

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Návrh modelu požadavků a testovací analýzy podnikové aplikace

Filip Kříž

© 2021 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Filip Kříž

Systémové inženýrství a informatika
Informatika

Název práce

Návrh modelu požadavků a testovací analýzy podnikové aplikace

Název anglicky

Design of application model and test analysis of business application

Cíle práce

Cílem práce, je analýza uživatelských požadavků a návrh případů užití, uživatelského rozhraní a scénářů testovací analýzy aplikace pro správu kancelářských prostor.

Metodika

1. Na základě studia odborných zdrojů popište problémovou doménu formou literární rešerše
2. Provedte analýzu uživatelských požadavků na podnikovou aplikaci a vytvořte pomocí vhodných nástrojů (Use Case apod.) model požadavků
3. Pro podnikovou aplikaci navrhnete model uživatelského rozhraní
4. Sestavte scénáře testovací analýzy aplikace

Doporučený rozsah práce

30-40 stran

Klíčová slova

podniková aplikace, uživatelské požadavky, uživatelské rozhraní, testovací analýza

Doporučené zdroje informací

KLČOVÁ, H. – SODOMKA, P. *Informační systémy v podnikové praxi*. Brno: Computer Press, 2010. ISBN 978-80-251-2878-7.

SCHUTTA, N T. – ASLESON, R. *Ajax : vytváříme vysoce interaktivní webové aplikace*. Brno: Computer Press, 2006. ISBN 80-251-1285-3.

Předběžný termín obhajoby

2019/20 LS – PEF

Vedoucí práce

Ing. David Buchtela, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 14. 3. 2020

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 14. 3. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 22. 03. 2020

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Návrh modelu požadavků a testovací analýzy podnikové aplikace" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.3.2021

Poděkování

Rád bych touto cestou poděkoval panu Ing. Davidovi Buchtelovi, Ph.D. za vedení a cenné rady, při přípravě této práce.

Návrh modelu požadavků a testovací analýzy podnikové aplikace

Abstrakt

Zadáním práce, je vytvoření návrhu modelu požadavků a testovací analýzy podnikové aplikace. Splnění zadání má být dosaženo díky studiu odborné literatury. V první části práce, jsou popsána teoretická východiska práce. Jsou zde definovány druhy požadavků a způsob, jakým je doporučeno požadavky sbírat. Dalším tématem je definování a způsob vytvoření případů užití a jejich scénářů, které jsou pro tuto práci klíčové. Popis, principy a druhy UML diagramů tvoří obsah další kapitoly. Součástí praktické části této práce je i návrh logiky uživatelského rozhraní, proto je v teoretické části i podklad pro tuto část. Poslední kapitolou teoretických podkladů je testovací analýza, kde jsou situovány informace, nezbytné pro vytvoření testovací analýzy. V praktické části je popsána metodika, použitá pro přípravu těchto analýz. Během požadavkového meetingu byly definovány uživatelské požadavky, nedostatky původní aplikace a vzorové osoby. Díky uživatelským požadavkům mohly vzniknout i případy užití, s jejich podrobnými scénáři. Diagramy návrhu logiky uživatelského rozhraní, vychází také z těchto případů užití. V závěru praktické části je popsán způsob přípravy scénářů pro akceptační testování.

Klíčová slova: podniková aplikace, uživatelské požadavky, analýza požadavků, případ užití, scénář případu užití, diagram případů užití, UML, uživatelské rozhraní, drátěný model grafického rozhraní, testovací analýza, testovací případ

Design of application model and test analysis of business application

Abstract

The assignment of the thesis is a draft application design and test analysis of the business application. Fulfill the assignment should be made possible through the study of specialized literature. The first part of the thesis describes the theoretical background of the thesis. It defines the types of requirements and the ways that it is recommended to collect it. Another topic is defining and applying the use cases and their scenarios, which are crucial for the thesis. Description, principles and types of UML diagrams form the content of the next chapter. The thesis also include design of user interface logic, therefore, in the theoretical parts is the theoretical background for this part. The last chapter of the theoretical background is a test analysis, where are situated the information to create test analysis. The practical part describes the methodology used to prepare these analyzes. During the request meeting, user requirements, shortcomings of the original application and user persona were defined. Thanks to the user user requirements were described Use cases, with their detailed scenarios. UI logic design schema, is also defined on the basis of these use cases. In the conclusion of the practical part there are prepared scenarios for acceptance testing.

Keywords: business application, user requirements, requirements analysis, use case, use case scenario, use case diagram, UML, graphical user interface, wireframe diagram, test analysis, test case

Obsah

1 Úvod.....	11
2 Cíl práce a metodika	12
2.1 Cíl práce	12
2.2 Metodika	12
3 Teoretická východiska	13
3.1 Podnikový informační systém a Podniková aplikace (Podnikový software)...	13
3.1.1 Podnikový informační systém	13
3.1.2 Podniková aplikace (Podnikový software)	13
3.2 Softwarové požadavky	13
3.2.1 Druhy softwarových požadavků	14
3.2.2 Sběr požadavků.....	14
3.2.3 Obecná pravidla pro sbírání požadavků.....	15
3.3 Případy užití	15
3.3.1 Co je Případ užití?.....	15
3.3.2 Obsah případu užití.....	16
3.3.3 Scénář případu užití	17
3.4 UML.....	20
3.4.1 Co je UML?	20
3.4.2 Princip UML	20
3.4.3 Typy UML diagramů	21
3.4.4 Diagram Případů užití.....	21
3.5 Model uživatelského rozhraní	22
3.6 Testovací analýza	24
3.6.1 Co je testování softwaru?.....	24
3.6.2 Cíle testování	24
3.6.3 Statické testování	24
3.6.4 Dynamické testování.....	25
4 Vlastní práce	27
4.1 Úvod.....	27
4.1.1 Popis zadavatelské společnosti	27
4.1.2 Požadavek firmy na vypracování a dodání analýzy.....	27
4.2 Sběr uživatelských požadavků	28
4.2.1 Voba metod získání uživatelských požadavků	28
4.2.2 Požadavkový meeting	29
4.2.3 Pozorování uživatelů při práci	34
4.2.4 Zpracování získaných informací.....	35

4.2.5	Podrobný postup tvorby uživatelských požadavků	36
4.2.6	Výsledek kapitoly	37
4.3	Případy užití a scénáře užití	37
4.3.1	Postup a způsob přípravy případů užití a scénářů užití	37
4.3.2	Volba obsahu případů užití	38
4.3.3	Podrobný postup tvorby případů užití	39
4.3.4	Nesprávný způsob zápisu případů užití	42
4.3.5	Akceptování analýzy zákazníkem	43
4.4	Diagram případů užití	43
4.4.1	Volba software	43
4.4.2	Tvorba diagramu	44
4.5	Model uživatelského rozhraní	46
4.5.1	Postup dle obecné metody návrhu	47
4.5.2	WireFrame	49
4.5.3	Kresba WireFrame	51
4.5.4	Ukázka hotového návrhu wireframe	52
4.6	Uživatelské akceptační testování	52
4.6.1	Tvorba testovacích případů	53
4.6.2	Obsah testovacího případu	54
4.6.3	Podrobný postup tvorby testovacích případů	55
4.6.4	Kontrola pokrytí případů užití testovacími případy	58
4.6.5	Software	59
4.6.6	Výsledky kapitoly	60
5	Výsledky a doporučení	61
5.1	Výsledky	61
5.2	Doporučení	62
6	Závěr	63
7	Seznam použitých zdrojů	64
8	Přílohy	66
8.1	Uživatelské požadavky	67
8.2	Vzorové persony	69
8.2.1	Primární vzorové persony	69
8.2.2	Sekundární vzorové persony	70
8.3	Případy užití a scénáře užití	71
8.4	Model uživatelského rozhraní	80
8.5	Testovací případy uživatelského akceptačního testování	82

Seznam obrázků

Obrázek 1: Obsah případu užití od Martina Fowlera	17
Obrázek 2: Běžná a alternativní cesta	18
Obrázek 3: Přehled UML vztahů	20
Obrázek 4: Ukázka diagramu případů užití	22
Obrázek 5: Ukázka návrhu logiky uživatelského rozhraní	23
Obrázek 6: Poměrné náklady na opravu chyby podle toho, kdy byla objevena	25
Obrázek 7: Rozdělení priorit	31
Obrázek 8: Symbol aktéra	44
Obrázek 9: Symbol systému	44
Obrázek 10: Symbol případu užití	44
Obrázek 11: Diagram případů užití	46
Obrázek 12: Prvek Text field	49
Obrázek 13: Prvek Action button	49
Obrázek 14: WireFrame - Hlavní menu	50
Obrázek 15: Náskres okna pro zobrazení plánu podlaží	51
Obrázek 16: WireFrame - Grafický přehled rezervací	52
Obrázek 17: Přehled pokrytí případů užití testovacími případy	59
Obrázek 18: WireFrame - Přihlašovací okno	80
Obrázek 19: WireFrame - Hlavní menu	81
Obrázek 20: WireFrame - Grafický přehled rezervací	81
Obrázek 21: WireFrame - Zobrazení plánů	82

1 Úvod

V současné době je každá firma více, či méně závislá na informačních systémech. Zároveň, s rostoucími požadavky na pracovní výkonnost a efektivnost, je třeba tyto systémy stále vylepšovat a modernizovat, aby byly efektivněji a snadněji použitelné a firma díky zvýšené produktivitě dosahovala vyššího zisku.

