přítomným úředníkem, vrátným, či jinou pověřenou osobou v budově.

Zlepšením životní úrovně zrakově handicapovaných se v současné době zabývá mnoho center a společností, zaměřených na vývoj a distribuci asistivních technologií. Zapojení laických dobrovolníků a zainteresovaných osob do této problematiky by mohlo znamenat nezanedbatelné zlepšení.

### 1.1. Projekt NaviTerier

Navigací pro zrakově handicapované (dále jen ZP) zaměřenou na pohyb v interiérech se už řadu let zabývá na katedře počítačů fakulty elektrotechnické ČVUT celá skupina vývojářů. Vzniklo již několik prací souvisejících s touto problematikou. Jednou z dalších je projekt o navigaci nevidomých uvnitř budov [1]. Na základě testovaných návrhů způsobu navigace je zde upřednostňována varianta popisu trasy jako souvislého řetězce informací o procházených objektech od vstupu až k cíli. Na tento výzkum navázala diplomová práce „Uživatelské rozhraní pro navigaci nevidomých osob v budovách pomocí mobilního telefonu“ [2], kde je navržena a testována aplikace NaviTerier, která pomocí převodu textu z displeje mobilního telefonu do zvukové podoby navádí ZP na jím zvolené místo v budově. Jednou z výhod aplikace NaviTerier je její dostupnost. Pro její spuštění je zapotřebí pouze obyčejný mobilní telefon s možností instalace aplikace v programovacím jazyce Java a syntézy textu z obrazovky do hlasové podoby. Při vývoji proběhlo několik testování s uživateli. Z výsledků testů vzešly návrhy a doporučení, které budu využívat ve své práci.

### 1.2. Tvorba mapových podkladů pro NaviTerier

Pro hledání a popis cesty potřebuje NaviTerier [2] strukturovaná data mapující budovu, která by bylo možné zpracovávat podle potřeby různými algoritmy, pro hledání konkrétních míst a nejefektivnějších cest. Vytváření těchto popisů by zabralo stovky hodin. Proto je vhodné navrhnout aplikaci, která by pomocí

jednoduchého a intuitivního uživatelského rozhraní umožnila snadnější tvorbu popisu budovy a sama by z uživatelem zadaných dat generovala popis pro aplikaci NaviTerier (například v XML nebo jiném jazyce pro popis strukturovaných dat). Na základě výsledků testování projektu NaviTerier bylo popsáno několik doporučení pro návrh grafického rozhraní této aplikace.

*„Pro usnadnění tvorby popisů je vhodné vytvořit aplikaci zajišťující podporu tvorby popisů budov. Hlavní části aplikace by měl tvořit editor segmentů a editor budovy. Editor segmentů slouží pro definování typu segmentu, počtu existujících propojení a jejich rozmístění. Dále se v něm vytvářejí popisy jednotlivých segmentů. Při vytváření popisu segmentů je možné využít symetrie některých objektů pro zjednodušení zadávání popisů. Řadu popisů je také možné pouze kopírovat, a pak upravovat detaily. Pro zlepšení orientace mezi jednotlivými segmenty je možné ke každému segmentu přidat výřez z plánu budovy zachycující reálnou situaci. Funkce dynamického hledání trasy (routing) bude vyžadovat ještě údaj o délce každého segmentu v metrech. Je možné přidat i speciální metriku hodnotící náročnost průchodu segmentem. Tím je možné zohlednit delší trasu, která je na průchod jednodušší, oproti nejkratší trase. Editor budovy zajišťuje zadávání a správu korespondencí propojení a vytváří z popsaných segmentů model budovy. Editor také kontroluje, zda jsou všechna připojení napojena na další segmenty a žádné připojení nekončí ve volném prostoru.“ [2]*



## 2. Popis řešené problematiky

Účelem této práce je navrhnout a otestovat grafické rozhraní aplikace, která by umožnila snadnější tvorbu popisů budov pro navigační systém NaviTerier [2]. Navržené rozhraní by mělo být co nejjednodušší, intuitivní a snadno zapamatovatelné, aby bylo možné zapojit do tvorby map i laické dobrovolníky.

Tvorba popisů budov pro samostatný pohyb ZP je citlivý problém. ZP se v tomto případě musí spolehnout na navigační technologii a popis budovy, který je z části vygenerován. Pokud dojde k chybnému navigování během samostatného pohybu ZP, bude v lepším případě ohrožena jeho důvěra k navigační aplikaci a v horším případě jeho zdraví. Proto bude navržený prototyp pro tvorbu popisů budov testován a výsledky budou doporučeny k dalšímu vývoji.

### 2.1. Problém rozdílných požadavků

Potřeby ZP a jejich požadavky na navigaci jsou velmi různorodé. První příčinou je rozmanitost vad zraku a míry jeho poškození.

Sjednocená organizace nevidomých a slabozrakých SONS [4] rozděluje vady zraku do tří základních skupin:

- poruchy ostrosti vidění
- výpadky zorného pole
- poruchy barvocitu

Podle Světové zdravotnické organizace v ČR [5] můžeme definovat 5 úrovní ztráty zraku:

- střední a silná slabozrakost
- těžce slabý zrak
- praktická nevidomost
- úplná nevidomost

V neposlední řadě bychom měli brát v úvahu i potřeby uživatele vycházející z jeho zvyků nebo schopností [8]. Někteří ZP žijí se svým handicapem už od narození, mají tedy lépe rozvinutou schopnost orientace bez obrazových vjemů než osoby, které přišly o zrak v průběhu života a samostatnému pohybu se učí teprve krátkou dobu. Je také možné, že ZP, který vlastní asistenčního psa, bude potřebovat o něco méně informací, než nevidomý, který používá hůl a například při nástupu do výtahu se musí spolehnout pouze na zvuková znamení a sám na sebe.

Uživatelé programu tvoří většinou popis pro konkrétní osoby nebo ZP, se kterými jsou ve styku. Jedná se o osoby blízké, uživatelé tedy znají orientační návyky a potřeby nevidomých nebo se jich na konkrétní problémy mohou zeptat. I přesto, že jsou uživateli právě osoby blízké, je nutné, aby byl popis

snadno upravitelný a umožňoval jiným uživatelům, aby mohli popis jednoduše uzpůsobit pro potřeby dalších ZP.

## **2.2. Analýza cílové skupiny uživatelů**

### **2.2.1. Technická zdatnost**

Pro práci s aplikací na tvorbu map potřebuje uživatel pouze základní dovednosti práce s počítačem na úrovni, která je v současné době popsána v osnovách předmětu informatika pro šestou až sedmou třídu druhého stupně základních škol. Jedná se o práci s hardwarovým zařízením počítače jako je myš a klávesnice, základní orientaci v práci se soubory a složkami a ovládání běžných uživatelských aplikací.

Úroveň vzdělání uživatelů není specifikována. Nepředpokládáme předchozí zkušenosti s grafickými programy, pouze základní, výše uvedené dovednosti.

### **2.2.2. Srozumění s prostorovou orientací nevidomých**

Cílovou skupinu tvoří lidé, kteří jsou zapojeni do problematiky zlepšení podmínek života ZP a jejich samostatného pohybu. Jedná se především osoby, které jsou s nimi v kontaktu. To mohou být členové rodiny, kamarádi, spolužáci, ale také lidé, kteří jsou aktivně zapojeni do řešení této problematiky jako dobrovolníci. Členové rodiny a kamarádi dokáží často vyzorovat orientační návyky ZP z chování v běžném životě, přesto ale není zaručeno, že dokáží vyhovět veškerým jejich požadavkům. Další část této skupiny tvoří zaškolení dobrovolníci, jejichž znalosti jsou často získané na školeních zásad pomoci ZP a aktivní dobrovolnickou činností v oblasti asistencí. Vzhledem k nesourodosti cílové skupiny uživatelů tedy nemůžeme počítat s vyšší úrovní znalostí z oblasti prostorové orientace ZP. Proto je nutné vytvořit materiálovou dokumentaci dostatečně obsáhlou a přehlednou, která by poskytla nezbytné informace o problematice nevidomých.

### **2.2.3. Zodpovědnost**

Závěrečným aspektem, který musíme brát v úvahu, je zodpovědnost a schopnost tvořit popisy budov pro samostatnou orientaci ZP. V tomto případě nelze přesně specifikovat věk ani vzdělání jedince, protože tato schopnost záleží především na mentální vyspělosti, intuici a aktivním zájmu o problematiku.

Případné nedostatky popisů budov, které by vznikly právě z absence těchto schopností by mohl vyřešit systém zpětné kontroly, kterým by uživatelem vytvořené popisy prošly.

## **2.3. Požadavky na systém**

Při analýze požadavků budu vycházet z výsledků testů použitelnosti aplikace NaviTerier [2] a z mé semestrální práce [3]. Ta se zabývala návrhem a testová-

ním uživatelského rozhraní aplikace pro tvorbu popisů budov určených pro NaviTerier.

### 2.3.1. Intuitivnost a jednoduchost

Grafické rozhraní aplikace by mělo být především jednoduché a intuitivní. Při jeho návrhu je třeba se vyhnout nepřehledně členěným panelům a složitým strukturám víceúrovňových přístupů k funkcím, které uživatele pouze zdržují. Samotná koncepce okna by měla být dobře zapamatovatelná, intuitivní a obsahovat pouze nejnütnější tlačítka funkcí logicky seskupená do panelů. S přibývajícím počtem tlačítek roste i nepřehlednost, prodlužuje se čas pro hledání funkce a snižuje se efektivita práce uživatele, který potřebuje více času, aby se v uživatelském rozhraní aplikace zorientoval.

Pojmenování tlačítek by mělo být jednoznačné a dostatečně charakterizovat jejich funkcionalitu. Často používané funkce by měly být dobře graficky zviditelněné a snadno přístupné. Mezi postupy pro vyvolání podobných akcí by měla fungovat analogie, aby si je uživatel mohl odvodit a lépe zapamatovat.

### 2.3.2. Náповěda

Pro všechny uživatele je důležitá nápověda. Tvorbě popisu budovy by mělo předcházet krátké školení nebo prostudování kvalitně zpracovaných návodů a vzorových příkladů, které musí nápověda aplikace obsahovat. Samotné tlačítko nápovědy by mělo být viditelné a přístupné ze všech stavů grafického rozhraní.

V ideálním případě by mohla nápověda obsahovat interaktivní část, která by reagovala na to, jaké akce uživatel zrovna používá a napovídat mu potencionálně další kroky.

Protože se předpokládá, že nápověda této aplikace bude velmi obsáhlá, mohla by se ke komplexním funkcím aplikace umístit kontextová nápověda spustitelná speciálním tlačítkem, které by v případě pochybností zobrazilo pouze informace o dané funkci.

### 2.3.3. Popis mapy jako struktury elementárních objektů

Myšlenka popisu budovy jako struktury propojených segmentů byla v práci o aplikaci NaviTerier [2] popsána takto.

*„Nasazení navigačního systému do praxe s sebou přináší nutnost vytváření popisů ve formě propojené struktury segmentů. Tato struktura musí umožňovat dynamické generování jednotlivých tras z různých startovních bodů do různých cílových bodů. Tvorba celého popisu se v takovém případě skládá ze dvou částí. První část spočívá ve vytvoření popisu jednotlivých segmentů a druhá část představuje spojení segmentů do jednoho celku podle reálné polohy segmentů.“ [2]*

### 2.3.4. Částečně předdefinovaný formulář na popis elementu

Pro popis segmentů by měl být použit speciální formulář, který by předdefinoval uživateli, které vlastnosti popsat a jakým způsobem. Zároveň by mu ulehčil

práci s vypisováním dlouhých řetězců informací díky předem definovanému výběru typických vlastností, ve kterém by uživatel pouze označil správnou volbu. Formulář by zároveň zajišťoval validaci zadaných dat a jejich přehlednost při následných úpravách. Z těchto informací by měl jít vygenerovat výstup ve tvaru „POPIS-AKCE“, který by pro ZP daný objekt nejdříve popsal a pak by mu sdělil následující akci, kterou musí vykonat, aby se dostal na popisované trase do dalšího objektu.

### **2.3.5. Kompromis mezi generováním a slovním popisem**

Aplikace by samozřejmě měla uživateli práci podstatně zjednodušit. Je tedy vhodné, aby se generoval co největší počet informací automaticky? Čeština je jazyk s poměrně složitou gramatikou, obsahuje velké množství pravidel, která se řídí spíše jazykovým citem než logikou. Syntaktická nesprávnost generovaného textu je v případě používání systému nevidomým člověkem naprosto nepřijatelná. Proto je třeba zvolit vhodný kompromis mezi automaticky vygenerovanými částmi a manuálním popisem elementu, který by neohrožoval jazykovou správnost výsledného popisu.

### **2.3.6. Zpětná editovatelnost mapy, uzpůsobení dalším uživatelům**

I přesto, že je tato aplikace primárně určena uživatelům, kteří budou tvořit mapu pro konkrétní blízkou osobu, není žádoucí, aby byl výsledkem mnohahodinové práce popis budovy vhodný pouze pro jednu zrakově handicapovanou osobu. Program by měl umožnit využití již vytvořených popisů a jejich částí, které by se daly snadno přizpůsobit potřebám jiných ZP.