V této bakalářské práci byla provedena analýza základní fáze nového systému pro správu kancelářských prostor, který nahradí původní systém, jenž je celkově zastaralý a společnost s ním nebyla spokojena.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem práce je provést analýzu uživatelských požadavků na podnikovou aplikaci a skrze případy užití vytvořit model požadavků. Následně bude vytvořen návrh uživatelského rozhraní skrze wireframe a vytvořeny testovací případy pro uživatelské akceptační testování.

2.2 Metodika

V první části této bakalářské práce, bude popsán teoretický podklad pro praktickou část. Zdrojem těchto informací bude odborná literatura.

Ve druhé části, bude situována praktická část této práce. Metodika bude založena zejména na studiu odborné literatury, citované v teoretické části.

Ve spolupráci se zadavatelem bude uspořádán požadavkový meeting, kde budou definovány uživatelské požadavky, nedostatky původní aplikace a vzorové osoby. Z uživatelských požadavků budou vytvořeny případy užití, spolu se scénáři popisujícími průběh. Pro lepší přehlednost bude vytvořen diagram případů užití. Za použití případů užití bude vytvořen model uživatelského rozhraní, jehož účelem bude definovat logiku zobrazování prvků aplikace. Dle případů užití bude také vypracována testovací analýza, obsahující scénáře akceptačního testování.

3 Teoretická východiska

3.1 Podnikový informační systém a Podniková aplikace (Podnikový software)

3.1.1 Podnikový informační systém

Podnikový software je podmnožinou podnikového informačního systému, kde sehrává roli právě programového vybavení komplexu. IS tyto webové, desktopové nebo mobilní aplikace spravuje a integruje do uceleného celku, kde s nimi dále manipulují jednotliví pracovníci. (1)

Procesy podnikového informačního systému můžeme rozdělit do tří kategorií:

1. Řídící procesy (strategické plánování, řízení kvality a inovací) – zabezpečují rozvoj a řízení výkonu společnosti a vytvářejí podmínky pro fungování ostatních procesů.

2. Hlavní procesy (výroba, logistika, řízení vztahů se zákazníky) – vytvářejí hodnotu v podobě výrobku nebo služby pro externího zákazníka, jsou tedy součástí hodnototvorného řetězce organizace.

3. Podpůrné procesy (ekonomika, řízení lidských zdrojů, IT) – zajišťují podmínky pro fungování ostatních procesů tím, že jim dodávají hmotné i nehmotné výstupy, přitom ale nejsou součástí hodnototvorného řetězce. (2, s. 43)

3.1.2 Podniková aplikace (Podnikový software)

Podnikový software, někdy též enterprise software, jsou stručně řečeno všechny programy a webové aplikace využívané v rámci firmy nebo společnosti, které podporují a zefektivňují řízení či fungování podniku. Podílejí se na řízení interních procesů, správě zdrojů, skladů nebo docházky, objednávek u dodavatelů či na přípravě mzdy. (3)

3.2 Softwarové požadavky

Obecně, lze rozdělit požadavky na software do čtyř skupin: podnikatelské, uživatelské, funkční a nefunkční.

3.2.1 Druhy softwarových požadavků

Podnikatelské požadavky

Podnikatelské požadavky z obecného úhlu pohledu popisují cíle, které by aplikace měla splňovat a definují tak i to, proč vlastně daná aplikace má vzniknout.

Uživatelské požadavky

Uživatelské požadavky konkrétněji definují cíle, jaké má být uživatel schopen prostřednictvím aplikace dosáhnout. Obvykle jsou zapisovány pomocí Případů užití, které jsou v této práci podrobněji popsány v následující kapitole.

Funkční požadavky

"Funkční požadavky popisují softwarovou funkcionalitu, kterou musí vývojáři do systému dostat, aby uživatelé mohli splnit své úkoly, a tím i podnikatelské požadavky." (4, s. 30).

Většinou se jedná o jednoduché formulace, typu "Uživatel se může z aplikace odhlásit." nebo "Po dokončení registrace je uživateli odeslán informativní e-mail."

Nefunkční požadavky

Prostřednictvím nefunkčních požadavků jsou popisovány například nároky na bezpečnost či výkonnost.

3.2.2 Sběr požadavků

Postup pro sběr požadavků je vždy více či méně individuální pro daný projekt. Lze nalézt mnoho obecných doporučení, jak postupovat. Při objednávce aplikace jsou již základní očekávání na aplikaci nastíněna, je třeba je však prozkoumat a definovat mnohem podrobněji. Odborné zdroje doporučují například svolat několikahodinový workshop, kterého se účastní všichni zadavatelé, případně i lidé, kteří budou aplikaci využívat. Všichni si tak mohou společně projít všechny zatím definované způsoby použití aplikace a pravděpodobně vymyslí i další a přidají spoustu námětů na její zefektivnění. V případě, že podobná aplikace již existuje, mohou analytici sledovat uživatele při jejím užívání. Lze postupovat i takovou formou, kdy lidé na workshopu pracují s papírem, pastelkami a

nůžkami. Pomocí vystřihování a nalepování papírků vytvoří základní podobu aplikace. Díky této, odlehčené metodě, lze dosáhnout velmi zajímavých výsledků. (4, s. 58, 59).

3.2.3 Obecná pravidla pro sbírání požadavků

Tato pravidla jsou popsána Karlem Wiegersem:

Dohodnout si základní pravidla – Například projednávání vždy jen jednoho tématu, zapojení všech účastníků či kritizovat návrhy, nikoli lidi.

Držet se stanoveného obsahu – Ve skupinových diskuzích se často stává, že řeč přesune mimo aktuálně projednávaný prostor. Je tedy třeba tento prostor jasně vymezit a dodržovat. Také je třeba neodbíhat do zbytečných podrobností.

Určité informace si uchovat na později – Během diskusí se občas objeví informace, které v danou chvíli nejsou relevantní, mohou být ale velmi důležité například pro jinou část aplikace.

Omezování se časovým limitem – Pro každý bod diskuse se doporučuje stanovit časový limit a hlídat jeho dodržení. Díky tomu se nestane, že celé setkání v určitém bodě uvázne.

Pečlivě vybírat účastníky a držet skupinu co nejmenší – Čím je skupina menší, tím bývá efektivnější. Ve velké skupině hrozí, že většina účastníků bude pasivní.

Zapojovat všechny účastníky – Je důležité, aby nenastal stav, kdy například mluví jen dva lidé. Pokud do diskuse někteří účastníci přestanou přispívat, ať už z toho důvodu, že jsou otráveni či mají pocit, že nejsou bráni dost vážně, je třeba tyto účastníky zapojit zpět do diskuse. (4, s. 116, 117).

3.3 Případy užití

3.3.1 Co je Příklad užití?

Příklad užití popisuje určitý cíl, jakého by uživatel měl mít možnost dosáhnout, skrze popisovaný software.

"Případy užití poněkud mění perspektivu sbírání požadavků a snaží se zjistit, čeho potřebují dosáhnout uživatelé - tradiční způsob se naproti tomu snažil zjistit, co by podle názoru uživatelů měl dělat systém." (4, s. 128)

Ideálně by měly být skrze Případy užití pokryty všechny činnosti, které uživatel může se softwarem dělat. V praxi bohužel většinou není možné pokrýt všechny případy, analytici se tedy snaží pokrýt co nejvíce možností.

V Případech užití se střídají dvě hlavní role. První z nich je role Aktér, která představuje uživatele aplikace. Druhou rolí je Systém. Scénář případu užití zachycuje jejich vzájemné interakce od počátku činnosti až po dosažení cíle. Vzniká tak jakýsi příběh, o použití určité softwarové funkce.

3.3.2 Obsah případu užití

Existuje několik pohledů, na definování podoby obsahu Případu užití.