### **2.3.7. Grafické zobrazení objektů a jejich struktury**

Nezbytnou součástí pro lepší orientaci a představu o podobě popisu budovy je grafické zobrazování postupně vytvářené struktury. Tato grafická podoba popisu by nezobrazovala přesné tvary půdorysu budov, ale pouze přibližné rozměry, rozmístění a propojení jednotlivých segmentů. Další důležitou funkcí vizualizace popisu by bylo vytvoření zpětné vazby mezi uživatelem a jím zvolenou akcí, která by se projevila změnou grafické podoby segmentu, popřípadě jeho částí.

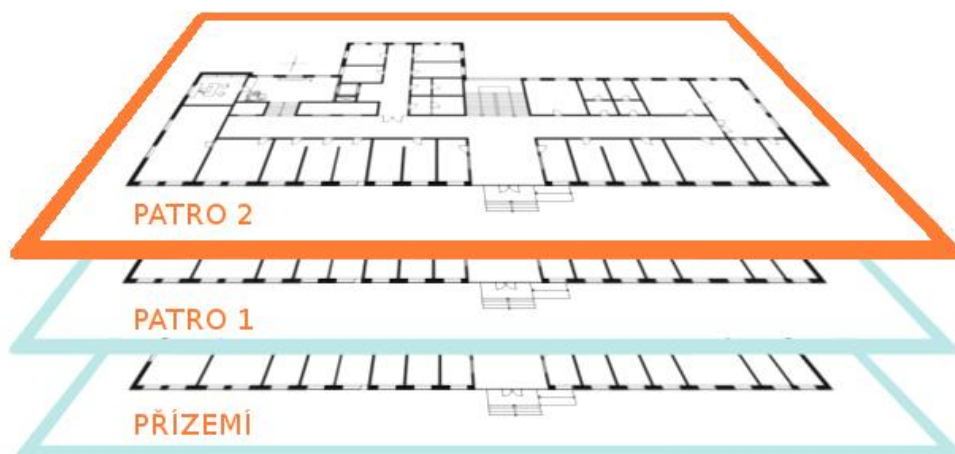
### **2.3.8. Možnost tzv. „krokování“**

Uživatel se často dostane do stavu, kdy se v důsledku chybné volby potřebuje vrátit i o několik akcí zpět. Pro tento případ by měla aplikace archivovat posloupnost určitého počtu uživatelem zadaných akcí. Využitím funkce „krokování“ by se uživatel mohl vrátit zpět až do stavu poslední archivované akce a opět postoupit vpřed.

### **2.3.9. Členění na vrstvy podle podlaží**

Podlaží by měla být tvořena zvlášť jako samostatné celky (Obr. 1)., každé ve vlastní vrstvě. Mezi vrstvami by se dalo přepínat a při vytvoření nového pod-

laží by se automaticky vytvořila nová vrstva. To by uživateli poskytlo lepší představu o prostorovém uspořádání popisované budovy.  
Podlaží by se dala propojovat pomocí schodů a výtahu. Řešení tohoto problému je popsáno v kapitole 5.4.5, která se zabývá propojením podlaží pomocí schodiště a v kapitole 5.6.4 o propojení podlaží pomocí výtahu.



Obr. 1 Vrstvy podlaží

### 3. Popis struktury bakalářské práce

První část mé práce se zabývá nástinem problematiky tvorby popisů budov pro ZP určených ke zpracování navigací NaviTerier [2].

V rané fázi návrhů budu navazovat na svůj semestrální projekt „Uživatelské rozhraní pro autoring tool aplikace NaviTerier“ [3]. V rámci tohoto projektu jsem provedla první návrhy uživatelského rozhraní, které jsem využila při tvorbě low fidelity prototypu popsaného ve čtvrté kapitole.

Po zvážení různých variant návrhu uživatelského rozhraní je detailně rozpracován high fidelity prototyp a jsou konkrétně popsány jednotlivé sektory rozhraní a jejich funkcionalita.

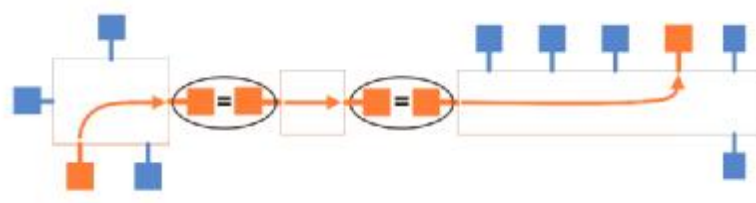
Šestá kapitola se pak věnuje především testování implementovaného prototypu a diskuzi výsledků testování s uživateli.

V závěru se zabývám vyhodnocením této práce a následným doporučením pro další vývoj.

## 4. Analýza a diskuze variant návrhu

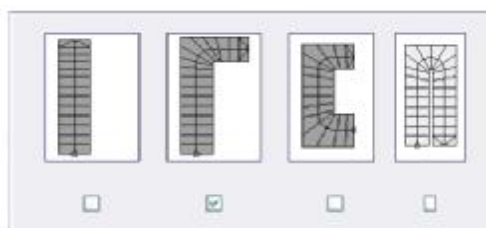
### 4.1. Struktura popisu zobrazované mapy

První varianta vychází z návrhů publikovaných ve výsledcích testování aplikace NaviTerier [2], kde je navržena možnost skládání popisu z jednotlivých segmentů se vstupními body, jejichž spojením vznikne propojená struktura segmentů vytvářející celkový popis budovy. (Obr. 1).



Obr. 2 Stávající spojení elementů

První varianta tedy funguje na principu skládání popisu budovy z jednoduchých tvarů (Obr. 2). Nabízí se ale i jiná řešení. Jako druhou variantu bychom mohli použít výběr objektu ze seznamu základních tvarů. Například, pokud by se přidávalo do popisu schodiště, aplikace by nabídla základní předdefinované tvary schodišť (Obr. 3). Stejně tak v případě chodby by byly nabídnuty různé tvarové typy, ze kterých by si mohl uživatel vybrat přímou chodbu, chodbu ve tvaru L, T nebo chodbu s rozcestím a jiné.



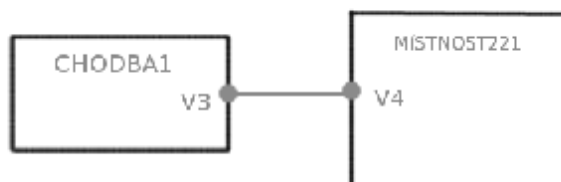
Obr. 3 Výběr podle tvaru

Budu pokračovat v rozvíjení první varianty, protože cílem není vytvoření co nejněvhodnějšího grafického zobrazení budovy, ale kvalitní popis budovy pro potřeby ZP. Grafické zobrazení jednotlivých objektů v aplikaci má funkci pouze orientační a slouží tvůrcům popisu pro zpřehlednění popisované reality. Autor popisu si musí uvědomit, že ZP bude mít o elementu pouze ty informace, které mu budou slovně popsány.

## 4.2. Elementy a Vstupy

Popis budovy bude tvořen ze dvou základních prvků. Prvním bude elementární objekt, který budu zatím nazývat „*element*“. Dalším prvkem budou průchozí body na obvodu elementu, které označím jako „*vstupy*“. Vstupy bude možné přemostit, a tak propojit jednotlivé elementy.

Názvy jsou pouze dočasné a mohou se stát předmětem diskuze v dotaznících při testování s uživateli.

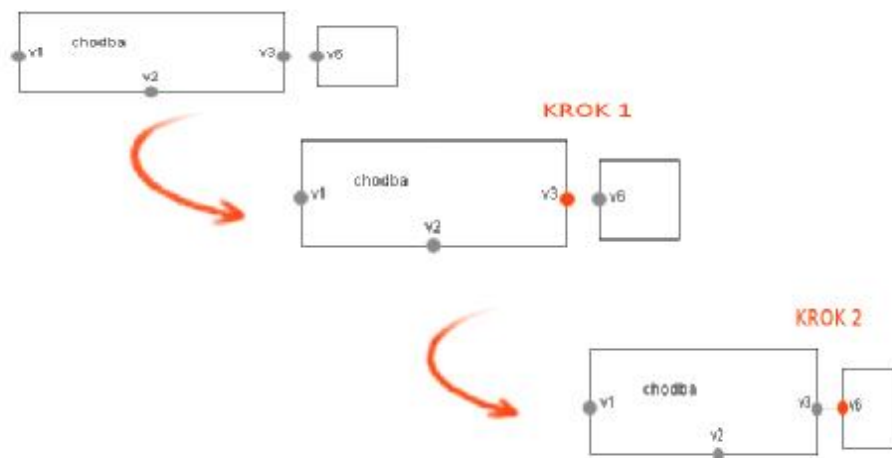


Obr. 4 Elementy a vstupy

V budovách potřebujeme evidovat pět základních typů elementů a to chodby, místnosti, schodiště, výtahy a dveře. Existuje i varianta zpracovat dveře jako součást chodby a místnosti. Budu se jí dále zabývat v části tvorby prototypů v kapitole 5.5.

Elementy budu zobrazovat jako obdélníky s rozměry proporcionálně podobnými skutečnosti. Při zobrazení vstupů navrhuji malou změnu oproti návrhu z obrázku (Obr. 2). Protože obrázek vyplněného čtverce s úsečkou zabírá hodně místa, zvolím zobrazení pomocí vyplněného kruhu, umístěného přímo na obvodu elementu. Při spojení vstupů potom bude uživatel pouze klikat myší na tyto body a vytvořené spojení se mu zobrazí jako úsečka z jednoho vstupu do druhého. Se vstupy bude možné hýbat po délce strany elementu, v případě spojovací úsečky budou oba konce umístitelné pouze do prostoru jednotlivých vstupů. Pokud bude označen první vstup, bude se od něj ke kurzoru průběžně vykreslovat úsečka, dokud nebude označen druhý vstup nebo zrušena akce.





Obr. 5 Spojování vstupů

### 4.3. Panely a jejich funkce

V aplikaci budeme potřebovat několik panelů. Okno by mělo obsahovat standardní menu pro práci s projektem, panel pro zobrazení mapy, panel pro přidání elementu a formulář pro popis jeho vlastností. Velmi důležitou součástí bude samozřejmě nápověda.

#### 4.3.1. Panel menu

Panel menu je standardní panel umístěný u horního okraje grafického rozhraní, který obsahuje alespoň tyto volby:

- nový projekt
- otevřít
- uložit

#### 4.3.2. Panel nápověda

Panel nápověda je panel, který bude rozdělen na dvě části. V jedné bude tlačítko s nápisem „nápověda“ a v druhé lišta s interaktivní nápovědou, která bude reagovat na uživatelem zvolené akce.

#### 4.3.3. Panel plátno

Panel plátno slouží k zobrazení a práci s grafickými symboly elementů. Všechny zobrazené elementy by měly být propojeny s údaji ve formuláři, aby se postupnou úpravou údajů měnily i jejich grafické symboly. Barevné zvýraznění elementů a vstupů by poskytovalo verifikaci pro volby vybrané ve formuláři.

#### 4.3.4. Panel vzorová mapa

Panel plátno procentuálně zaplní největší část okna, proto by se dala jeho část využít k zobrazení libovolného obrazového podkladu (půdorys nebo fotografie), který by zachycoval skutečnou podobu budovy a podle kterého by se autor při tvorbě popisu budovy mohl řídit.

Panel vzorová mapa proto umožňuje rozdělit plátno na dvě části a v druhé zobrazit jakýkoliv obrázek s kompatibilním formátem. Výhodou je skutečnost, že uživatel nemusí přepínat mezi oknem pro zobrazení podkladů k práci a oknem aplikace. Pokud by nebyla zvolena akce zobrazení vzorové mapy, panel plátno by zůstal nerozdělen.

Další možností jak zobrazit vzorovou mapu by bylo promítnutí obrázku jako podkladu na celé plátno. Tato varianta by byla vhodná zejména pokud by se jednalo o půdorys budovy.

#### 4.3.5. Panel elementy

Panel elementy obsahuje tlačítko „přidat element“, kterým se otevírá výběr z pěti předdefinovaných typů elementu:

- chodba
- místnost
- schodiště
- dveře
- výtah

#### 4.3.6. Panel element

Panel element obsahuje formulářový editor vlastností a popisů průchodu každého elementu. Pro zjednodušení bude dále v BP nazýván jako panel formulář.

#### 4.3.7. Panel makro

V panelu makro jsou umístěna tlačítka pro práci s makrem. Panel obsahuje alespoň tyto volby:

- uložit makro
- otevřít makro

#### 4.3.8. Panel podlaží

Panel umožňuje přepínat mezi vrstvami jednotlivých podlaží. Mohl by obsahovat například tyto volby:

- stávající podlaží (např.: přízemí)
- stávající podlaží (např.: patro 1)
- stávající podlaží (např.: patro 2)
- přidat nové patro

Možnost "přidat nové patro" jsem zařadila k výběru konkrétních podlaží, dalo by se to vyřešit i samostatným tlačítkem. Volbu finální podoby nechám na výsledcích testování.