Karl E. Wiegers popisuje tento obsah:

- Jedinečný identifikátor
- Stručný název uživatelského úkolu, například Podání objednávky
- Krátký textový popis úkolu v přirozeném jazyce
- Vstupní podmínky, které musí být splněné před začátkem případu
- Výstupní podmínky, které popisují stav systému po úspěšném skončení případu
- Číslovaný seznam kroků, na kterých je vidět posloupnost dialogů nebo interakcí mezi aktérem a systémem, a jež vedou od vstupních podmínek k výstupním. (4, s. 130)

12

Naproti tomu Martin Fowler popisuje obsah Případu použití jinak:

"Neexistuje žádný standartní způsob, jak zapisovat obsah případu užití, a v různých případech jsou dobře použitelné různé formáty. Obrázek 1 ukazuje obecný způsob, jak lze případ užití zapsat." (5, s. 104)

Obrázek 1: Obsah případu užití od Martina Fowlera

Zakoupení Produktu

Úroveň cíle: úroveň mořské hladiny

Hlavní úspěšný scénář:

1. Zákazník brouzdá katalogem a vybírá si položky, které si chce koupit
2. Zákazník uzavře objednávku
3. Zákazník vyplní detaily k dodání (adresu; termín dodání – další den nebo v průběhu 3 dní)
4. Systém zobrazí plné informace o ceně, včetně přepravného
5. Zákazník vyplní informace o kreditní kartě
6. Systém autorizuje objednávku
7. Systém ihned potvrdí objednávku
8. Systém pošle zákazníkovi potvrzující e-mail

Rozšíření:

3.a: Zákazník je pravidelným zákazníkem

- .1: Systém zobrazí aktuální informace o dodání, o tvorbě cen a o fakturaci
- .2: Zákazník může tyto implicitní informace buďto potvrdit, nebo přepsat, pak návrat do MSS do kroku 6

6.a: Systém selhal při pokusu autorizovat platbu kreditní kartou

- .1: Zákazník může znovu zapsat informace o kreditní kartě, nebo může vše stornovat

Zdroj: (5, s. 105)

Alistair Cockburn vytvořil podrobnější návod, kde oproti Wiegeresovi přidává například Zúčastněné strany a Zájmy, Spouštěč či Související informace. (6, s. 65)

3.3.3 Scénář případu užití

Dle popisu od Fowlera, by každý případ užití, s výjimkou velmi jednoduchých případů, měl obsahovat scénář, díky kterému jsou zaznamenány postupné interakce mezi uživatelem a systémem. K textovému popisu scénáře, je možné přidat i schematické diagramy GUI (Graphic User Interface, Grafické uživatelské rozhraní), které zlepší pochopení dané akce.

Martin Komárek popisuje tři základní typy scénáře:

Hlavní scénář: (Main/Basic/Happy flow) - Popisuje kroky případu užití, pokud vše jde jak má.

Alternativní scénář: (Alternative flow) - Zachycuje odchylky od hlavního scénáře.

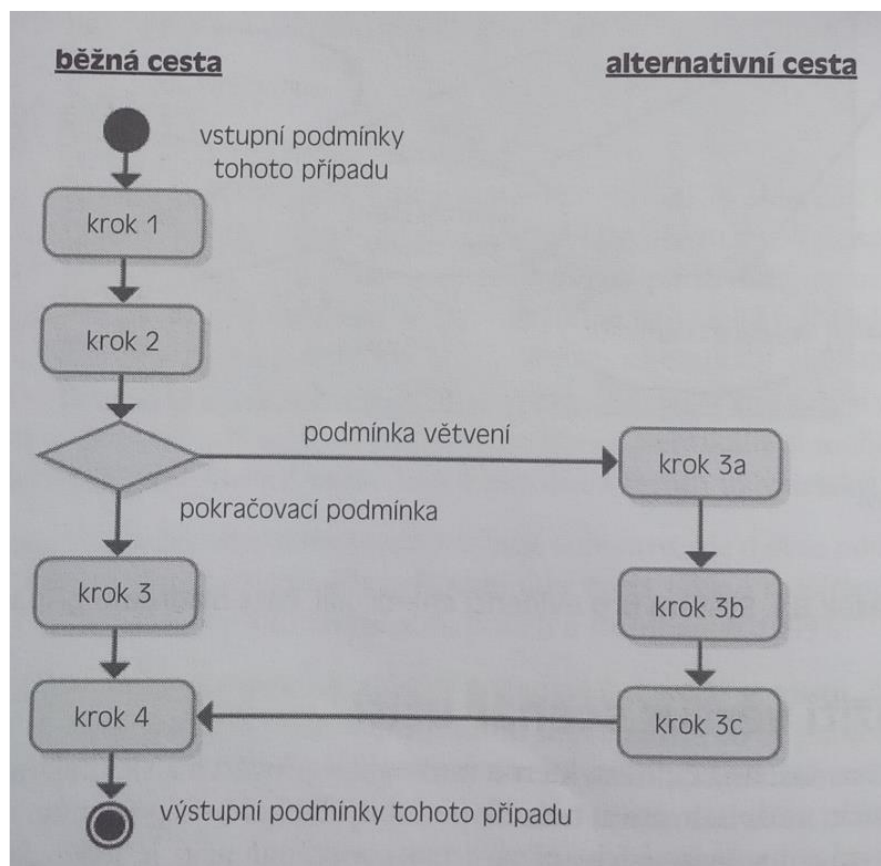
Výjimečný scénář: (Exceptional flow) - Popisuje varianty, kdy se nepodaří dosáhnout cíle případu užití. (7, s. 13)

Dle Komárka není doporučeno ve Scénářích případu užití používat názvy rolí. Pokud je tedy například Případ užití určen pouze pro roli Administrátor, je třeba toto uvést ve Vstupních podmínkách. Poté je Aktér označován vždy jako Uživatel. (7, s. 16)

Alternativní scénář:

Některých cílů se dá dosáhnout více způsoby. Například zobrazení určité informace, může aktér dosáhnout kliknutím na tlačítko na určité obrazovce a zároveň stejného výsledku docílí kliknutím na tlačítko situované na jiné obrazovce. Graficky je alternativní cesta naznačena na obrázku 2.

Obrázek 2: Běžná a alternativní cesta



Zdroj: (4, s. 130)

Vložení - Include

Stává se, že některé kroky se v různých Případech užití často opakují. Lze tedy postupovat tak, že pro opakující se kroky, se vytvoří samostatný Případ užití, který do jiných Případů užití vložíme. Viz. kapitola Diagram případů užití.

Rozšíření – Extend

Jedná se o určité rozšíření scénáře, které pro základní průchod není nezbytné.

3.4 UML

3.4.1 Co je UML?

UML (Unified Modeling Language, česky Sjednocený modelovací jazyk) je grafický jazyk, používaný při návrzích a implementaci softwaru. Jeho první verze se objevily v 90. letech, z důvodu rostoucí potřeby sjednocení mnoha používaných metod grafických diagramů. (5, s. 23)

3.4.2 Princip UML

UML je tvořeno třemi základními prvky. Jedná se o Předměty, Vztahy a Diagramy. (8)

1. Věci, předměty (Things)

Jedná se o samostatné předměty v modelu.

2. Vztahy (Relationships)

Vztahy představují propojení jednotlivých předmětů. Existuje více typů vztahů – obrázek 3.

Obrázek 3: Přehled UML vztahů

Typ relace	Syntaxe UML zdroj cíl	Stručný popis
Závislost (Dependency)>	Změna v určitém předmětu ovlivňuje význam závislého předmětu.
Asociace (Association)	—————	Popis množiny spojení mezi objekty.
Agregace (Aggregation)	◊—————	Cílový prvek je součástí zdrojového prvku
Kompozice (Composition)	◆—————	Silnější forma agregace (má více omezení)
Ochranná nádoba (Containment)	⊕—————	Zdrojový prvek obsahuje cílový prvek
Zobecnění (Generalization)	—————>	Jeden prvek je specializací jiného prvku a lze jej nahradit obecnějším (univerzálnějším) prvkem.
Realizace (Realization)>	Asociace mezi klasifikátory, kde jeden klasifikátor určuje dohodu, jejíž uskutečnění zaručuje druhý klasifikátor

Zdroj: (9, s. 36)

3. Diagramy (Diagrams)

Diagramy obsahují Předměty a vztahy. Celkově zachycují určitý průběh toho, co se v systému děje. Existuje několik základních druhů diagramů.

3.4.3 Typy UML diagramů

Existuje několik oficiálních typů UML diagramu. Například Diagram aktivit, používaný pro definování logiky průchodu procesů, Stavové diagramy pro znázornění aktuálního stavu systému, či Sekvenční diagram zobrazující komunikaci mezi různými objekty. V této práci byl použit Diagram případů užití, který je popsán dále podrobněji.

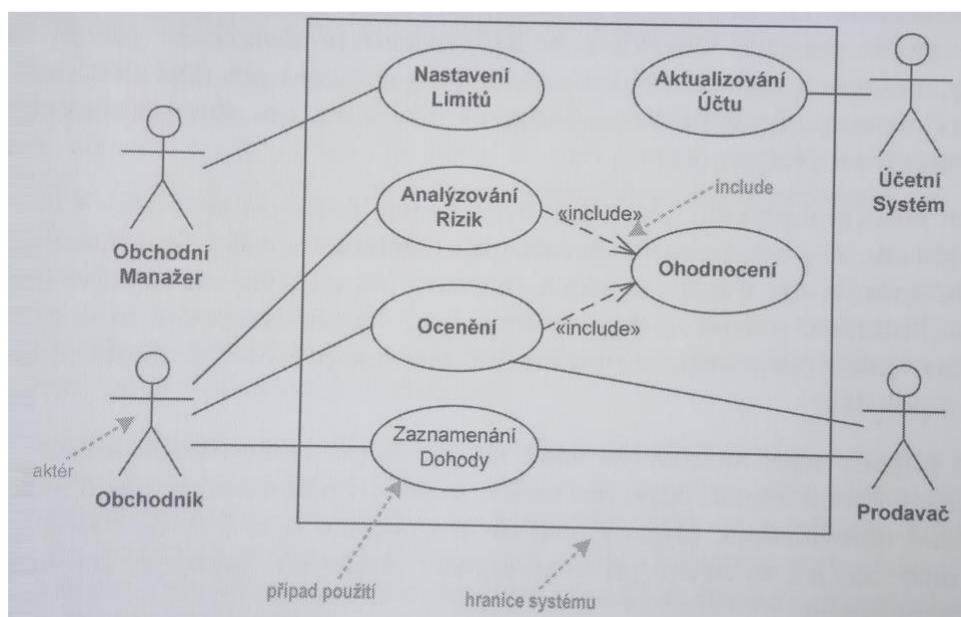
„Neexistuje žádné pevně stanovené pořadí, v němž byste měli své diagramy UML vytvářet. Přesto se obvykle začíná diagramem případu užití, který definuje rozsah platnosti navrhovaného systému.“ (9, s. 37)

3.4.4 Diagram Případů užití

Jak již název tohoto diagramu napovídá, pracuje se zde s Případy užití. Ty jsou zde zobrazeny všechny, spolu s Aktéry, kteří je používají. Použití Případu užití Aktérem je naznačeno čarou, která je spojuje. Aktérů je většinou více, jelikož různí aktéři představují různé uživatelské role. Šipka mezi aktéry může znázorňovat vztah dědičnosti, například aktér Administrátor dědí všechny Případy užití aktéra Uživatel. Samotné Případy užití jsou umístěny do čtverce či obdelníku, který zobrazuje hranice systému. V jednom diagramu může být i více systémů. (10)

Příklad Diagramu případů užití představuje obrázek 4.