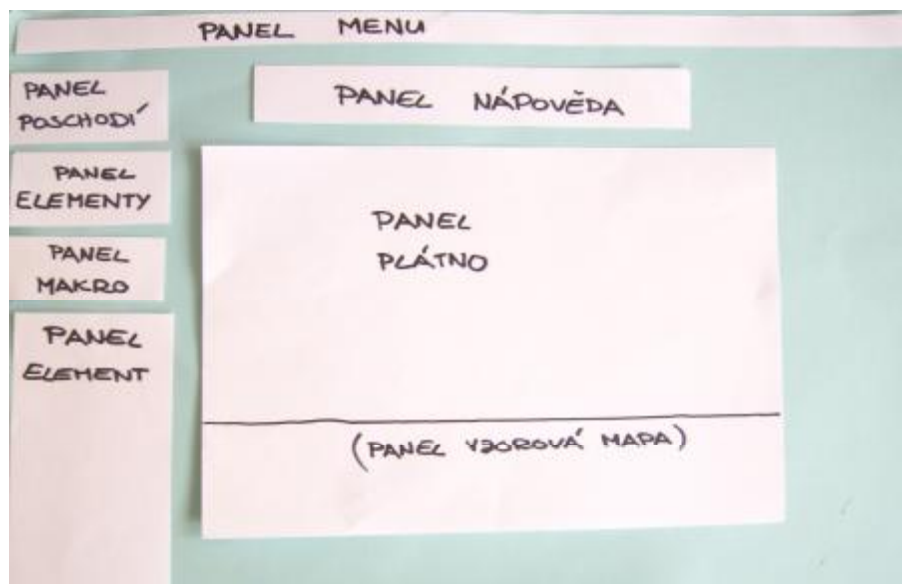
#### 4.4. Papírový prototyp

Za účelem návrhu rozložení panelů byl vytvořen jednoduchý papírový prototyp, který obsahuje varianty rozmístění panelů. Papírový prototyp se používá v rané fázi návrhu, jeho výhodou je jednoduchá a časově nenáročná tvorba.

Velikost jsem zvolila podle nejčastějšího poměru stran monitoru. Většina plochých monitorů "LCD" má poměr stran 5:4. Při volbě rozložení panelů hrálo hlavní roli zachování poměru stran panelu „plátno“.

Proto jsem vyloučila variantu B (Obr. 2728), která výrazně zmenšila jeho šířku. Varianty A (Obr. 6) a C (Obr. 2829) zachovávají rovnoměrné rozložení panelů v aplikaci a to i v případě, že se panel plátno vertikálně rozdělí zobrazením vzorové mapy.

Mezi nimi rozhodne až velikost panelu pro popis elementu. Prozatím zvolím variantu A, která má lepší poměr stran panelu „plátno“. V případě, že by bylo zapotřebí evidovat v popisu elementu větší množství informací, zvolím variantu C, která umožňuje využít pro panel „element“ celou výšku okna aplikace.

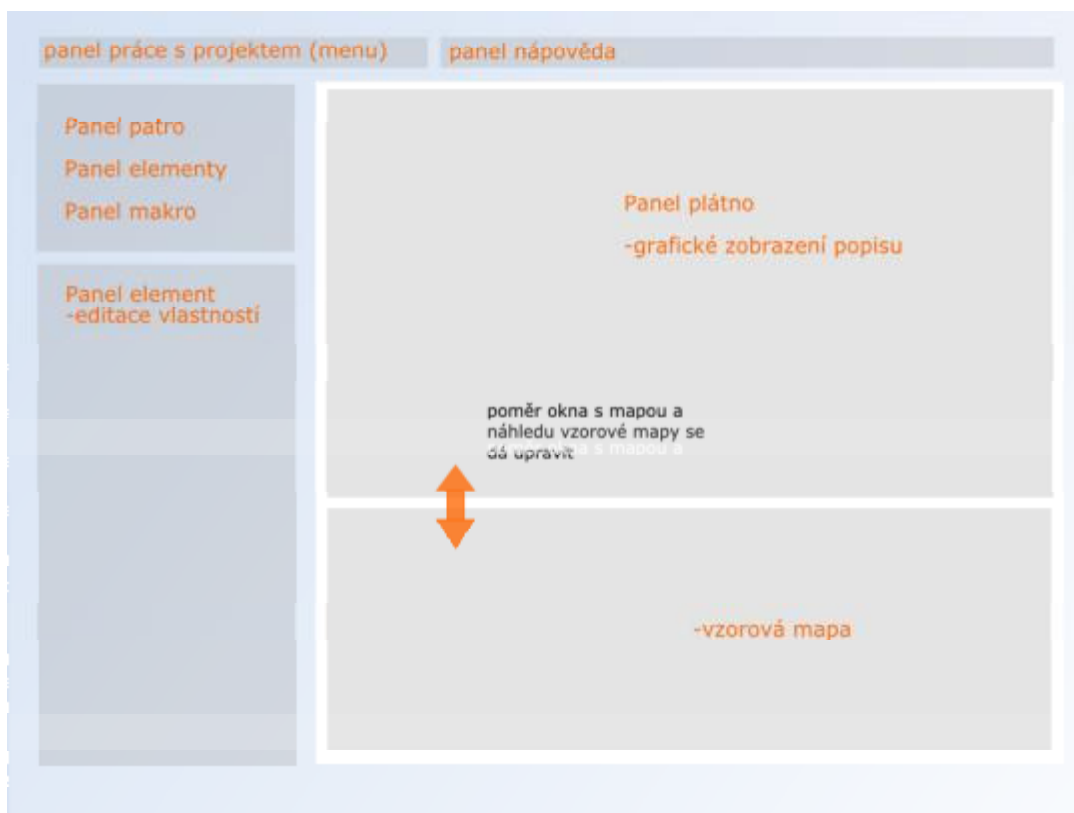


Obr. 6 Varianta A

#### 4.5. Low fidelity prototyp

Podle papírového prototypu jsem vytvořila grafický model se schématickým zakreslením panelů (Obr. 7) Na základě toho jsem vytvořila low fidelity prototyp,

kde je pevně dané rozmístění všech panelů a jejich částí. Je umístěn v příloze C. Vzhledem k tomu, že tento model neposkytuje zpětnou vazbu ani detailní vzhled formulářů pro popis všech pěti typů elementu, byl testován bez uživatelů metodou kognitivního průchodu. Výsledky testu jsou uvedeny v kapitole 7.1. Pro další testování funkcionality a použitelnosti návrhu s uživateli, jsem se rozhodla vytvořit věrohodnější prototyp. Proto se budu v následující části kapitoly věnovat diskuzi a výběru variant popisu elementů a průchodu přes ně.



Obr. 7 Vybrané rozložení panelů

## 4.6. Formulář pro popis elementů

Formulář se automaticky otevře při výběru nového elementu. Pro úpravu vlastností stávajícího elementu vytvořeného v panelu plátno, klikne uživatel na daný element a zobrazí se mu příslušný formulář pro editaci údajů.

### 4.6.1. Data společná pro všechny druhy elementu

Formulář by měl evidovat základní informace, které se budou zaznamenávat pro každý element. Patří mezi ně název nebo jiný jednoznačný identifikátor, podle kterého by algoritmus programu NaviTerier vybíral v budově element odpovídající zadanému cíli. Každý element a vstup bude mít samozřejmě přiděleno unikátní identifikační číslo ID, které bude automaticky generováno

bez vědomí uživatele. Je totiž zbytečné, aby byl uživatel zatěžován informacemi o vnitřních procesech aplikace.

Ve formuláři budou evidovány přibližné rozměry elementů.

Volba přidání vstupů na jednotlivé strany by mohla být řešena samostatným tlačítkem, pomocí kterého by uživatel přidával vstup kliknutím na dotyčnou stranu. O něco přehlednější řešení by poskytovalo formulářové pole, které by umožnilo číselné zadání počtu vstupů. V případě dlouhých chodeb s mnoha dveřmi, které budou častou součástí popisovaných budov by to i podstatně urychlilo práci. Vstupy by byly rovnoměrně rozloženy po celé délce strany a jejich poloha by se dala upravit tažením myši.

Mezi informace, které by mohly posloužit ZP, patří i povrch podlahy (konkrétně jeho změna). Je neefektivní, aby při průchodu několika stejnými chodbami za sebou sdělovala navigace v každé chodbě, že je na podlaze koberec. Lepší by bylo, kdyby navigace oznamovala povrch podlahy pouze při jeho změně.

Formulář by měl také uživateli umožnit zaznamenat pomocí textového pole další specifické informace, které v něm nejsou předdefinovány, ale které ZP pro orientaci může použít.

#### 4.6.2. Obtížnost průchodu

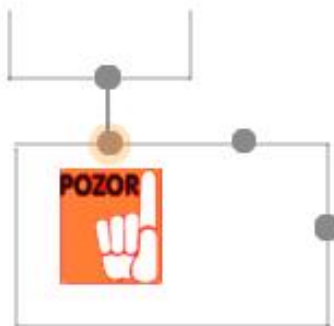
Při návrhu vycházím z toho, že výsledná data bude zpracovávat algoritmus pro hledání nejlepší možné cesty k cílovému elementu. Pokud se bude zvažovat jenom délka cesty jako součet elementů, nebudou výsledky výběru správné. Musíme zohlednit i náročnost průchodu, která u ZP podstatně ovlivňuje konečný čas průchodu trasou. Náročnost by se mohla počítat z informací o elementu. Dalším řešením je nechat uživatele, aby ohodnotil náročnost průchodu sám. To by umožnilo přizpůsobit daný popis lépe pro konkrétního ZP, pro kterého uživatel popis tvoří. Například pokud ZP preferuje pohyb po budově bez použití výtahu, elementy výtah by byly hodnoceny jako velmi náročné na průchod a algoritmus by je do nejlepší trasy vybral, jen pokud by významně zkrátil trasu.

#### 4.6.3. Nebezpečí

Jednou z nejdůležitějších informací, kterou musíme zaznamenat, je nebezpečí, hrozící při průchodu elementem. Jedná se o stálá nebezpečí, která může představovat například snížený průchod, nečekaný schod, překážka v oblasti hlavy a jiné nebezpečné objekty, zejména ty, které se nedají zaznamenat pomocí slepecké hole.

Dalším problémem je nebezpečí, které hrozí těsně po příchodu do elementu (Obr. 8). S největší pravděpodobností se s ním ZP setká ještě dřív, než ho na něj stačí navigace upozornit. Pro tento případ by mělo být nebezpečí ohlášeno už v předchozím elementu. Nutit uživatele, aby zaznamenával do elementů informace o následujícím nebezpečí, je nepřehledné a složité. Zejména proto, že by při vyplňování formuláře musel zohledňovat i další elementy, které by ještě neměl zaznamenat. Musel by se v popisu vracet, formuláře dodatečně upravovat a to by nebylo efektivní. Jednou z variant je navrhnout do algoritmu funkci, která automaticky doplní informaci o nebezpečí

do sousedních elementů. To sice nezatěžuje uživatele, ale bude to pravděpodobně zatěžovat ZP, protože dané nebezpečí následuje většinou pouze po průchodu jedním konkrétním vstupem. Nejlepší možností by tedy bylo navázat popis nebezpečí na konkrétní vstup. Znamenalo by to vytvořit speciální pole pro popis nebezpečí, které následuje těsně po průchodu.



Obr. 8 Nebezpečí po vstupu

#### 4.6.4. Způsob popisu průchodu elementem

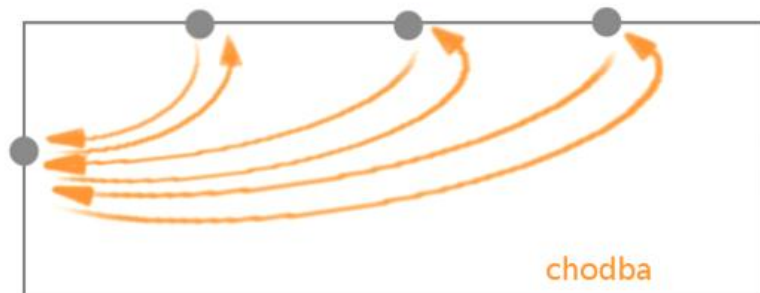
V práci o aplikaci NaviTerier [2] je způsob popisu průchodu navržen takto:

*„Každý segment je třeba popsat tak, aby do segmentu bylo možné kdekoliv vstoupit a kdekoliv z něj zase vystoupit. Nejobecněji je třeba vytvořit vždy  $n \times (n - 1)$  popisů, kde  $n$  je počet propojení na okolní segmenty. Jak ukazuje Obr. 7.1 a) pro jeden vstup do segmentu je třeba vytvořit  $(n - 1)$  popisů pro výstupní propojení. Tato situace platí pro  $n$ -ární kategorii. U některých kategorií může být počet nutných popisů nižší. Pro výtah platí, že ač do něj nastoupíme v jakémkoliv patře, vždy je použitý popis závislý pouze na cílovém patře. Proto u výtahu postačí s  $n$  popisy. V případě chodby může být počet popisů také nižší, pokud nepředpokládáme navigaci z jedné kanceláře do druhé kanceláře v rámci jedné chodby. Pro pouhý průchod chodbou postačí 2 popisy. Jeden pro směr tam a druhý pro směr zpět. Pro nalezení konkrétních dveří kanceláře je třeba  $2 \times (n - 2)$  popisů. V chodbě je  $(n - 2)$  kanceláří a do chodby je možné přijít ze dvou směrů. Pro jednu kancelář je tedy potřeba vytvořit 2 popisy jak ukazuje Obr. 7.1 b). Pro východ z kanceláře je možné použít pouze 2 obecné popisy – jeden pro cestu chodbou vlevo a druhý pro cestu chodbou vpravo. Jelikož jsou 2 strany chodby, je třeba použít 4 popisy pro výstup z kanceláří.“*

Jednou z hlavních částí popisu by tedy měl být popis průchodů elementem. Každý element obsahující více než jeden vstup má nejméně dvě možnosti průchodu. U chodby, do které vedou čtvery dveře, se dostáváme až na 12 kombinací, protože nesmíme zapomínat na to, že mezi dvěma vstupy se dá procházet v obou směrech (Obr. 9).