Obrázek 4: Ukázka diagramu případů užití



Zdroj: (5, s. 106)

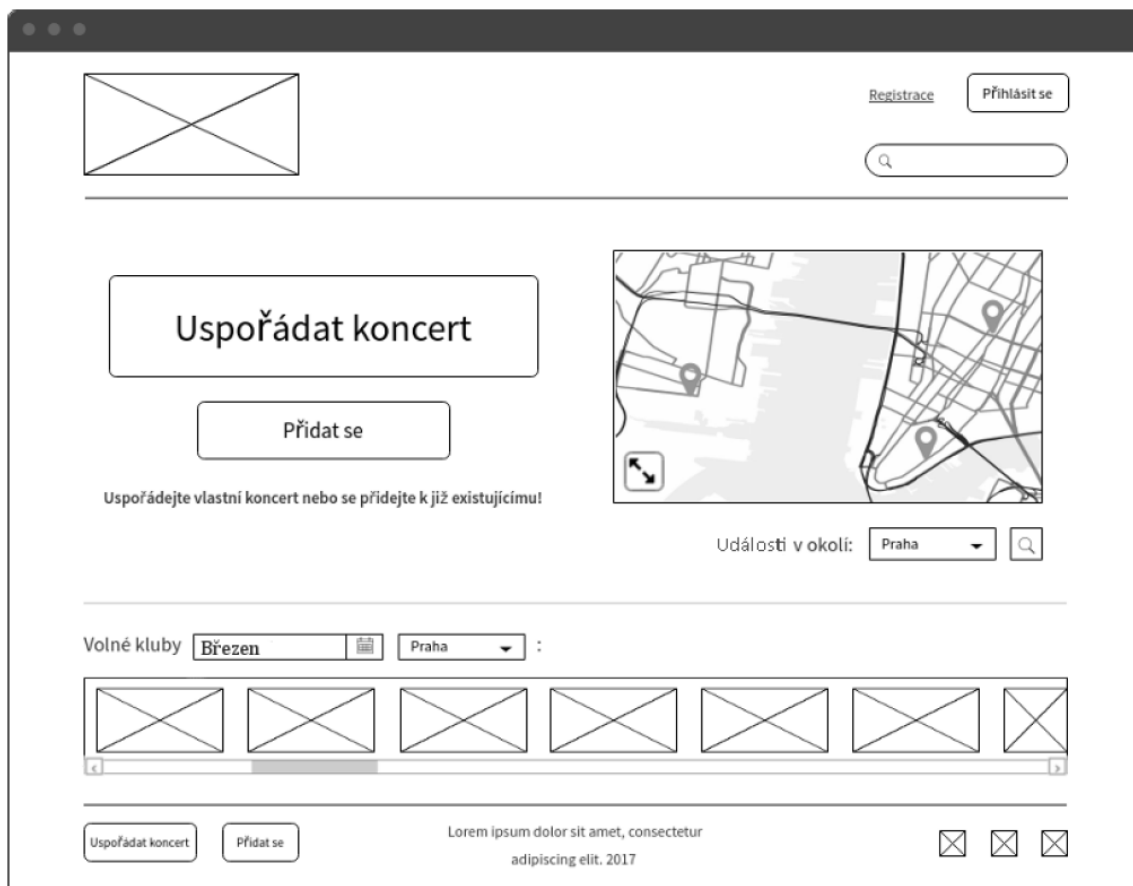
3.5 Model uživatelského rozhraní

Návrhy modelu uživatelského rozhraní se vytvářejí jako drátěné modely (anglicky Wire Frame). Smyslem není zachytit grafickou podobu, ale logické rozmístění obsahu stránky – tedy informace a ovládací prvky.

Modely pokud možno neobarvujeme, a pokud ano, jen barvami zvýrazňující logiku chování, nikoliv aby model představoval budoucí grafický vzhled. Modely se snažíme držet v „šedém“ schématu. Tím dosáhneme popisu umístění jednotlivých ovládacích prvků, i jejich chování, nenutíme ale čtenáři jejich barevné schéma. (11, s. 62)

Na obrázku 5 je vidět příklad návrhu logiky uživatelského rozhraní.

Obrázek 5: Ukázka návrhu logiky uživatelského rozhraní



Zdroj: (11, s. 69)

3.6 Testovací analýza

3.6.1 Co je testování softwaru?

Testování softwaru je pojem, označující širší množství činností. Patří mezi ně vytvoření test analýzy (tedy plán, jak bude testování probíhat), příprava testovacích dat, exekuce testů, reportování nalezených chyb a po opravě i jejich retestování.

3.6.2 Cíle testování

S výjimkou extrémně jednoduchých aplikací, není možno aplikaci otestovat zcela kompletně. Důvodem je to, že i u méně složitých aplikací je počet vstupů a výstupů příliš velký. (12, s. 34).

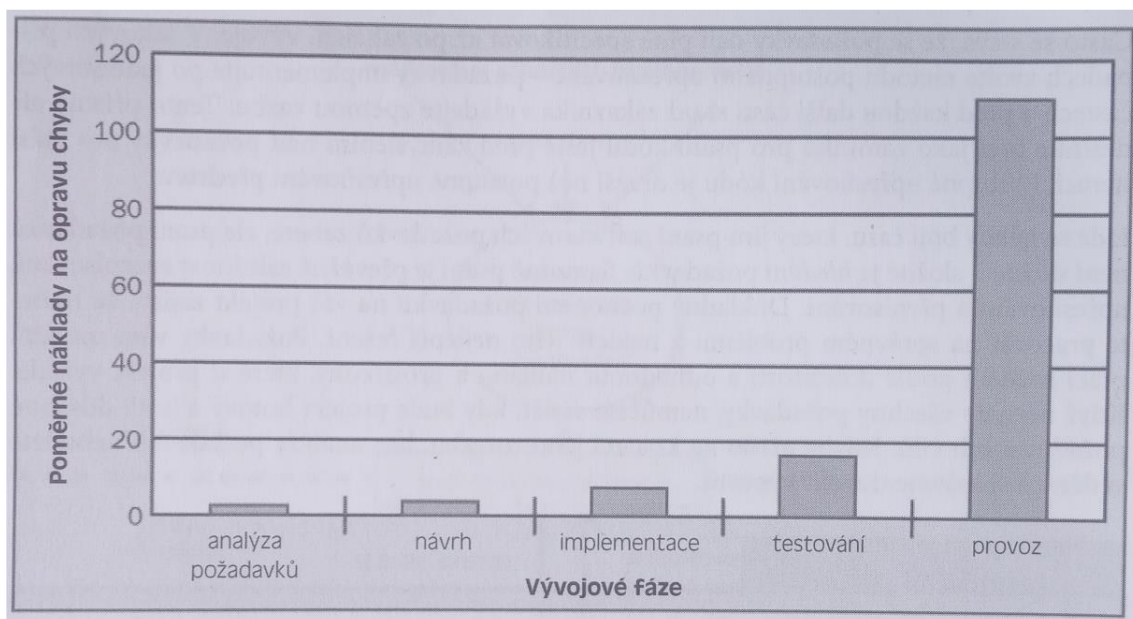
Obecně nelze určit jeden obecný cíl testování, jelikož cíle se liší v rámci různých druhů testů. Například u statického testování je cílem především opravit chyby vzniklé již na úrovni analýzy a snížit tak riziko nákladných oprav ve fázi programování či pilotního provozu.

„Během akceptačního testování může být jedním z cílů potvrzení, že systém funguje podle očekávání a splňuje požadavky. Dalším cílem akceptačního testování může být poskytnutí informací zúčastněným stranám týkající se rizika vydání v daném časovém okamžiku.“ (13, s. 13)

3.6.3 Statické testování

Statické testování označuje takové testování, kdy nedochází ke spuštění kódu. Zpravidla je prováděno ještě před zahájením samotného programování aplikace. Jde tedy například o využití nástrojů pro validaci kódu. Pokud tedy má vývojář v programovacím prostředí aktivní validaci právě psaného kódu, jedná se o statické testování. Statické testování zahrnuje i kontroly specifikací, dokumentace, ale i testwaru, tedy testovací analýzi a skriptů. Hlavním přínosem statického testování je možnost výrazně rychlejší a levnější opravy nalezených chyb – viz. obrázek 6.