Už v druhé kapitole otvíráme téma vhodné kombinace mezi automatickým generováním a manuální tvorbou popisu. V tomto případě bude nejjistější zvolit variantu ručního popisu, kdy uživatel popíše všechny používané možnosti prů-

chodu elementem. Vzhledem k tomu, že některé elementy obsahují vysoké množství průchodů a orientace v nich je velmi nepřehledná, budu se mu snažit práci co nejvíce usnadnit. Především by bylo dobré, aby program vygeneroval všechny možné varianty průchodu elementem a uživatel je pouze popsal. Dále by bylo vhodné, aby mohl získat přehled, které kombinace už vyplnil a které ne.



Obr. 9 Varianty průchodu v elementu

## 5. Výsledný návrh

V této části se budu zabývat popisem struktury a funkcí výsledného návrhu a odůvodněním jejich výběru [9]. Některé volby mají více variant. Výběr finální varianty bude vybrán na základě výsledků testování s uživateli.

### 5.1. Hlavní okno

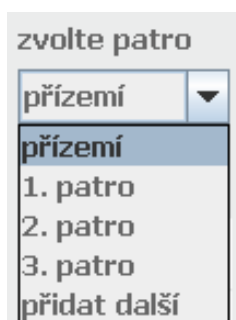
Hlavní okno je rozčleněno do několika panelů. Obrázek hlavního okna je umístěn v příloze D. Nahoře se nachází panel menu umožňující práci s projektem a panel nápověda. Vlevo jsou vertikálně pod sebou panely pro práci s elementy, makry a vzorovou mapou. Vpravo je prostor načítání jednotlivých formulářů pro popis elementu. Prostor uprostřed vyplňuje panel plátno, na kterém se pracuje s grafickým zobrazením elementů.

#### 5.1.1. Panel menu

Menu je umístěno jako u většiny standardních programů nahoře. Zajišťuje práci s projekty, jako je vytvoření nového projektu otevření již vytvořeného projektu nebo uložení. Funkce "uložit" by v sobě měla do budoucna zahrnovat volbu pro nahrání hotové mapy nebo makra do centrální databáze.

#### 5.1.2. Panel elementy a Panel makra

Panel elementy je rozčleněn na několik sekcí. První sekce zajišťuje přepínání mezi podlažími. Při vytvoření nového projektu je automaticky nastaveno výchozí podlaží přízemí. Výběr umožňuje i přidání nového podlaží, to by se mohlo řešit také samostatným tlačítkem. Druhá varianta je poněkud intuitivnější, první zase nezatěžuje hlavní okno větším množstvím tlačítek. Volbu konečné varianty vybereme podle výsledku testování.



Obr. 10 Volba podlaží

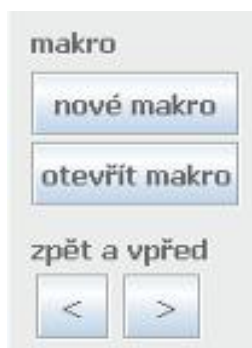
Druhá sekce je skupina tlačítek pro volbu jednoho z pěti základních objektů (tzv. elementů). V prvních návrzích se sice objevovalo rozbalovací menu, ale ukázalo se, že vzhledem k častému používání těchto tlačítek je možnost volby konkrétního elementu přímo z první úrovně časově efektivnější.





Obr. 11 Volba elementu

Další částí jsou tlačítka umožňující práci s makry. Makra by měla zajistit možnost následného upravení libovolně zvoleného celku. Tato funkce podstatně usnadní a zrychlí práci při popisu budov, ve kterých se vyskytují půdorysně podobné části, a vytváření dalších popisů s jinou úrovní složitosti. Nezbytnou součástí návrhu je možnost krokování zpět a vpřed, kterou ocení zejména nezkušení uživatelé.



Obr. 12 Panel makro a krokování

Ve spodní části aplikace se nachází tlačítko "otevřít mapu". Jeho stisknutím se uživateli otevře nabídka kompatibilních souborů, které může načíst do náhledu vzniklého horizontálním předělením panelu plátno manipulovatelnou lištou.

### 5.1.3. panel plátno

Interaktivní okno zobrazuje jednotlivé elementy, se kterými se dá v rámci aktivní plochy okna pracovat (bude upřesněno v jednotlivých formulářích pro popis elementu). Vytváří schématický obraz tvořeného celku. Poskytuje nám grafické ověření hodnot, které zadáme do formuláře a navíc umožňuje propojení jednotlivých vstupů elementů do konečného celku. Každý objekt (element nebo vstup) se kterým se pracuje, je pro lepší přehlednost barevně zvýrazněn a dá se s ním po označení dodatečně hýbat. Při kliknutí na již vložený element v panelu

plátno se automaticky objeví formulář pro jeho editaci. Při kliknutí na šipku označující konkrétní průchod elementem se v políčku volby popisovaného průchodu automaticky zvolí tento označený průchod. Každá důležitá akce, provedená ve formuláři, vyvolá odezvu na kreslícím plátně a naopak. Tak bude uživateli poskytnuta dostatečná verifikace jím zvolených akcí.

Okno lze rozdělit na dvě části. Při načtení mapy se automaticky přepůlí horizontální pohyblivou lištou, se kterou se dá manipulovat tažením za šipky. V obou částech jsou vpravo nahoře umístěny malé ikony s lupou a funkcí "zoom", pomocí kterých lze obraz přiblížit nebo oddálit. S celým náhledem se potom dá hýbat pomocí tzv. "scrollovacích lišt" nebo prostředního tlačítka myši, jako u standardních grafických programů.

V rámci okna se zatím pracuje pouze myší, do budoucna by se určitě dalo využít označování elementů a vstupů pomocí hromadných výběrů či klávesových zkratk.

#### 5.1.4. Panel nápověda

Panel nápověda je panel s tlačítkem, které otevře okno s klasickou nápovědou, průvodcem pro první tvorby map, tutoriály, vzorově vyřešenými popisy, základy prostorové orientace nevidomých a dalšími dokumenty.

Druhou část panelu by tvořila dynamicky generovaná nápověda. Tuto myšlenku využívají některé systémy pro tvorbu technických výkresů, jako je například AutoCad. V liště jsou automaticky generované nápovědy, návody a upozornění podle toho, kde se uživatel právě nachází, jakou akci spustil nebo jaká akce se od něj zrovna očekává. To podstatně urychlí práci zejména v začátcích, kdy uživatel ještě přesně neví, jak se program chová. Pokud nastane úskalí při použití určitého nástroje, nemusí začít listovat rozsáhlou nápovědou, ale může mu pomoci jedna krátká věta, která mu poradí jak dál.

Pro lepší představu zde uvádím příklad:

**Situace:** Uživatel klikne na vstup V3.

Nápověda zobrazí z těchto dvou jednu náhodně vybranou informaci:

*Vstup přesunete pomocí tažení myší.*

*Pokud chcete vstup V3 spojit s jiným, označte druhý vstup kliknutím.*

**Situace:** Uživatel vytvořil poschodí „patro 1“ a zvolí nové poschodí s názvem „patro 2“. Nápověda zobrazí tuto informaci:

*Pokud je "patro 1" podobné jako "patro 2", uložte "patro 1" jako makro, v "patro 2" je otevřete a upravte.*

### 5.1.5. Panel formulář

Panel formulář slouží k popisu elementů. Každý formulář je uzpůsoben tak, aby naváděl k popisu vlastností, které potřebujeme u elementu zjistit a zároveň poskytoval možnost zaznamenat libovolné informace.

V další části kapitoly popíšu jednotlivé formuláře. Všech pět typů formulářů má stejnou strukturu. Díky tomu by měl mít uživatel možnost postupovat při vyplňování analogicky a při pochopení významu jednoho formuláře použít tyto vědomosti k vyplňování dalšího. Každý typ formuláře je rozdělen do tří podpanelů. První se věnuje obecným vlastnostem elementu, druhý je určen pro zaznamenání obtížnosti průchodu chodbou a nebezpečí a ve třetím se popisují jednotlivé průchody elementem.

## 5.2. Formulář pro popis chodby

### 5.2.1. Název elementu

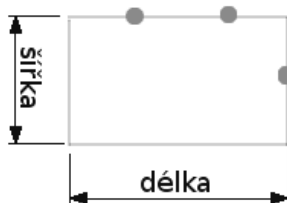
Název elementu je nepovinný. Vyplňovat název chodby má smysl pouze v případě, že se jedná o cílový objekt. Abychom zamezili vzniku redundantní informace o názvu a typu, předdefinujeme typ elementu v políčku pro název. Pokud bychom tak neučinili a uživatel by napsal do názvu „*hlavní chodba*“, vygenerovaný popis by zněl: „*Chodba hlavní chodba, dlouhá ....*“.

V případě, že bychom se této redundanci vyhnuli tak, že by algoritmus negeneroval automaticky do čteného textu typ, mohlo by se zcela dobře stát, že by uživatel předpokládal, že název chodba bude vygenerován a vyplnil by jenom název „*vstupní*“. Výsledek by vypadal asi takto: „*Následuje vstupní, dlouhá...*“

Dalo by se to samozřejmě obejít přidáním podmínky do algoritmu, která by prohledala obsah formulářového pole „název“. Pokud by neobsahoval podřetězec „chodba“, doplnil by ho tam. To má ale hned dvě nevýhody. Takto generovaný výsledek by mohl porušit syntaktickou správnost věty a navíc by nebyl transparentní pro autora. Proto jsem zvolila kombinaci předdefinovaného typu elementu jako součásti názvu a zpětné kontroly algoritmem.

### 5.2.2. Přibližné rozměry

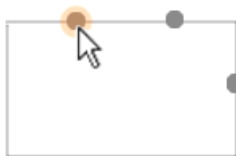
Přibližné rozměry jsou jen orientačním udáním délky a šířky elementu. Slouží pro vytvoření schématického zobrazení elementu na mapě, pro přibližné určení délky trasy vybrané navigací a samozřejmě jako informace generovaná do popisu elementu nevidomému.



Obr. 13 Rozměry elementu

### 5.2.3. Počet vstupů

Tato čtyři políčka slouží ke zvolení počtu vstupů, které vedou do elementu. Po zvolení jakéhokoliv počtu se na příslušné straně elementu automaticky rozloží schematické značky vstupů v podobě vyplněného barevného kruhu. V rámci grafického zobrazení elementů bude možné pomocí levého tlačítka myši a tažením pohybovat s jednotlivými vstupy v rámci dané strany elementu (obr.15).



Obr. 14 Manipulace se vstupem

### 5.2.4. Povrch podlahy

Povrch podlahy se určuje zaškrtnutím políček předdefinované volby. Je zbytečné, aby navigace nevidomého při průchodu pěti stejnými chodbami pětikrát upozornila na to, že je na podlaze koberec. V ideálním případě by měla navigace oznámit pouze změnu stávajícího povrchu na jiný. A protože nemůžeme uživatele zatěžovat zbytečnostmi, závěrečná data projdou před uložením algoritmem, který uloží informaci o povrchu pouze tehdy, pokud nebude u sousedního elementu povrch stejný. Tyto informace by si mohla upravovat až navigace NaviTerier, ale vzhledem k tomu, že se jedná o mobilní aplikaci, měla by být co nejméně zatěžována vedlejšími úpravami dat a jinými procesy.

### 5.2.5. Doplnující informace

Toto textové pole slouží k zachycení kterýchkoliv informací, které nejsou zaznamenány ve formuláři, a přesto autor uzná, že je vhodné je zaznamenat. Slouží i k přizpůsobení popisu na míru konkrétní zrakové vadě. Pokud například nevidomý vnímá ostré přechody světla, může zde autor zaznamenat výskyt velkých oken, popřípadě jiné světelné zdroje.