Obrázek 6: Poměrné náklady na opravu chyby podle toho, kdy byla objevena



Zdroj: (4, s. 36)

3.6.4 Dynamické testování

K Dynamickému testování, je již třeba spuštění kódu.

Existuje několik obecných testovacích metod.

Třídy ekvivalence – Metoda určená pro části aplikace, kde jsou zadávána data. Test analytik připraví množiny dat, která dosahují stejného, tedy ekvivalentního výsledku. Pokud například aplikace vyžaduje zadání číselné hodnoty $\langle 0;100 \rangle$, vznikají Platné třídy $\langle 0;100 \rangle$ a Neplatné třídy $(-\infty;0)$ a $(100;\infty)$.

Analýza hraničních hodnot – Stejně jako předchozí metoda, je toto používáno pro situace, kde uživatel zadává do aplikace nějaká data. Například, pokud má systém akceptovat celá čísla $\langle 0;100 \rangle$, v rámci testů zkusíme použít hraniční hodnoty minimální hodnoty, tedy -1, 0, 1 a maximální hodnoty, tedy 99, 100, 101.

Testování dle rozhodovací tabulky – V rámci této metody test analytik připraví tabulku, kde jsou popsány výstupy reagující na různé vstupy. Testování poté probíhá na základě této tabulky.

Testování přechodů stavů – Pro tuto metodu je klíčová analýza toho, jaké stavy aplikace obsahuje a kdy se tyto stavy mění. Testování probíhá na základě změn těchto stavů.

Testování případů užití – Testy jsou připraveny dle Případů užití. V rámci jednoho Případu užití může vzniknout několik testovacích scénářů. (13, s. 55, 56, 57, 58)

4 Vlastní práce

4.1 Úvod

V praktické části práce je popsán postup tvorby návrhu modelu požadavků, pro podnikovou aplikaci. Je zde podrobně zaznamenán postup získání uživatelských požadavků skrze metodu „požadavkový meeting“ a „pozorování uživatelů při práci“ a jejich zaznamenání skrze případy užití. Dále je zde popsán postup pro vytvoření schematického návrhu uživatelského rozhraní a analýzy pro testování. Kompletní výstup těchto analýz je připojen v příloze.

4.1.1 Popis zadavatelské společnosti

Zadavatelská firma je středně velkou společností, sídlící v Praze. Zabývá se účetnictvím a finančním poradenstvím. Mezi její klienty patří největší české i mnohé zahraniční firmy. Na trhu působí již přes 20 let, nicméně v posledních třech letech proběhlo prudké navýšení počtu zaměstnanců i zakázek. Bylo tedy nezbytné navýšit i počet kancelářských prostor. Firma nyní disponuje osmi různými kancelářskými prostory, ve dvou budovách. Pro jejich efektivní využívání, zejména zasedacích místností, bylo nezbytné zajistit pro zaměstnance kvalitní aplikaci, usnadňující orientaci v prostorech a umožňující spravovat rezervace zasedacích místností. Podobná aplikace již dříve byla ve firmě využívána. Z důvodu její zastaralosti a nespokojenosti ze strany zaměstnanců, bylo u příležitosti výše popsané expanze, rozhodnuto o zadání zakázky pro vytvoření aplikace nové.

4.1.2 Požadavek firmy na vypracování a dodání analýzy

Zadavatelská firma potřebuje vytvořit co nejpřínosnější a uživatelsky nejpříjemnější aplikaci. Proto se rozhodla do procesu tvorby uživatelských požadavků zapojit i běžné zaměstnance firmy, kteří s aplikací mají v budoucnu pracovat. Dodavatel tedy v první fázi připravil pouze rámcové požadavky.

Součástí zadání pro dodavatele je provést studii uživatelských požadavků, do které budou zapojeni i běžní uživatelé aplikace. Firma věří, že pokud běžní zaměstnanci budou sami vyjádřit svůj názor na aplikaci, bude dosaženo lepších výsledků.

Firma dále uvedla požadavek na předložení výsledných případů užití a testovacích případů uživatelského akceptačního testování, ze kterých bude vývoj a testování aplikace čerpat.

Samotný vývoj aplikace měl být zahájen až po schválení těchto analýz.

4.2 Sběr uživatelských požadavků

Zadavatelská firma dodala stručný, rámcový popis toho, co od aplikace očekává.

Stanovila tyto body:

- Aplikace bude zobrazovat plány kanceláří.
- V plánech kanceláří bude možné vyhledávat.
- Aplikace bude obsahovat systém pro rezervace zasedacích místností.

Jednalo se tedy pouze o obecné myšlenky, takovýto popis nebylo možné programátorům předložit. Myšlenky lze interpretovat velmi různě. Není například jasné, dle čeho se bude v plánech vyhledávat.

Jak je popsáno výše, zadavatelská firma záměrně poskytla tyto požadavky pouze obecné. Studium přesných uživatelských požadavků bylo úkolem pro dodavatele, kdy do procesu získávání uživatelských požadavků musí být zapojeni i uživatelé, kteří budou aplikaci používat. Uživateli se rozuměli zaměstnanci firmy, od manažerů přes řadové zaměstnance i občasné uživatele aplikace.

Bylo nutné připravit analýzu uživatelských požadavků, se zapojením zaměstnanců zadavatelské firmy.

4.2.1 Voba metod získání uživatelských požadavků

Určení způsobu získání uživatelských požadavků – tedy výběr metod a jejich vedení, bylo úkolem autora této práce, který zastával pozici analytika.

Nejprve bylo třeba zjistit, co uživatelé vlastně od aplikace potřebují, respektive – jakých cílů skrze aplikaci potřebují dosáhnout. Odborná literatura popisuje více metod, jak toto zjistit. Doporučuje se nezaměřit se pouze na jednu metodu, jelikož použití více metod se navzájem doplňuje a může tak být dosaženo kvalitnějších výsledků.

V této práci byly použity metody dvě:

1. Požadavkový meeting
2. Pozorování uživatelů při práci.

První sloužila jako primární zdroj informací, druhá jako sekundární.

Způsob naplánování a vedení metod byl autorem proveden na základě studia odborné literatury odkazované v teoretické části práce a na osobních zkušenostech.

4.2.2 Požadavkový meeting

Požadavkový meeting je nejčastější metodou sběru požadavků. Může být prováděn individuálně či v rámci skupiny. Pro tuto práci byl zvolen meeting v rámci skupiny.

1. Účastníci meetingu

Za stranu dodavatele se meetingu zúčastnili tři lidé. Autor této práce zastával funkci analytika a zároveň moderátora – tedy vedoucího schůze. Zápis ze schůze prováděl zapisovatel. Účastnil se i zástupce programátorů, kteří měli aplikaci po schválení analýzy programovat.

Analytik požádal zákazníka, aby se tohoto meetingu zúčastnili manažeři, dále zástupci jejich interního IT oddělení a alespoň pět zástupců budoucích řadových uživatelů této aplikace. Poslední jmenovaní byli zaměstnanci, kteří byli vybráni zadavatelem z řad dobrovolníků.

Úmyslně nebyla skupina velká. Účast zástupců uživatelů aplikace byla velmi výhodná, jelikož dodavatel skrze komunikaci s nimi má příležitost daleko přesněji zjistit jejich potřeby a očekávání. Celkem se takto schůze účastnilo deset lidí. To odpovídá doporučení, aby skupina nebyla příliš velká. Pro moderátora je tak jednodušší zajistit, aby nebyli někteří účastníci pasivní a opravdu se všichni k tématům vyjadřovali.

2. Způsob průběhu schůze

Jak již bylo řečeno, schůzi vedl analytik, který je zároveň autorem této práce. Začátkem diskuze u každého tématu bylo pokládání otevřených otázek.

Úkolem analytika bylo zajistit, aby se diskuze vždy držela daného tématu a byl zachováván časový limit – ten je popsán v dalším odstavci. V rámci zachování aktivity všech účastníků je důležité, aby schůzka účastníky alespoň trochu bavila. Pokud se

účastníci budou nudit, hrozí, že nebudou dostatečně aktivní a nebudou přinášet nové podněty.

3. Program schůze

Program schůze připravil analytik, ještě před samotnou schůzkou. Zúčastněným osobám za stranu zákazníka byl program zaslán předem, aby mohla proběhnout příprava. Zúčastněné osoby si některé věci předem promysleli – především uživatelské cíle a nedostatky původní aplikace.

Program by se měl dodržovat. Pokud se řeč dostala k tématu jiné části programu, analytik zareagoval slovy „k tomuto se dostaneme v další části programu“ a řeč vrátil do aktuálního tématu.

- 1. Představení, jak bude schůzka probíhat (5 minut)
- 2. Uživatelské role (20 minut)
- 4. Uživatelské cíle (2 hodiny)
- 3. Nedostatky původní aplikace (1 hodina)
- 5. Vzorové osoby (1 hodina)

U každého bodu byl stanoven časový limit, který bylo třeba dodržovat. Během této práce se limity podařilo dodržet. Pokud by se nepovedlo během limitu téma dořešit, je vhodné naplánovat další schůzku, kde se diskuse na dané téma dokončí.

4. Uživatelské role

Během této části bylo definováno, jaké uživatelské role budou existovat v rámci této aplikace.

Aplikace rozlišuje dvě uživatelské role - Uživatel a Administrátor. Administrátor má všechny možnosti jako Uživatel a disponuje ještě dalšími třemi, ke kterým Uživatel přístup nemá.

5. Uživatelské cíle

V případě, kdy se analytik zeptá na popis aplikace, dostane většinou velmi chaotický, případně i protiřečící si popis. Během vedení schůze tedy byla základní otázka - čeho uživatelé potřebují skrze aplikaci dosáhnout.

V tomto okamžiku byl využit rámcový popis požadované aplikace, který již analytik měl k dispozici. Diskuzi tedy zahájoval otázkami typu: „Jak přesně si toto představujete?“, či „Jak toto bude přesně probíhat?“.

Všechny odpovědi byly zapsány a poté postupně probrány podrobněji. Odborná literatura doporučuje, aby se moderátor snažil rozmluvit zástupce ze strany zákazníka obecnými otázkami, například „Jak se sem toto dostalo?“ či „Co se stane dál?“. Díky tomu bylo získáno mnoho cenných informací, ze kterých vznikly další uživatelské požadavky.

Je také nutné, se zabývat prioritou požadavků. Zde měli hlavní slovo pro změnu manažeri firmy a zástupce IT oddělení. Analytik byl připraven na velmi častou situaci, kdy účastníci meetingu považují téměř všechny požadavky za maximálně důležité. Proto účastníkům vysvětlil, že priorita je klíčová při řešení případných problémů, které na projektech často nastávají. Pokud určí priority požadavků sám dodavatel, může vzniknout rozpor se stranou zákazníka, proto bylo třeba toto řešit již nyní.

Na základě doporučení odborné literatury byla zvolena metodika rozdělení priorit dle Wiegese (4, s. 228). Toto rozdělení se řídí dle dvou úrovní, důležitostí a naléhavostí.

Obrázek 7: Rozdělení priorit

	Důležité	Nedůležité
Spěchají	vysoká priorita	neimplementovat!
Nespěchají	střední priorita	nizká priorita

Zdroj: (4, s. 228)

Výsledkem diskuze o prioritách v rámci této práce bylo například definování, že „UP002: Zobrazení plánu kanceláří“ musí mít nejvyšší prioritu. Jedná se o základní funkci, bez které aplikace v podstatě využití nemá. Naopak „UP003: Vyhledávání v plánech kanceláří“ má prioritu Střední, jelikož se jedná o důležitou funkci, ale aplikace může být částečně využívána i bez ní – například pouze k prohlížení plánů, bez vyhledávání.

6. Vzorové osoby

Důležitým bodem, řešeným na meetingu, bylo definování čtyř ukázkových uživatelů. Toto později posloužilo při návrhu uživatelského rozhraní– více v příslušné kapitole.

Jednalo se o interní aplikaci, která nebyla určena nikomu jinému, než zaměstnancům firmy – tedy různým typům zaměstnanců. Cílem bylo vytvořit model osoby, nejednalo se o informace o skutečném, konkrétním člověku.

Možností bylo zvolit buďto cílovou skupinu, či pouze samostatné osoby. Koncoví uživatelé aplikace tvoří relativně homogenní skupinu. Proto na základě doporučení odborné literatury, byla zvolena ukázková osoba, nikoli cílová skupina. Při sestavování skupiny byl kladen důraz na to, aby osoby definovaly co nejrozumnější budoucí uživatele. Analytik tedy otázkami diskuzi navedl směrem, jací různí lidé s různými technickými schopnostmi ve firmě jsou.

Cílem bylo vytvořit dvě primární vzorové osoby a dvě sekundární. Analytik vysvětlil, že primární osobou se myslí taková, která bude aplikaci využívat nejčastěji. Naopak sekundární osoba je taková, která nebude aplikaci využívat tak často, přesto je ale třeba, aby v případě potřeby aplikaci mohla použít. Příkladem může být osoba asistentky, která plánuje meetingy, tudíž služby aplikace využije častěji než například údržbář, který ji použije občas - například pokud bude mít nahlášenu prasklou žárovku v některé ze zasedaček.

Byly tedy definovány celkem čtyři fiktivní osoby, které reprezentují různé typické budoucí uživatele této aplikace. Dvě z nich jsou primární a dvě sekundární. Tyto osoby byly definovány těmito informacemi:

Jméno

Pohlaví

Věk

Pracovní pozice

Stručný životopis

Životní cíle

Zkušenosti s prací na PC

Typický den

Koníčky

Konkrétní výsledky jsou připojeny v příloze této práce.

7. Nedostatky původní aplikace

Jednalo se o vytvoření nové aplikace, která má nahradit starou aplikaci, se kterou byla společnost nespokojena. Proto byl vysloven dotaz, které věci lidem vadily nejvíce. Zde byla naplno využita přítomnost zástupců řadových uživatelů aplikace.

První problém byla absence možnosti obnovení hesla v případě, kdy jej uživatel zapomněl. V takovém případě byla nutná asistence správců systému, kteří heslo manuálně obnovili.

Druhý nedostatek nastával při situaci, kdy uživatel potřeboval zobrazit pouze plán určitého podlaží. Tato situace nastala například, pokud se uživatel potřeboval zorientovat, kterým směrem je kuchyňka. V původní aplikaci bylo v takovém případě nutné provést vyhledání určité místnosti. Poté byl zobrazen plán celého patra, se zvýrazněním vyhledávané zasedačky. Zadavatel si přeje, aby bylo možné provést přímé zobrazení plánu podlaží..

Třetím hlavním nedostatkem byla absence grafického přehledu existujících rezervací. Byl zde pouze textový přehled, který nebyl příjemný pro uživatelské použití. Zadavatel si proto přeje, aby nový systém již tento přehled obsahoval.

Tyto informace byly zaznamenány zapisovatelem a poslouží při přípravě případů užití v následující kapitole.

8. Zápis schůze

Po celou dobu konání meetingu byly všechny zjištěné informace dokumentovány zapisovatelem.

Na to nebyl třeba žádný speciální software, postačil program Poznámkový blok.

Po ukončení schůze zapisovatel své poznámky utřídil do přehledných informací a vytvořil tak zápis ze schůze. Ten byl poté odeslán všem účastníkům schůze, kteří tak ještě dostali možnost se k výsledkům schůze zpětně vyjádřit.

4.2.3 Pozorování uživatelů při práci

Jako sekundární zdroj sbírání požadavků, byla zvolena metoda pozorování uživatelů při práci. Konkrétně se jednalo o práci s původním systémem, který měl být nahrazen novou aplikací. Jednalo se spíše o doplnění k požadavkovému meetingu, proto byla tato akce naplánována až po něm. Mělo by tak být především lépe zmapováno, s čím mají uživatelé problémy a co je pro ně nepohodlné.

Původní aplikace neměla jako součást logovací systém, musela tedy být zvolena přímá metoda, kdy analytik pozoruje uživatele při práci.

Jelikož se nejedná o systém, který by uživatel používal delší dobu v kuse, ale používá ho pouze občas v případě potřeby, nebylo efektivní uživatele sledovat ve zcela běžném provozu.

Bylo tedy naplánováno setkání s několika náhodně vybranými, běžnými uživateli aplikace. Byl jim předložen seznam úkolů, které mají v aplikaci provést.

Úkoly:

1. Proved' přihlášení do aplikace.
2. Zobraz plán podlaží, na kterém bude zvýrazněna určitá zasedačka (byl přiložen název konkrétní zasedačky).
3. Vytvoř novou rezervaci (byl přiložen název konkrétní zasedačky a termínu).
4. V přehledu rezervací vidíš, že máš být za 5 minut přítomen na schůzi. Za použití plánů podlaží najdi, kde se tato zasedačka nachází.

Na základě pozorování uživatelů analytik vyhodnotil dva poznatky, které nebyly zaznamenány během požadavkového meetingu.

První problém nastával při úkolu 2, tedy vyhledávání určité místnosti, kdy bylo nutné zadat jako vyhledávací kritérium zároveň i název příslušného patra. Uživatel tak například znal název zasedačky kterou hledal, ale pokud neznal i název příslušného patra, bylo vyhledání náročné.

Druhý problém se projevil v úkolu 4, kdy uživatelé měli zobrazenou informaci o názvu požadované místnosti, nicméně pro její zobrazení v plánu podlaží ji museli manuálně vyhledat.

Po informování zadavatele o těchto nových zjištěních byly tyto informace přidány k materiálům, získaných z požadavkového meetingu, pro finální zpracování uživatelských požadavků.

4.2.4 Zpracování získaných informací

Další zpracování informací, získaných během požadavkového meetingu a pozorování uživatelů při práci, bylo opět úkolem pro příslušného analytika, tedy autora této práce.