Toto textové pole však neslouží pro zaznamenání informace o nebezpečí, pokud nehrozí v celém elementu. Vzhledem k tomu, že nebezpečí většinou není

přítomno v každém průchodu elementem, je třeba toto nebezpečí při konkrétním vstupu zaznamenat do formulářového pole k tomu určeného.

element CHODBA

název elementu

přibližné rozměry: délka (m)

šířka (m)

**počet vstupů (tzv. pointerů)**

nahoře  dole

vlevo  vpravo

**povrch podlahy (zatrhněte)**

koberec  parkety  beton

lino  dlažba

**doplňující informace**

Obr. 15 Formulář popisu chodby

### 5.2.6. Index obtížnosti průchodu chodbou

Ve čtvrté kapitole jsem zmínila potřebu zaznamenat obtížnost průchodu elementem. K tomu jsem zvolila tzv. index obtížnosti průchodu, což je číslo od jedné do deseti, které přibližně charakterizuje zátěž a míru nebezpečí, které je nevidomý při průchodu vystaven. Index nemá obecně stanovené hranice pro konkrétní případy a nezáleží na tom, jestli jeden autor bude jednu a tu samou chodbu označovat indexem 2, zatímco jiný indexem 4. Důležité je, aby byly elementy v rámci jedné budovy hodnoceny podle stejného měřítko. Výsledné určení časové náročnosti trasy by potom mělo být kombinací součtu délek elementů a indexů obtížnosti. Samozřejmě že výsledek není úplně přesný. Délka je určena pro element obecně a v některých případech nemusí nevidomý projít přes celý element, nebo se může vyhnout nebezpečí, které zvyšuje index obtížnosti průchodu. Ale pro potřebu výběru trasy, které se bude řešit při implementaci navigace NaviTerier je třeba zavést tuto informaci do popisu elementu.

Základní informace o tom, co je to index obtížnosti průchodu a příklady použití, by měly být dostupné z místní nápovědy spustitelné kliknutím na tlačítko „?“ . Tyto informace samozřejmě najde uživatel i v nápovědě, ale z hledem k tomu, že sekce nápovědy bude velmi rozsáhlá, umístila jsem do této části formuláře rychlý vstup do nápovědy, který zobrazí pouze informace o indexu obtížnosti průchodu. Není to obvyklé, proto o umístění tohoto tlačítka rozhodnu až na základě toho, jak se osvědčí při testování a jestli tuto možnost uživatelé vyhodnotí pozitivně.

### 5.2.7. Hned při vstupu hrozí nebezpečí

V požadavcích na systém zmiňuji nutnost včasného varování před nebezpečím, které následuje bezprostředně po vstupu do elementu. Velmi pravděpodobně dojde k situaci, kdy nevidomý projde dveřmi a informace o dalším elementu si pustí, až když už stojí na následující chodbě nebo v místnosti. Proto jsem navrhla speciální část formuláře zabývající se tímto nebezpečím, kde autor vyplní popis nebezpečí a označí vstupy, přes které se k němu přistupuje. Možnost výběru více variant jsem zvolila proto, že k jednomu nebezpečí může často vést i více vstupů.

V případě, že se takových nebezpečí vyskytuje víc, stiskne uživatel volbu „ulož nebezpečí“, data se uloží a dialogové okno ho informuje o tom, že nebezpečí bylo uloženo a může vyplnit další.

Jako ve všech částech formuláře, kde se označují konkrétní vstupy nebo průchody elementem, je možné vybrat vstup dvěma způsoby. První umožňuje výběr z formuláře, kdy se zaškrtnutím políčka u určitého vstupu potvrdí volba jeho zvýrazněním na grafickém schématu popisu. Druhá možnost je kliknutí přímo na grafickou značku vstupu, po kterém se automaticky označí daný vstup ve formuláři.

průchod chodbou

index obtížnosti průchodu 1 ?

**hned při vstupu hrozí nebezpečí**

vede k němu vstup(y)  V1  V2

V3

ulož nebezpečí

Obr. 16 Panel obtížnosti průchodu

### 5.2.8. Popisy průchodů chodbou

Pro popisy průchodů je oddělený panel, který obsahuje vygenerované dvojice vstupů označeného elementu, textové pole pro popis a tlačítko pro uložení.

Po zvolení libovolné kombinace vstupů z nabídky a vyplnění textového pole se popis uloží stiskem tlačítka „uložit cestu“. Uložená dvojice vstupů z nabídky zesvětlá ve formuláři i v grafickém schématu, aby měl uživatel přehled o tom, které průchody už vyplnil a které ne.

Do budoucna je možné předdefinovat libovolnou strukturu textového pole pomocí předdefinovaných částí textu a navést tak uživatele na správný způsob popisu.

Konkrétní kombinace dvou vstupů lze vybírat stejně jako v předchozím panelu dvěma způsoby. Buďto se vybere možnost ve formuláři a v grafickém schématu elementu se automaticky zvýrazní šipka představující vybraný průchod přes element a nebo se vybere cesta kliknutím na šipku v grafickém schématu a ve formuláři se označí sama.

Panel s titulem "popisy průchodů chodbou". Vnitřní obsah: text "popište cestu mezi těmito vstupy (pointry)", textové pole "ze vstupu V1 do V2" s výběrovou šipkou, velký prázdný textový pole s vertikálními posuvnými prvky (šipky nahoru a dolů, a tři horizontální čáry), a tlačítko "uložit cestu" s oranžovým textem.

Obr. 17 Panel popisy průchodů

Pokud budeme možné dvojice vstupů generovat, s velkou pravděpodobností se stane, že se vygenerují i průchody, které nebude nevidomý určitě potřebovat. Ukázkovým případem je chodba kanceláří a jiných místností. Je zbytečné, aby uživatel popisoval cestu z kanceláře do vedlejší místnosti určené výhradně pro personál. Popíše tedy jen ty průchody, které se opravdu používají a ostatní bude moci v grafickém schématu odstranit. Učiní tak označením šipky, znázorňující průchod mezi vstupy a jejím smazáním.

### 5.3. Formulář pro popis místnosti

Formulář místnost má stejná formulářová pole jako chodba. Jedinou výjimkou je název, který je z identifikačních důvodů u místností povinný. Tento element bude pravděpodobně nejčastěji voleným cílem trasy. Proto je nutné, aby jeho název obsahoval jednoznačný identifikátor. Z tohoto důvodu by bylo vhodné do budoucna zvážit i možnost zadávání názvu místnosti do jednoho textového pole a jména osoby, která sídlí v dané místnosti, do pole dalšího. Tyto možnosti se budou zvažovat až na základě informačních potřeb algoritmu, který bude v hotovém popisu budovy hledat nejlepší trasu z výchozího bodu do cíle, zadaného ZP.

Nabízí se otázka, proč jsme nesloučili formulář místnosti a chodby, když jsou identické a nerozdělili ho pouze přidáním pole, kde by se zvolilo, zda se jedná o chodbu nebo o místnost. Je to zejména proto, že by to bylo pro uživatele neefektivní a matoucí. Byl by nucen vyplňovat další informace a volba elementu by byla méně intuitivní.

Výhodou je, že nemusíme uživateli definovat, co je chodba a co místnost a nenutíme ho, aby se v nejasných případech řídil podle této definice, protože oba formuláře poskytují pole pro zaznamenání stejných informací.

## 5.4. Formulář pro popis schodiště

### 5.4.1. Název elementu a délka

Název elementu je nepovinný. Nepředpokládáme, že by ZP zadával schodiště jako cíl cesty, ale vyloučit to nemůžeme. Je možné, že se v budově nachází velké a specifické schodiště, a proto ho zvolí jako místo schůzky s další osobou.

Délka je přibližně zadávána v metrech stejně jako u předchozích elementů. Umístění pole pro zadání šířky je otázkou. Vycházím-li z toho, že nevidomý se zřejmě bude držet jedné strany schodiště, pravděpodobně té se zábradlím, nebude snad zadání šířky potřebné. Toto tvrzení ale nemohu dokázat, proto informaci o šířce schodiště do formuláře raději zařadím.

### 5.4.2. Typ schodiště a sklon

Jednou z důležitých informací je typ schodiště. Pokud bychom tuto informaci nechali zadat uživatele pouze do textového pole a nepředdefinovali mu možnosti, mohlo by se stát, že by nepochopil význam a zaznamenal by něco jiného. Proto mu nabídneme možnost výběru z několika předdefinovaných obecně platných pojmů.

Necháme mu také možnost, aby si vybral z několika dalších informací a popřípadě je zaznamenal výběrem z výše zmiňovaných variant.

*Typ schodiště: přímé, točité, lomené*

*Schodiště je: neurčeno, pozvolné, příkré s úzkými schody*

The image shows a screenshot of a web form titled "element SCHODIŠTĚ". The form contains the following fields:

- "název elementu" with the value "schodiště" and "(nepovinné)" next to it.
- "přibližná délka" with an empty input field and "(m)" next to it.
- "přibližná šířka" with an empty input field.
- "typ schodiště" with a dropdown menu showing "přímé".
- "schodiště je" with a dropdown menu showing "neurčeno".

Obr. 18 Vlastnosti schodiště



### 5.4.3. Povrch a doplňující informace

Další část panelu je stejná jako v ostatních formulářích. Uživatel vyplní povrch schodiště a dopíše informace, které by mohly ZP pomoci a nejsou výše předdefinovány.

### 5.4.4. Průchod schodištěm a popis průchodu

Tento panel má stejnou funkci a strukturu jako u ostatních formulářů. Umožňuje zaznamenat nečekané nebezpečí, které hrozí při vstupu na schodiště.

Vstup je z každé strany pouze jeden, a to buď z navazující chodby nebo po-desty. Průchod schodištěm má tedy v tomto případě pouze dvě varianty, tam a zpět. Uživatel v popisu doporučí vodící linii, kterou může být například zábradlí.

Obr. 19 Panel průchodu schodištěm

### 5.4.5. Schodiště navazuje na vstup

Určení návaznosti schodiště je celkem obtížné zejména proto, že jedna část většinou navazuje na jiné poschodí. I toto vybrání příslušného vstupu lze uskutečnit dvěma způsoby. Buď uživatel napíše číslo vstupu a zvolí poschodí sám a na plátně s grafickým schématem popisu se mu ukáže vybrané poschodí se zvýrazněným vstupem, nebo si pomocí změny patra zvolí jiné a označí vstup kliknutím na jeho grafické znázornění. Tato volba se automaticky vyplní ve formuláři.

Obr. 20 Panel pro napojení schodiště

## 5.5. Formulář pro popis dveří

V kapitole 4.2 “Elementy a vstupy“ se zabývám možností, ve které dveře nebudou tvořit samostatný typ elementu, ale uživatel je popíše v rámci jiného elementu. Při zvolení této varianty by musely být informace, které jsou nezbytné k průchodu dveřmi, zaznamenány spolu s informacemi o průchodu daným elementem. Uživatel by pak musel zaznamenat podstatně více informací najednou, proto jsem zvolila možnost samostatného formuláře pro popis dveří. Pro uživatele to bude přehlednější a méně náročné. Rozhodující bude, jak takto generovaný popis přijmou uživatelé aplikace NaviTerier. To odhalí další fáze testování doporučená v kapitole 8, kde navrhuji provést další testování se ZP uživateli.

### 5.5.1. Všeobecný popis

Název dveří by měl být podobný jako název místnosti. Proto pokud je za nimi místnost K123, nazveme je například „dveře do místnosti K123“. Do typu dveří jsem zahrnula 5 standardních typů, do kterých patří:

- otevíravé s klikou*
- otevíravé bez kliky*
- posuvné*
- turniketové*
- jiné-specifikuj dole v popisu průchodu*

**element DVEŘE**

název elementu

typ dveří

**povrch podlahy (nepovinné)**

koberec    parkety    lino

dlažba    beton

**doplňující informace**

Obr. 21 Formulář pro popis dveří

### 5.5.2. Panel průchod dveřmi

Panel průchodu dveřmi je stejný jako u ostatních elementů. Obsahuje hodnocení obtížnosti průchodu dveřmi a textové pole pro zaznamenání nebezpečí, které následuje v elementu přímo po vstupu dovnitř.

### 5.5.3. Popis průchodu dveřmi

Popis průchodu dveřmi obsahuje kontextovou nápovědu stejně jako panel pro určení obtížnosti průchodu dveřmi, kde by mělo být pár stručných návodů s doporučenou strukturou popisu a informacemi, které by měl uživatel zaznamenat. Při kliknutí na grafické znázornění vstupu se ve formuláři zvýrazní název vstupu, na který uživatel kliknul. A pokud uživatel klikne na název vstupu ve formuláři, barevně se vyznačí jeho značka v grafickém zobrazení popisu.

Obr. 22 Panel popisu průchodů

## 5.6. Formulář pro popis výtahu

### 5.6.1. Název elementu

Název elementu není povinným polem. Měl by být pouze informativního charakteru. Zejména pokud se jedná například o tzv. "páter noster" nebo jiné specifické druhy výtahů, měl by tuto informaci uživatel zaznamenat.