Ze získaných informací byl nyní sestaven seznam požadavků. Bylo nutné být připraven na případnou zpětnou reakci zákazníka, k odeslanému zápisu ze schůze. V případě, kdyby zpětné změny byly výraznější, bylo by možné svolat nový meeting, kdy by byly změny probrány, podobným způsobem jako při prvním meetingu.

Způsob zápisu uživatelských požadavků:

1. Identifikátor požadavku

Bylo nutné pro každý požadavek uvést jednoznačný identifikátor. Ten může být využit při odkazech či žádostech o změnu či zrušení požadavku.

Pro tuto práci byl zvolen systém postupného číslování, tedy nejjednodušší, kdy první požadavek nese označení UP001, druhý UP002 atd.

2. Priorita

Priorita požadavků byla určena zákazníkem již při požadavkovém meetingu. Analytik v tomto případě nenašel důvod u některého požadavku prioritu měnit. Pokud by ale takový důvod u některého z požadavků našel, změnu by musel konzultovat s dodavatelem.

3. Popis

Popis požadavku byl zapisován v přirozeném jazyce. Nebyl třeba žádný složitý popis, výhodou takového zápisu je, že ho pochopí v podstatě každý, analytik, manažer, vývojář i uživatel.

4.2.5 Podrobný postup tvorby uživatelských požadavků

Na základě dohody s vedoucím této práce byly dva uživatelské požadavky rozebrány podrobněji. Na stejné požadavky je poté navázáno v následujících kapitolách při přípravě případu užití a testovacích případů. Autor práce pro tento rozbor zvolil zobrazení grafického schématu a možnosti jeho filtrování.

Dle požadavkového meetingu byla požadována funkce pro zobrazení grafického schématu rezervací zasedaček s možností filtrování skrze tři parametry. Možnost filtrování navazuje na možnost zobrazení schématu, nicméně se jedná o dva různé uživatelské požadavky.

Požadavek na grafické schéma:

První z nich popisuje samotné zobrazení grafického schématu.

Byla pro něj zvolena priorita Vysoká, jelikož se jedná o základní požadavek, na který byly navázány další funkce. Součástí tohoto požadavku byla i možnost posouvání zobrazení po dnech – toto patřilo k základnímu zobrazení schématu, proto bylo posouvání začleněno přímo do tohoto požadavku a nebylo vytvořeno jako samostatný požadavek.

Výsledný uživatelský požadavek:

UP004: Grafické schéma rezervací zasedaček

Priorita: Vysoká

Popis: Potřebuji mít možnost zobrazení grafického schématu znázorňujícího existující rezervace jedné či více zasedaček pro určitý den. Přehled je možné posouvat po dnech.

Požadavek na filtr v grafickém schématu:

Druhý uživatelský požadavek popisuje možnosti filtrování grafického přehledu – jednotlivé možnosti jsou zde konkrétně uvedeny.

Priorita byla zvolena Střední. Jednalo se o důležitý požadavek, proto nemohl mít nízkou prioritu. Nicméně grafický přehled byl použitelný i bez něj, proto nebyla zvolena priorita Vysoká.

Výsledný uživatelský požadavek:

UP005: Filtr zobrazení v grafickém schématu rezervací zasedaček

Priorita: Střední

Popis: Potřebuji, aby v rámci grafického schématu rezervací zasedaček byla možnost výběru zobrazení všech zasedaček v určité budově, v podlaží či pouze jedné zasedačky.

S grafickým schématem souvisí i další požadavky – vytvoření nové rezervace, zobrazení detailu atd... Jedná se ale o další, samostatné požadavky.

Kompletní seznam uživatelských požadavků je součástí přílohy této práce.

4.2.6 Výsledek kapitoly

Nyní bylo dosaženo získání uživatelských požadavků, které posloužily jako podklad pro zpracování případů užití, řešených v následující kapitole. Vytvoření modelu požadavků bylo dosaženo až skrze případy užití. Požadavky byly po zpracování uloženy do Specifikace požadavků, též nazývanou Funkcí specifikace.

4.3 Případy užití a scénáře užití

Jak již bylo popsáno v teoretických podkladech této práce, případ užití popisuje určitý cíl, kterého má uživatel skrze software dosáhnout. Scénář užití popisuje dosažení tohoto cíle podrobněji, skrze jednotlivé kroky.

4.3.1 Postup a způsob přípravy případů užití a scénářů užití

Případy užití vytvořil opět autor této práce, tedy analytik, který se již účastnil požadavkového meetingu a vytvořil i seznam uživatelských požadavků. Případy užití a scénáře užití vychází z uživatelských požadavků, které byly popsány v předcházející kapitole.

Během zpracování byl kladen důraz na to, aby případy užití pokud možno nikdy nepopisovaly grafické prvky. Grafickému návrhu je věnována jedna z následujících kapitol. Důvodem je, že grafický návrh se vytváří až po dokončení všech případů užití. Pokud by grafiku popisovaly již případy užití, pravděpodobně by nastal zmatek a redundance informací. Při přípravě případů užití, v rámci této práce, byly dle Pavlíčka (11, s. 66, 67,

68) rozlišovány výrazy „očekává“ a „požaduje“. Výraz „očekává“ je použit ve většině případech. Vyjadřuje uživatelské očekávání pro dosažení jeho cíle. Termín „požaduje“ se naopak používá v případě, kdy opravdu není jiná možnost a je třeba popsat konkrétní prvek uživatelského rozhraní. Taková situace může nastat například v případě, kdy klient výslovně takový prvek požaduje.

Použití těchto termínů vyplývá již z uživatelských požadavků, popsaných v předchozí kapitole a které slouží jako podklad přípravy případů užití. Při práci byl využit program Microsoft Excel, ve kterém byla vytvořena šablona s tabulkou, do které byl poté dle potřeby vkládán text. Poté byly tabulky přeprogramovány do programu Microsoft Word, který posloužil i pro převod do formátu .pdf.

4.3.2 Volba obsahu případů užití

Nikde není přesně definováno, co přesně má případ užití obsahovat a jak má být zapsán. Odborná literatura nabízí různá obecná doporučení. Určujících atributů může být více, může jich ale být i velmi málo. Forma zápisu tedy do určité míry závisí na analytikovi.

V této práci byla jako podklad využita doporučení dle Wiegerse (4, s. 136) a Fowlera (5, s. 104, 105).

Pro případy užití byly popsány tyto atributy:

1. Identifikátor - Každý případ užití musí obsahovat jedinečný identifikátor. V této práci byla zvolena metoda jednoduchého číslování. První případ užití je identifikován jako PU001, druhý PU002 a podobně.

2. Název - Název případu užití. Je třeba aby byl krátký a výstižný.

3. Uživatelské požadavky - Odkaz skrze identifikátor na uživatelský požadavek, ze kterého byl tento případ užití vytvořen.

4. Aktér - Aktér je uživatelská role, kterou v tomto případě užití disponuje uživatel. Tyto role byly definovány během požadavkového meetingu, popisovaném v předchozí kapitole.

5. Priorita - Prioritu je třeba určit pro případ, pokud by bylo kvůli nedostatku času či prostředků nutné některé požadavky odstranit, či odložit na později. Hlavní hlas u určení priorit měl zadavatel. K prioritám se vyjadřoval i zástupce vývojářů, jelikož mohlo vyjít

najevo, že některé funkce jsou pro vývoj náročnější, a tedy i nákladnější, než se původně zdálo, což může názor zákazníka na prioritu požadavku ovlivnit.

Samotná priorita případu užití vychází z uživatelského požadavku, na základě kterého byla vytvořena. Rozdělení se tedy opět řídí dle důležitosti a naléhavosti, a má čtyři stupně, tak jak je popsáno v kapitole o uživatelských požadavcích.

6. Popis - Stručný, věcný popis v přirozeném jazyce.

7. Vstupní podmínky - Jedná se o jednu nebo více podmínek, které musí být splněny, aby tento případ užití mohl být zahájen.

8. Výstupní podmínky - Výstupní podmínky určují, zda byly cíle případu užití splněny.

4.3.3 Podrobný postup tvorby případů užití

Nyní se opět vraťme ke dvěma uživatelským požadavkům, popsaných v minulé kapitole. Aktuálním cílem je vytvořit z nich případy užití.

Případ užití vytvoření z UP004

Prvním podrobněji rozebraným případem užití je PU004: Grafické schéma rezervací zasedaček. Vychází z UP004.

Priorita: byla nastavena jako Vysoká. Bylo tak určeno na schůzi. Jedná se o základní funkci aplikace. Bez tohoto případu užití nelze grafický přehled použít.

Vstupní podmínky: základní podmínkou je typ uživatelské role. Tento případ užití je přístupný pro obě role, tedy Uživatel i Administrátor. Dalším požadavkem pro spuštění případu užití, je aby byl uživatel přihlášen do aplikace, jelikož bez toho případ užití nelze zahájit.

Výstupní podmínky: tento případ užití je považován za dokončený v okamžiku, kdy je úspěšně dokončen poslední bod – tedy aplikace zobrazuje grafické schéma a uživatel změnil datum zobrazení.

Případ užití: příliš se neliší od uživatelského požadavku. Je přeformulován do podoby uživatelského očekávání - toto odpovídá tomu, že případy užití jsou psány z pohledu dosahování uživatelských cílů.