### 5.6.2. Tlačítko na vyvolání výtahu a otvírání dveří

Pole nadepsané textem „*tlačítko na otvírání výtahu se nachází*“ obsahuje tyto možnosti:

*vpravo vedle dveří*  
*vlevo vedle dveří*

Další informací, kterou ZP využije při nástupu do výtahu je způsob, jakým se otvírají dveře do výtahu. Tato rozbalovací nabídka obsahuje tyto možnosti:

*Automaticky*  
*tahem za kliku*  
*posunutím do stran*  
*jiné...*

### 5.6.3. Ovládání výtahu a doplňující informace

Do tohoto textového pole by měl uživatel jednoduše popsat, kde se ve výtahu nachází jeho ovládání. V případě, že má výtah atypickou konstrukci a není

možné ho popsat pomocí předdefinovaných polí, popíše ho uživatel do textového pole pro doplňující informace.

element VÝTAH

název elementu

tlačítko na vyvolání výtahu se nachází

dveře od výtahu se otevírají

ovládání výtahu se nachází

popis panelu pro ovládání

Obr. 23 Část panelu pro popis výtahu

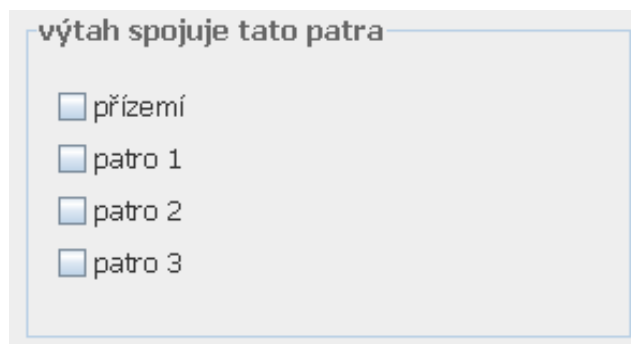
Popis panelu pro ovládání uživatel vytvoří podle instrukcí ve směru zdola nahoru. Typický zápis vypadá takto:

*„zvonek, stop, suterén, přízemí, první patro, druhé patro, třetí patro“.*

V rámci doplňujících informací, které tvoří poslední formulářové pole popisu elementu, může uživatel zaznamenat specifika výtahu jako je přepis tlačítek ovládání do Braillova písma nebo východ na opačnou stranu.

#### 5.6.4. Výtah spojuje tato patra

Při prvním vkládání elementu výtah se vyplní všechny informace. Při dalším použití tohoto elementu v rámci jiného podlaží bude stačit vyplnit jméno výtahu a pokud se bude shodovat s již popsaným výtahem, automaticky se nastaví stejný obsah formuláře pro popis.



Obr. 24 Panel propojení pater pomocí výtahu

### 5.7. Závěrečné zpracování dat algoritmem

Po zpracování celého popisu zadá uživatel volbu uložit. Ještě před tím, než se data uloží, spustí se závěrečný algoritmus, který zpracuje a zvaliduje data. Algoritmus by měl zahrnovat veškeré úpravy vstupů zadané uživatelem do datové struktury, zvolené pro NaviTerier včetně úpravy informace o povrchu podlahy, kontroly pospojování všech vstupů nebo názvů elementů.

### 5.8. Menu u elementu vyvolané stiskem pravého tlačítka myši

Je pravděpodobné, že se v popisu budovy vyskytne několik velmi podobných objektů. Aby tyto elementy nemusel uživatel vytvářet pokaždé znovu, bude moci elementy zkopírovat a pouze poupravit údaje, které se mu implicitně vyplní podle kopírované předlohy. Učiní tak použitím klávesových zkratk Ctrl+c (kopírovat) a Ctrl+v (vložit) a nebo kliknutím pravým tlačítkem myši na element, které vyvolá malé menu u elementu, kde vybere možnosti kopírovat element a vložit.

### 5.9. Klávesové zkratky

Při dalším vývoji aplikace by měly být do nástrojů pro ovládání zavedeny klávesové zkratky. Ty by posloužily zejména zkušeným uživatelům, kteří by tak mohli zvýšit efektivitu své práce.

## 6. Tvorba a implementace prototypů

Low fidelity prototyp byl vytvořen pomocí grafického programu Inkscape. Tento program definuje grafické prvky pomocí vektorů a křivek a je vhodný pro tvorbu grafiky s kontrastními přechody.

### 6.1. High fidelity prototyp

Pro tvorbu high fidelity prototypu bylo nutné zvolit technologii, která by umožnila interaktivní práci s prototypem. Rozhodla jsem se tedy vytvořit aplikaci v programovacím jazyce Java [10]. Java je objektově orientovaný programovací jazyk, který vyvinula firma Sun Microsystems. Pro tvorbu prototypu jsem využila platformu Java SE (Java Standard Edition), určenou pro desktopové počítače. Výhodou jazyka Java je nezávislost na operačním systému. Ke spuštění programu je potřeba pouze to, aby byl instalován správný virtuální stroj (JVM). Pro editaci a kompilaci kódu jsem použila vývojové prostředí Netbeans IDE.

#### 6.1.1. Formuláře

Při vytváření grafického rozhraní jsem využila framework Swing. Framework [6] je softwarová struktura, která slouží jako podpora při programování a vývoji softwaru. Může obsahovat například podpůrné programy nebo knihovnu API. Swing [7] je knihovna uživatelských prvků na platformě Java pro snadnější tvorbu grafického rozhraní. Knihovna Swing poskytuje aplikační rozhraní pro tvorbu a obsluhu klasického grafického uživatelského rozhraní aplikace. Pomocí Swingu je možno vytvářet různé panely, dialogy, tlačítka, rámečky, rozbalovací seznamy a jiné formulářové prvky. S použitím frameworku Swing byly vytvořeny všechny komponenty a panely prototypu vyjma panelu plátno.

#### 6.1.2. Panel plátno

Pro zobrazení textu, základních geometrických obrazců a práci s nimi není potřeba žádných složitých technologií. Pro jednoduché kreslení v javě se obvykle používají dvě metody. Je možnost napsat třídu dědící z `JComponent` (komponenta frameworku swing) a přepsat její metodu `paint(Graphics g)` tak, aby se do grafického kontextu `g` vykreslily námi požadované geometrické obrazce. Vybrala jsem si druhou metodu a to využití třídy `Canvas`. Komponenta, dědící od komponenty `Canvas`, představuje prázdnou obdélníkovou oblast, na kterou se dá pomocí přepsání metody `paint(Graphics g)` vykreslit libovolný grafický objekt. Zároveň poskytuje metody, pomocí kterých se dají jednoduše zpracovávat vstupy od uživatelů.

#### 6.1.3. Omezení funkcionality prototypu

Funkcionalita prototypu byla upravena podle zadání testu. Metoda `paint` byla napsána tak, aby v první části testu refletovala přidávání elementů popsaných v zadání. Pro spojování vstupů z druhé části zadání se přepne pomocí tlačítka „>“ funkce krokování do stavu, kdy se vykreslí popis z části jedna a je možné pospojovat aktivně reagující vstupy.

## 7. Ověření návrhu uživatelského rozhraní

Pro ověření uživatelského rozhraní se používají různé metody testování [11]. Ty se provádějí ve všech fázích návrhu, aby se odhalily a odstranily nedostatky před tím, než bude naprogramována plně funkční aplikace a veškeré zásahy budou stát mnoho času a finančních prostředků.

### 7.1. Výsledky ověření low fidelity prototypu

Pro ověření návrhu low fidelity prototypu z kapitoly 4 zobrazeného v příloze C, jsem použila testování metodou kognitivního průchodu. Ta je popsána v mé semestrální práci [3]. Bylo testováno několik základních úkolů, jako je přidání elementu nebo nahrání vzorového obrázku. Nebyly zjištěny žádné závažné nedostatky návrhu.

Při detailním rozpracování obsahu formulářů pro popis elementu jsem usoudila, že bude lepší umístit panel s formuláři na pravou stranu aplikace. Tuto možnost uvádím jako alternativní řešení rozmístění panelů v kapitole 4.4 .

Při zkoumání posloupnosti akcí pro přidání nového elementu se ukázalo, že víceúrovňové menu je neefektivní. První úroveň v podobě tlačítka “přidat element“, které otevře nabídku jednotlivých elementů je zbytečná. Uživatel bude tlačítka pro přidání elementu používat velmi často. Proto je efektivnější, aby byla přístupná hned z první úrovně.

### 7.2. Testování high fidelity prototypu

#### 7.2.1. Metoda testování

Pro verifikaci high fidelity prototypu jsem zvolila testování s uživateli. Použiji metodu testování použitelnosti (Usability testing). Tato metoda je velmi efektivní. Uživatelské rozhraní se na základě výsledků testu přizpůsobuje potřebám uživatelů, kteří si sami určí, co vyhovuje jejich potřebám a co jim naopak činí problémy..

#### 7.2.2. Tvorba prototypu

Pro potřeby testování jsem vytvořila částečně funkční prototyp. Při tvorbě jsem zkombinovala metodu horizontálního a vertikálního prototypování. Vytvořila jsem kompletní uživatelské rozhraní a přiřadila jsem funkcionalitu jen těm částem aplikace, které budou použity při testování.

#### 7.2.3. Výběr účastníků testu

Kvalitativní verze testování zahrnuje 6-12 účastníků. Volba kvantitativní verze by nebyla vhodná zejména proto, že účastník stráví přípravou na testování mnoho času. Pro výběr vhodné skupiny uživatelů jsem použila průzkum pomocí krátkých dotazníků (tzv. screeners), ve kterých jsem zjišťovala základní údaje o uživatelích, jako je věk, pohlaví, úroveň vzdělání a doplňující otázky, na které se odpovídalo pouze ano nebo ne.



Na základě výsledků z těchto dotazníků jsem zvolila sedm potenciačních uživatelů, kteří souhlasili s účastí při testování.

IDO	Pohl	věk	Vzdělání a obor	Znalost práce s PC	Znalost podobného SW	Znalost problematiky ZP (doplňující pozn.)
1	muž	24	VŠ, technický	průměrná	ano	špatná (zajímá se o pomoc ZP pasivně)
2	žena	49	VŠ, humanitní	velmi špatná	ne	průměrná (zajímá se o pomoc ZP žáků)
3	muž	32	SOU bez maturity	průměrná	ne	velmi dobrá (pracovník se ZP)
4	žena	18	VOŠ	průměrná	ne	velmi dobrá (dobrovolný pracovník se ZP)
5	muž	23	VŠ technický	dobrá	ano	velmi dobrá (má nevidomou kamarádku)
6	muž	22	VOŠ zdravotnický	velmi dobrá	ano	velmi dobrá (dobrovolný pracovník se ZP)
7	žena	27	VŠ Humanitní	špatná	ne	velmi dobrá (zajímá se aktivně o pomoc ZP)

Tab. 1 Výběr účastníků testu

S těmito účastníky byl domluven datum a okolnosti testování. Všichni byli seznámeni s průběhem testování a ujištění o tom, že budou výsledky prezentovány anonymně. Bylo zdůrazněno, že se nejedná o testování uživatelů, ale o testování aplikace. Každý účastník obdržel předem informativní materiály k testování.

#### 7.2.4. Materiály pro testování

Pro uživatele bylo vytvořeno několik informačních dokumentů, obsahujících seznámení s projektem NaviTerier [2], testovanou aplikací a s principy tvorby popisů budov pro NaviTerier. Každý účastník měl navíc k dispozici tutoriál prototypu testované aplikace, kde se pomocí náhledů grafického rozhraní a popisů funkcionality mohl s prototypem předem seznámit.

#### 7.2.5. Popisovaný objekt

Vzhledem k tomu, že se testy prováděly individuálně s každým uživatelem zvlášť a na jiném místě, bylo obtížné vybrat část budovy, která by tvořila předlohu pro testování. Fotodokumentace není vždy přehledná a také je obtížné nalézt část objektu, na které by se při popisu využilo co nejvíce funkcí. Z těchto

důvodů byla vytvořena virtuální budova. Pro tu část budovy, která byla předmětem popisu jsem vytvořila prostorový model (Obr. 25). Ten byl nasnímán z několika úhlů a tyto obrázky byly použity jako podklady pro testování místo fotodokumentace reálného objektu.

Všechny tyto materiály měly účastníci testu po celou dobu k dispozici v papírové i v elektronické podobě.



Obr. 25 Model budovy

### 7.2.6. Zadání testu

Pro přiblížení reálné situaci byly vytvořeny tři krátké profily fiktivních ZP osob. Každý uživatel psal popis pro jednu z nich.

Zadání testu bylo rozděleno na dílčí úkoly. Účastníci postupovali přesně podle instrukcí. V první části testu uživatelé přidávali a popisovali elementy. Za každý typ byl zvolen minimálně jeden element. Jeho vlastnosti byly vybrány tak, aby se dalo při popisu využít co nejvíce formulářových polí. U některých elementů byl vybrán konkrétní průchod, který měli účastníci popsat.

V druhé části testu uživatelé pospojovali přidané elementy pomocí propojení jejich vstupů. Plné znění zadání je připojeno v příloze E.

Po celou dobu testu byly k dispozici obrázky budovy a tutoriál v papírové podobě. Ten mohli účastníci používat místo nápovědy programu, která ještě nebyla implementována.

### 7.2.7. Průběh testování

Na začátku byl proveden pilotní test, který měl za úkol odhalit nedostatky v podkladech pro testování. Test odhalil pouze dvě nejasnosti v popisu elementů a jeden chybný název ve formulářovém poli testovaného prototypu. Tyto nedostatky byly odstraněny.

Prototyp se testoval na notebooku s úhlopříčkou obrazovky 14,1 palců a rozlišením 1400x1050 pixelů. Jeho nastavení bylo upraveno podle individuálních potřeb účastníků, kteří mohli využít touchpad nebo externí myš. Možnost použití externí klávesnice nikdo nevyužil.

Celý test byl sledován a dokumentován. Účastníci v průběhu komentovali svou práci a sdělovali své postřehy. Na konci testování vyplnili připravený dotazník, ve kterém měli možnost se subjektivně vyjádřit ke vzhledu a funkcionalitě aplikace.

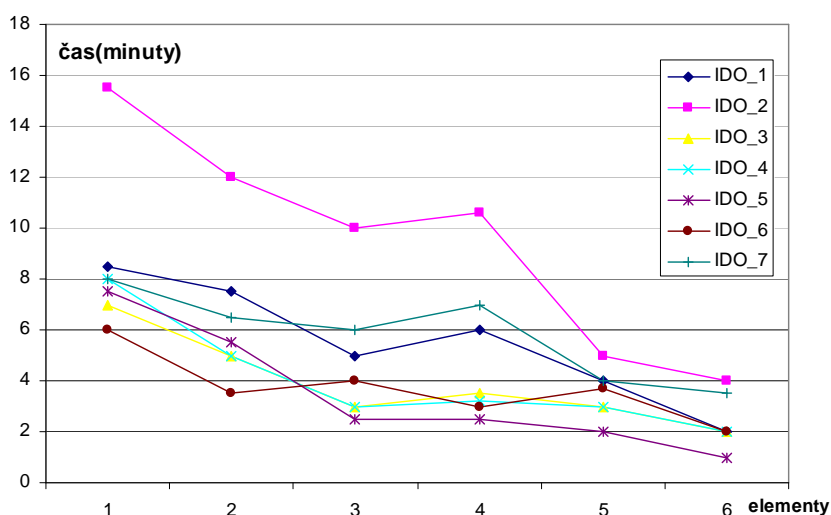
### 7.3. Výsledky testů

Všichni účastníci splnili zadání bez větších obtíží a v přijatelném časovém rozmezí. V následující části kapitoly se budu věnovat jednotlivým problémům, které se při testování vyskytly, připomínkám účastníků a vlastním postřehům z průběhu testování.

#### 7.3.1. Časové výsledky testu

Důležitým parametrem hodnocení aplikace byla naučitelnost, tedy schopnost zapamatovat si jednotlivé části uživatelského rozhraní a využít tyto vědomosti při další práci s aplikací. Tuto vlastnost můžeme posoudit z časových průběhů testu (Obr. 26). V úvahu se musí brát i objektivní časová náročnost úkolů, která je pro prvních 5 elementů srovnatelná. U třetího, pátého a šestého elementu se nemusel popisovat žádný průchod, což výsledné hodnoty zkresluje.

U účastníka s IDO\_5 došlo k výraznému prodloužení dob pro popis elementu. Bylo to způsobené především velmi pomalým psaním a nejistotou práce s myší. Není však tolik podstatné, jakých hodnot graf nabýval, jako jejich změny v průběhu testování. Z nich můžeme usoudit, že časy, které potřebovali účastníci testování, se s každým dalším elementem snižují. Z pochopitelných důvodů nabývá graf na začátku vysokých hodnot. S dalšími elementy ale hodnoty klesají. Výrazné změny u elementů 3, 5 a 6 sice ovlivnily průběh testování, ale i tak lze říci, že je výsledek časové analýzy příznivý.



Obr. 26 Graf časového průběhu testování

### 7.3.2. Problémy odhalené testováním

Při testování nedošlo k odhalení žádných závažných chyb prototypu. Přesto je nutné provést malé úpravy návrhu. A zvážit další doporučení.

#### Seznam nejčastějších chyb

Podle výsledků testování je zřejmé, že uživatelé dělají velmi často stejné chyby. Patří mezi ně například nedodržování gramatiky nebo popisování předmětů, jejichž poloha se může snadno změnit. Navrhuji vytvořit dokument s výčtem častých chyb a radami, jak se jim vyhnout a zařadit ho do materiálů v nápovědě.

#### Gramatické a syntaktické chyby

Většina uživatelů dělala při popisu gramatické chyby. Někteří si po sobě vyplněná pole přečetli, několik z nich však chyby přehlédlo a pokračovalo dalším úkolem. Druhou častou chybou bylo psaní skloňovaných číslovek číslicí. V běžném psaném projevu to není problém, ale musíme si uvědomit, že zařízení pro čtení textu displeje mobilu přečte číslovku 1 vždy jako „jedna“, aniž by ji upravil do patřičného tvaru podle pádu. Na všechny tyto chyby je třeba upozornit hned při uložení elementu. Validáčnı́ algoritmus by měl tato pole označit a nepustit uživatele dále, dokud chyby neopravı́.

#### Používání internı́ch označení (vstup, element) v popisu pro ZP

Jeden uživatel se pokoušel včlenit do popisu elementu označení vstup a element. Proto je třeba upozornit uživatele na to, že tato označení se používají pouze k zpřehlednění popisu autorovi a do popisu určeného ZP rozhodně nepatří.

#### Změna zobrazení vstupů

Při popisu složitějších budov by se mohlo stát, že se popisky vstupů u jejich grafického zobrazení stanou nepřehlednými. Proto bych změnila jejich podobu z barevně vyplněného kruhu na podbarvenou kružnici, obsahující uvnitř číslo vstupu bez prefixu V. Bude tak jasné, ke kterým vstupům číslo patří a vynecháním prefixu V se zvětší prostor pro číslo vstupu o jednu cı́fru.

#### Přidat patro

Pouze jeden uživatel označil, že by uvı́tal volbu přidat patro jako samostatné tlačı́tko.

#### Elementy a vstupy

Během testování se žádný účastník nezmı́nil o tom, že by mu vadily tyto názvy. V dotazníku však čtyři uvedli, že by byl vhodnější jiný název než element, tři byli pro více variant a ostatním současný název nedělal problém. Označení „vstup“ vyhovovalo všem testovaným.

### 7.3.3. Připomínky k formulářům elementů

#### Nejasnost povinných a nepovinných polí

Dva uživatelé se při testování dotazovali, která pole jsou povinná a která ne. Byli zmateni, protože u jednoho pole bylo připsáno „nepovinné“. Z toho logicky vyplývalo, že ostatní pole, která budou nepovinná, budou označena stejným způsobem. Ve formuláři se tedy musí zřetelně označit, zda je pole nepovinné. U diskutabilních případů by se měla pevně zvolit jedna varianta.

#### Vlastnosti schodiště

Ve formuláři schodiště v panelu pro popis vlastností není možnost označit více variant najednou. Například pokud jsou schody strmé a úzké, máme možnost označit jenom jednu variantu. Je proto nutné zvolit takový typ formulářového pole, který by umožňoval výběr více možností najednou.

#### Počet schodů

Tři účastníci při testování zmínili, že by bylo vhodné předdefinovat jako vlastnost schodiště počet schodů. Jedná se o detailní informaci, kterou mohou někteří ZP uvítat, jiné může naopak zdržovat. Pokud bude toto pole předdefinováno, tak pouze jako nepovinná informace.

#### Výtah

Výtah jezdí v budově většinou do všech podlaží, proto by bylo rychlejší, kdyby byly v panelu pro označení podlaží, která výtah spojuje implicitně označeny všechny možnosti.

#### Neutrální volba

Každé formulářové pole, které obsahuje výběr z několika možností, by mělo zahrnovat možnost „nezvoleno“. Zamezí se tak tomu, že uživatel bude muset vybrat jednu z možností, i když ani jedna nebude odpovídat realitě.

#### Index obtížnosti

Současný stav návrhu vztahuje index obtížnosti k celému elementu. Pro zpřesnění určení obtížnosti by se měl tento index vztahovat ke konkrétním průchodům elementem. Je totiž velký rozdíl, jestli projdeme složitou chodbu celou, nebo zahňeme do prvních dveří.

### 7.3.4. Zajímavé postřehy uživatelů

Mezi připomínkami testů se objevil i návrh, aby navigace oslovovala nevidomého jménem, bylo by to pro ZP více osobní. Pokud bude uživatel tvořil popis budovy pro jednu konkrétní osobu, nebyl by žádný problém, aby do popisu průchodu zařadil oslovení.

## 8. Závěr

Účelem mojí práce byl návrh uživatelského rozhraní aplikace, pomocí které by se daly vytvářet popisy budov pro již existující aplikaci NaviTerier. Cílovou skupinou uživatelů byly osoby blízké, tedy kamarádi a dobrovolníci z pomáhajících profesí. Testováním s uživateli se měla ověřit jednoduchost a intuitivnost ovládání, na které byl kladen velký důraz.

### 8.1.1. Zhodnocení

Na základě prozkoumání aplikace NaviTerier a výsledků jejího testování[2] jsem navrhla uživatelské rozhraní aplikace určené na tvorbu mapových podkladů pro navigaci NaviTerier. Vzhled a funkce aplikace jsem přizpůsobila požadavkům uvedeným v diplomové práci o této navigaci[2]. Podle návrhu jsem vytvořila částečně funkční prototyp a otestovala ho s uživateli. Výsledky testů byly příznivé. I přestože se každý účastník musel nejdříve seznámit s problematikou tvorby map pro NaviTerier, v průběhu testu se všichni dříve nebo později zorientovali a ke konci testování prováděli úkony s větší jistotou a rychleji. Návrh i testování hodnotili uživatelé kladně a projevíli zájem o pomoc při vývoji a dalším testování této aplikace, což považuji za nejdůležitější výsledek této práce.

### 8.1.2. Přístupnost aplikace přímo z internetu

Při návrhu aplikace jsem zvažovala, zda by bylo vhodné navrhnout tuto aplikaci jako webovou aplikaci, kdykoliv přístupnou ze všech počítačů s připojením na internet. Nevýhodou by byla samozřejmě nutnost připojení. Jedinou výhodou je v tomto případě úspora času a problémů s instalací a dostupnost ze všech počítačů s připojením na internet. Doba pro instalaci programu je ale zanedbatelná v porovnání s tím, kolik času uživatel stráví nad tvorbou kvalitní mapy rozsáhlé budovy a je pravděpodobné, že daných map vytvoří pro svou blízkou osobu, která používá navigaci NaviTerier, více. Čas instalace pak bude zanedbatelný. Pokud by byla aplikace kdykoli přístupná z internetu, mohli by někteří lidé, navštívit odkaz na aplikaci pro tvorbu map a rozhodnout se pomoci. Během tvorby popisu by však mohli ztratit zájem a zbytek popisu by nezpracovali kvalitně. Při instalaci aplikace je více času na rozmyšlení, zda za to dané úsilí stojí, než když je aplikace přístupná pomocí jednoho kliknutí na odkaz.

### 8.1.3. Další testování

Další iterace testů by se měla zaměřit na popis architektonicky složitých budov. Testování uživatelé by vytvořili sérii popisů. Tyto popisy by prošly kontrolou a poté by byly zařazeny do testování se ZP. Další vývoj uživatelského rozhraní by se měl soustředit na přizpůsobení funkcionality požadavkům, které budou kladeny na systém při vývoji trasy hledacího algoritmu aplikace NaviTerier.

#### **8.1.4. Datový výstup**

Výstupní data programu budou dále zpracovávána navigací NaviTerier. Proto je třeba zvolit vhodnou formu, která by umožnila mezi popisy elementů vyhledávat cesty a hodnotit jejich náročnost.

Jako výstup validačního algoritmu bych zvolila XML (eXtended Markup Language) nebo jiný značkovací jazyk schopný jednoduše zaznamenat strukturu a hierarchii dat.

#### **8.1.5. Propagace NaviTerieru**

Zařazení mobilních technologií do řešení problematiky samostatného pohybu nevidomých v budovách je velmi dobrá myšlenka. K rozšíření povědomí o této navigaci by mohlo pomoci zařazení NaviTerieru do seznamu dostupných kompenzačních pomůcek pro nevidomé na oficiálních stránkách SONS [4] nebo jiných nevidomými často navštěvovaných stránkách.

## 9. Reference

- [1] PILMANOVÁ, Lenka. *Systém pro navigaci nevidomých uvnitř budov.*, 2007.  
České vysoké učení technické v Praze. Fakulta elektrotechnická. Katedra výpočetní techniky. Vedoucí diplomové práce Ing. Zdeněk Míkovec, Ph.D.  
Dostupný z WWW:  
<[https://dip.felk.cvut.cz/browse/pdfcache/pilma1\\_2008dipl.pdf](https://dip.felk.cvut.cz/browse/pdfcache/pilma1_2008dipl.pdf)>.
- [2] VYSTRČIL, Jan. *Uživatelské rozhraní pro navigaci nevidomých osob v budovách pomocí mobilního telefonu.*, 2008.  
České vysoké učení technické v Praze. Fakulta elektrotechnická. Katedra výpočetní techniky. Vedoucí diplomové práce Ing. Zdeněk Míkovec, Ph.D.  
Dostupný z WWW:  
<[https://dip.felk.cvut.cz/browse/pdfcache/vystrj1\\_2008dipl.pdf](https://dip.felk.cvut.cz/browse/pdfcache/vystrj1_2008dipl.pdf)>.
- [3] MAREŠOVÁ, Petra. *Uživatelské rozhraní pro autoring tool aplikace NaviTerier.*, 2009  
České vysoké učení technické v Praze. Fakulta elektrotechnická. Katedra výpočetní techniky.
- [4] SONS : *Sjednocená organizace nevidomých a slabozrakých ČR* [online].  
Praha : 2002-2007 [cit. 2009-22-05]. Dostupný z WWW:  
<<http://www.sons.cz/index.php>>.
- [5] WHO : *World Health Organization, Světové zdravotnické organizace v ČR*  
Dostupný z WWW: < <http://www.who.cz/> >.
- [6] *Wikipedia: The Free Encyclopedia: Framework.* [online].  
2010 [cit. 2010-05-25].  
Dostupný z WWW: < <http://cs.wikipedia.org/wiki/Framework>>.
- [7] FOLWER, AMY. *Painting in AWT and Swing* [online].  
Dostupný z WWW:<http://java.sun.com/products/jfc/tsc/articles/painting/>
- [8] JANÍČKOVÁ, Gabriela. *Prostorová orientace a samostatný pohyb lidí s vážně postiženým zrakem.*,2008 Masarykova universita. Fakulta humanitních věd. katedra speciální pedagogiky.  
Dostupný z WWW:  
[http://is.muni.cz/th/92766/pedf\\_m/Prostorova\\_orientace\\_a\\_samostatny\\_pohyb\\_lidi\\_s\\_vazne\\_postizenym\\_zrakem.pdf](http://is.muni.cz/th/92766/pedf_m/Prostorova_orientace_a_samostatny_pohyb_lidi_s_vazne_postizenym_zrakem.pdf)
- [9] SNEIDERMAN, Ben. PLAISANT, Catherine. *Designing the user interface.*  
2005. 606 s. ISBN 0-321-53735



- [10] BLOCH, Josua. *Effective Java (2nd Edition)*. 2008  
ISBN 0-321-355683
- [11] RUBIN, Jeffrey. CHISNELL, Dana. *Handbook of Usability Testing: Howto  
Plan, Design, and Conduct Effective Tests* .2008  
ISBN 978-0-470-18548-3

## **A Seznam použitých zkratk**

<b>ZP</b>	zrakově postižený
<b>IT</b>	informační technologie
<b>SW</b>	software
<b>LCD</b>	displej z tekutých krystalů (Liquid crystal display)
<b>GPS</b>	Global Positioning System
<b>VŠ</b>	vysoká škola
<b>VOŠ</b>	vyšší odborná škola
<b>SOU</b>	střední odborné učiliště

## B Varianty rozložení panelů



Obr. 27 Varianta B



Obr. 28 Varianta C

## C Low fidelity prototyp

nový projekt otevřít uložit

zvolte patro přízemí

přidat element přidat makro zpět vpřed

chodba místnost schodiště dveře výtah

element umístíte kliknutím na bílou plochu

náplavěda

element chodba

délka v metrech:

šířka v metrech:

podlaha:  koberec  parkety  kámen  lino

doplňující informace:

popis průchodu ze vstupu  do vstupu

uložit

načíst vzorovou mapu

## D High fidelity prototyp

The screenshot shows a software interface for creating a high-fidelity prototype of a staircase. The interface is divided into several functional areas:

- Top Bar:** Contains menu options: "Nový prvek", "Uložit", "panel menu", and "Nápověda".
- Left Panel (panel nápověda):** A text box with the instruction: "element umístíte kliknutím na kreslici plochu." Below it are buttons for "příběh", "příběh", "příběh", and "příběh".
- Bottom Left Panel (panel elementy):** A list of element types: "příběh", "chodba", "místnost", "schodiště", "dveře", "výťah", and "makro".
- Bottom Right Panel (panel makro):** A list of macro actions: "nové makro", "upravit makro", and "zrušit makro".
- Central Workspace:** Displays a diagram of a staircase with various elements labeled: "chodba 2", "v10", "v4", "v7", "v8", "v9", "chodba", "v3", "v7", "v8", "v9", "místnost K11.2", "v6", and "v1".
- Right Panel (panel formulář elementu):** A configuration panel for the selected element, titled "element CHODBA". It includes:
  - Fields for "název elementu" (set to "chodba"), "přibližné rozměry" (width, height, depth), and "počet vstupů (tov. poutavě)".
  - Buttons for "nahoru", "dolů", "vlevo", and "vpravo".
  - Checkboxes for "povrch podlahy (zahrnout)", "koberec", "parquet", "beton", "lino", and "dlažba".
  - A "doplňující informace" text area.
  - Buttons for "průchod chodbou", "index obdĺnosti průchodu", "hledat při vstupu", and "vede k němu vstup(y)".
  - Buttons for "ok", "cancel", and "ok".

## E zadání testu s uživateli

- Pročtěte si informační materiály “Než začneme“.
- Seznamte se s prostředím programu Mapping a prohlédněte si stručnou dokumentaci, kterou budete mít v průběhu testu k dispozici jako nápovědu.
- Prohlédněte si obrázkovou a mapovou dokumentaci k testu
- Pročtěte si profil fiktivního kamaráda, pro kterého budete tvořit mapu.
- Můžete začít s testováním.

**Situace:** .. *jméno přidělené persony* ... má jít na pracovní pohovor ke slečně C., která má kancelář K112.

**Úkol:** Popište pro ... *jméno přidělené persony* ... vybranou část budovy, podrobně zobrazenou v obrázkové dokumentaci pomocí testovaného prototypu. V první části vytvořte jednotlivé prvky, v druhé části spojte vstupy prvků. Po celou dobu práce sledujte řádek s nápovědou a pokud si nebudete jistí, podívejte se do dokumentace (tutoriálu).

**Rada:** Při testování mluvte nahlas o tom, o čem přemýšlíte. Pokud nevíte, v jakém tvaru popsat nebezpečí, popisy průchodů nebo doplňující informace, představte si, že stojíte s nevidomým kamarádem na popisovaném místě a větu formulujte tak, jak byste ji řekli Vašemu kamarádovi. Uvědomte si, že všechny věty, co napíšete, budou nevidomému přečteny a musí dávat smysl.



## 1.část

1. Vytvořte a umístěte prvek **chodba** .
2. Zaznamenejte důležité údaje o chodbě.



### Popis chodby:

Chodba je zhruba 10 metrů dlouhá, 4 metry široká. Naproti schodišti je napevno zabudovaná sedačka a věšák na odložení svršků. Podlahu tvoří dlažba, která trochu klouže. Díky oknům dopadá do místnosti hodně světla. Po levé straně je výtah. Po pravé straně je v podlaze zabetonovaný cihlový květináč s fíkusem, jehož listy vyčnívají do prostoru. Květina je vysoká skoro dva metry. Za květinou je jeden schod, nalézá se zhruba v jedné čtvrtině chodby a je poměrně blízko přede dveřmi. Za ním jsou dveře do kanceláře K112.



3. Popište jeden ze šesti průchodů chodbou (je označen na mapce oranžovou šipkou a vede od schodiště k místnosti)

4. Vytvořte a umístěte prvek **schodiště**.
5. Zaznamenejte důležité údaje o schodišti

### Popis schodiště:

Schodiště je přímé, asi 4 metry dlouhé a 3 metry široké, celkem pozvolné se širokými schody. Zábradlí je pouze po pravé straně při průchodu směrem dolů do chodby s květinou. Schodiště je dřevěné. Na obou stranách spojuje poschodí „přízemí“.



6. Popište jeden průchod schodištěm ve směru cesty shora dolů.

7. Vytvořte a umístěte prvek **výtah**.
8. Zaznamenejte o něm důležité údaje.



**Popis výtahu:** Výtah má dvoukřídlé dveře, které se otevírají automaticky při příjezdu výtahu.

Tlačítko na přivolání výtahu je vpravo vedle dveří asi metr a půl nad zemí. Příjezd není oznámen akustickým znamením, ale samotné otevření dveří je dobře slyšet. Tlačítka na ovládacím panelu zespodu nahoru jsou: „zvonek, stop, suterén, přízemí, první patro, druhé patro, třetí patro“. Panel pro ovládání výtahu se nachází v kabině výtahu, vpravo za dveřmi.

9. Vytvořte a umístěte na mapě prvek **dveře**.  
10. Zaznamenejte o něm důležité údaje.

**Popis dveří:** Dveře vedou do místnosti K112. Nacházejí se vlevo od schodů. Klika je vpravo a dveře se otevírají tahem směrem do chodby. Jsou označené štítkem. Není u nich žádný zvonek, musí se zaklepat. Slečna C má úřední hodiny od pondělí do pátku vždy od sedmi do dvanácti hodin.



11. Popište jeden ze dvou průchodů dveřmi (ten ve směru z chodby do místnosti).

12. Vytvořte a umístěte prvek **místnost**.  
13. Zaznamenejte o ní důležité údaje.

**Popis místnosti:**

Místnost je pět metrů dlouhá a pět metrů široká. V místnosti sedí slečna C. nebo jiná sekretářka, která při zaklepaní otevře a ujme se klienta. Místnost není průchozí a v nepřítomnosti slečny C. je zamčena.



14. Vytvořte a umístěte prvek **chodba2**.  
15. Zaznamenej důležité údaje o chodbě2.

**Popis chodby:**

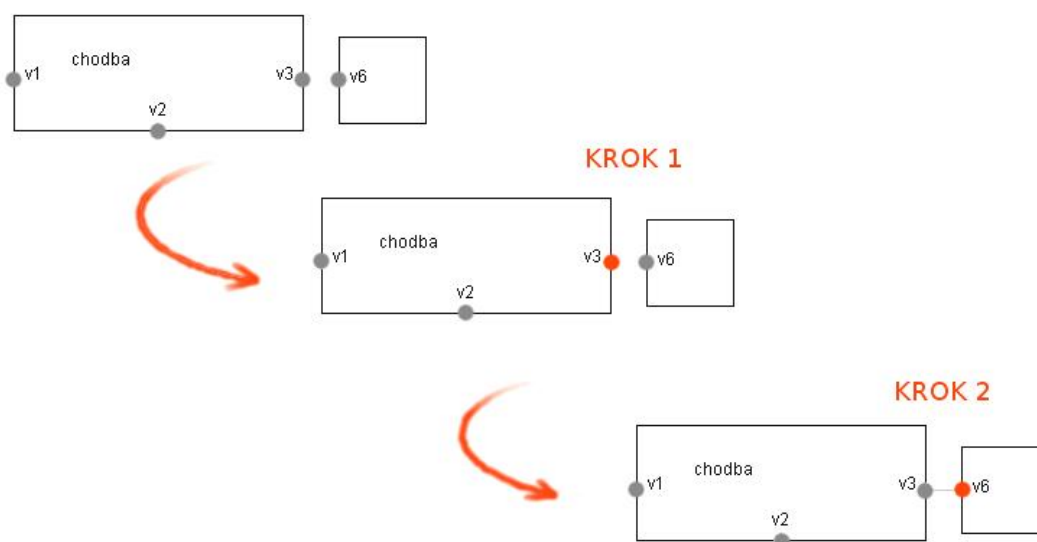
Chodba je zhruba 20 metrů dlouhá, 3 metry široká. Naproti schodišti jsou čtyři dveře. Na podlaze jsou dlaždice.





## 2.část

1. **Pospojte vstupy** jednotlivých elementů tak, aby vytvořily ucelenou část mapy. Dva vstupy spojte tak, že označíte kliknutím první vstup (KROK 1) a pak druhý (KROK 2).  
Mezi vstupy se vytvoří vazba v podobě šedé úsečky.



2. Zkuste přidat další patro

## **F     Obsah příloženého CD**

<b>text/</b>	- bakalářská práce
<b>prototyp/</b>	- komprimovaná složka s testovaným prototypem
<b>test_info/</b>	- seznámení s projektem pro účastníky testu
<b>test_zadani/</b>	- úplné znění zadání pro uživatele testu
<b>test_tutorial/</b>	- tutoriál k prototypu pro testování s uživateli
<b>test_dotazniky/</b>	- vyplněné dotazníky z testování
<b>test_img/</b>	- obrázkové podklady k popisované části budovy