Scénář užití: uživatel volí možnost pro zobrazení grafického schématu – tímto zahajuje scénář užití. Reakce systému je, že zobrazí grafické schéma. Zde bylo zdůrazněno, že zobrazení se týká aktuálního dne.

Druhý krok scénáře užití popisuje změnu zobrazeného data. Uživatel zvolí příslušnou možnost, systém provede změnu. Úmyslně zde nejsou uvedeny detaily – názvy tlačítek či způsob výběru datumu. Takto konkrétní popis bude součástí budoucích analýz, ve scénáři užití je nežádoucí.

Výsledný případ užití:

PU004: Grafické schéma rezervací zasedaček

Priorita: Vysoká

Uživatelský požadavek: UP004

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Uživateli je zobrazeno grafické schéma rezervací zasedaček pro datum dle uživatelova výběru.

Případ užití:

Uživatel očekává možnost zobrazení grafického schéma znázorňujícího existující rezervace zasedaček pro určitý den. Přehled je možné posouvat po dnech.

Scénář užití:

1. Uživatel zvolí možnost pro zobrazení grafického schématu.
2. Systém provede zobrazení grafického schématu pro aktuální den.
3. Uživatel vybere možnost pro změnu data.
4. Systém zobrazí grafické schéma pro datum, dle uživatelova výběru.

Případ užití vytvoření z UP005

Druhým podrobněji popsáním případem užití je PU005.

Vychází z UP005, který úzce souvisí s UP004, jak bylo popsáno v minulé kapitole.

Priorita: byla určena jako střední. Bylo přihlédnuto k požadavku z meetingu a k faktu, že použití grafického přehledu je možné i bez filtrování. To je ale zároveň důležitou funkcí, zlepšující použití, proto nemůže být priorita ani nízká.

Vstupní podmínky: opět byla určena podmínka pro vlastnictví role Uživatel nebo Administrátor, jelikož případ užití je dostupný pro obě role. Zároveň je pro spuštění případu užití třeba přihlášení do aplikace a zobrazení přehledu rezervací.

Výstupní podmínky: případ užití je ukončen v okamžiku, kdy proběhlo zobrazení schématu, na základě uživatelova filtrování.

Případ užití: podobně jako v minulém případě, byl případ užití přepsán do podoby uživatelova očekávání.

Scénář užití: uživatel zahájí případ užití výběrem možnosti pro filtrování zobrazených místností. Systém reaguje tak, že uživateli nabídne možnost zadat filtrovací kritéria – na základě toho, jaký způsob filtrování uživatel zvolil. Opět je toto záměrně popisováno velmi obecně.

Ve druhém kroku uživatel zadává vyhledávací kritéria a potvrdí vyhledání. Aplikace zobrazuje výsledek dle uživatelova vyhledávání a tímto scénář končí. Úmyslně je zde zdůrazněno, že zobrazení rezervací se týká aktuálního dne. Úkol změny datumu je již popsán v minulém případě užití, proto zde není vhodné ho opakovat, z důvodu redundance informací.

Výsledný případ užití:

PU005: Filtr zobrazení v grafickém schématu rezervací zasedaček

Priorita: Střední

Uživatelský požadavek: UP005

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace. Uživatel má zobrazen přehled rezervací.

Výstupní podmínky: Systém provedl filtrování zobrazených zasedaček, dle všech možností.

Případ užití:

Uživatel v grafickém schématu očekává možnost filtrování zobrazených zasedaček.

- a. Očekává možnost zobrazení všech zasedaček v určité budově
- b. Očekává možnost zobrazení pouze jedné zasedačky
- c. Očekává možnost zobrazení všech zasedaček v určitém podlaží.

Scénář užití:

1. Uživatel zvolí, zda chce v grafickém přehledu zobrazit zasedačky v určité budově, v podlaží či pouze jedné zasedačky.
2. Systém, dle uživatelského výběru, zobrazí možnosti zadání vyhledávacích kritérií.
3. Uživatel zadá vyhledávací kritéria a vyhledání potvrdí.
4. Aplikace zobrazí grafický přehled obsazenosti zasedáček či zasedačky, dle uživatelského výběru, defaultně pro aktuální den.

4.3.4 Nesprávný způsob zápisu případů užití

Během zpracování případů užití došlo k problému, kdy případy užití byly napsány způsobem, který programátoři označili jako obtížně použitelný.

Při tvorbě případů užití je důležité zachovávat popis obecným způsobem, z pohledu dosahování cílů skrze persony. V případě, kdy například scénář případu užití popisuje, že se modální okno zavírá křížkem v pravo nahoře, může nastat zmatek, jelikož se jedná o obecný případ užití. Některé z případů užití se mohou, a pravděpodobně i budou měnit, případně na různých případech užití může pracovat i více různých lidí.

Tento způsob zápisu je zastaralý, nicméně na některých projektech se stále využívá.

Příklad části nevhodně popsaného scénáře užití:

Hlavní scénář:

1. Aktér: Systém

Systém zobrazuje okno „VR1“, v defaultním stavu.

Grafický přehled rezervací je nyní prázdný, protože nebyl proveden žádný výběr.

Tlačítko „Změnit datum“ je nyní neaktivní.

2. Aktér: Uživatel

Uživatel klikne na tlačítko "Vyhledat zasedačky v budově".

3. Aktér: Systém

Systém zobrazí modální okno „VR2“.

V modálním okně je zobrazen seznam všech pater, v horní části je filtr s jedním parametrem - Budova.

Po kliknutí do pole filtru lze psát název budovy, filtrování probíhá ihned po vepsání textu. Pokud je v textovém poli filtru nějaký záznam, v jeho pravé části se zobrazí křížek,

po kliknutí na něj se vyplněný text v textovém poli vymaže, nalezené výsledky jsou ihned aktualizovány.

4.3.5 Akceptování analýzy zákazníkem

Spolu s dokumentem obsahujícím testovací případy uživatelského akceptačního testování (jehož tvorba je popsána v jedné z následujících kapitol), byly případy užití odeslány ke schválení zadavatelské firmě.

Ta po prostudování tyto analýzy akceptovala a bylo tedy možné zahájit další vývoj aplikace.

4.4 Diagram případů užití

V rámci přípravy modelu požadavků, autor rozhodl o vytvoření diagramu případů užití. Jeho přínosem bylo především zpřehlednění toho, jaké uživatelské role mají přístup ke kterým případům užití. To využijí všichni, kdo budou dokumentaci studovat. Jak již název napovídá, vychází z již zdokumentovaných případů užití.

O připravení diagramu případů užití bylo rozhodnuto i z důvodu dostatečného časového prostoru. Diagram případů užití není věcí, bez které by se dokumentace nemohla obejít. Odborná literatura nedoporučuje do diagramu případů užití vkládat mnoho energie v případě, pokud by hrozilo, že by jinde mohla chybět.

4.4.1 Volba software

V této práci byl jazyk UML použit pouze jako jednoduchý náčrt (sketch). Tento proces modelování není příliš technicky náročný, existuje tedy velké množství placených i neplacených aplikací, skrze které lze UML diagramy vytvářet. Pro tuto práci byla zvolena aplikace <https://www.draw.io/>. Její výhodou je možnost využití zdarma a zároveň to, že obsahuje velké množství možností pro tvorbu UML diagramů. Výkresy lze ukládat na lokální disk ve formátu „drawio“, což umožňuje jejich opětovné otevření a další úpravy. Výkresy je též možno exportovat jako hotový obrázek ve formátu „png“. Aplikace má jednoduché, intuitivní uživatelské rozhraní, skrze které je možno vkládat do výkresu předdefinované objekty a následně je editovat, čímž vzniká výkres.

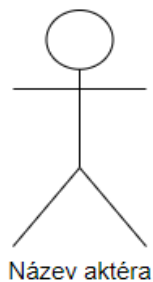
4.4.2 Tvorba diagramu

Jak již bylo popsáno v teoretické kapitole, UML se dělí na tři části – Věci, či předměty (Things), Vztahy (Relationships) a Diagramy (Diagrams).

V této práci byly použity tyto položky:

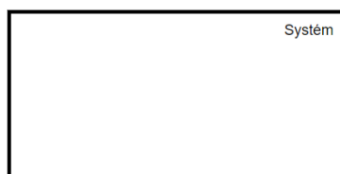
1. Předměty:

Obrázek 8: Symbol aktéra



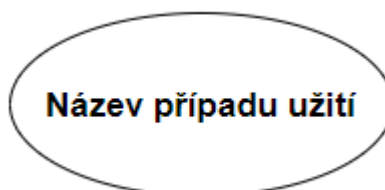
Takto je značen aktér. V tomto případě jsou evidováni dva aktéři – Uživatel a Administrátor. Název aktéra se označuje textem, ve spodní části.

Obrázek 9: Symbol systému



Obdélníkem se značí hranice systému. V rámci jednoho diagramu případů užití může být zobrazeno více systémů, v tomto případě je evidován pouze jeden.

Obrázek 10: Symbol případu užití



Případ užití se vkládá jako ovál. Doprostřed tohoto oválu je vložen název příslušného případu užití.

2. Vztahy

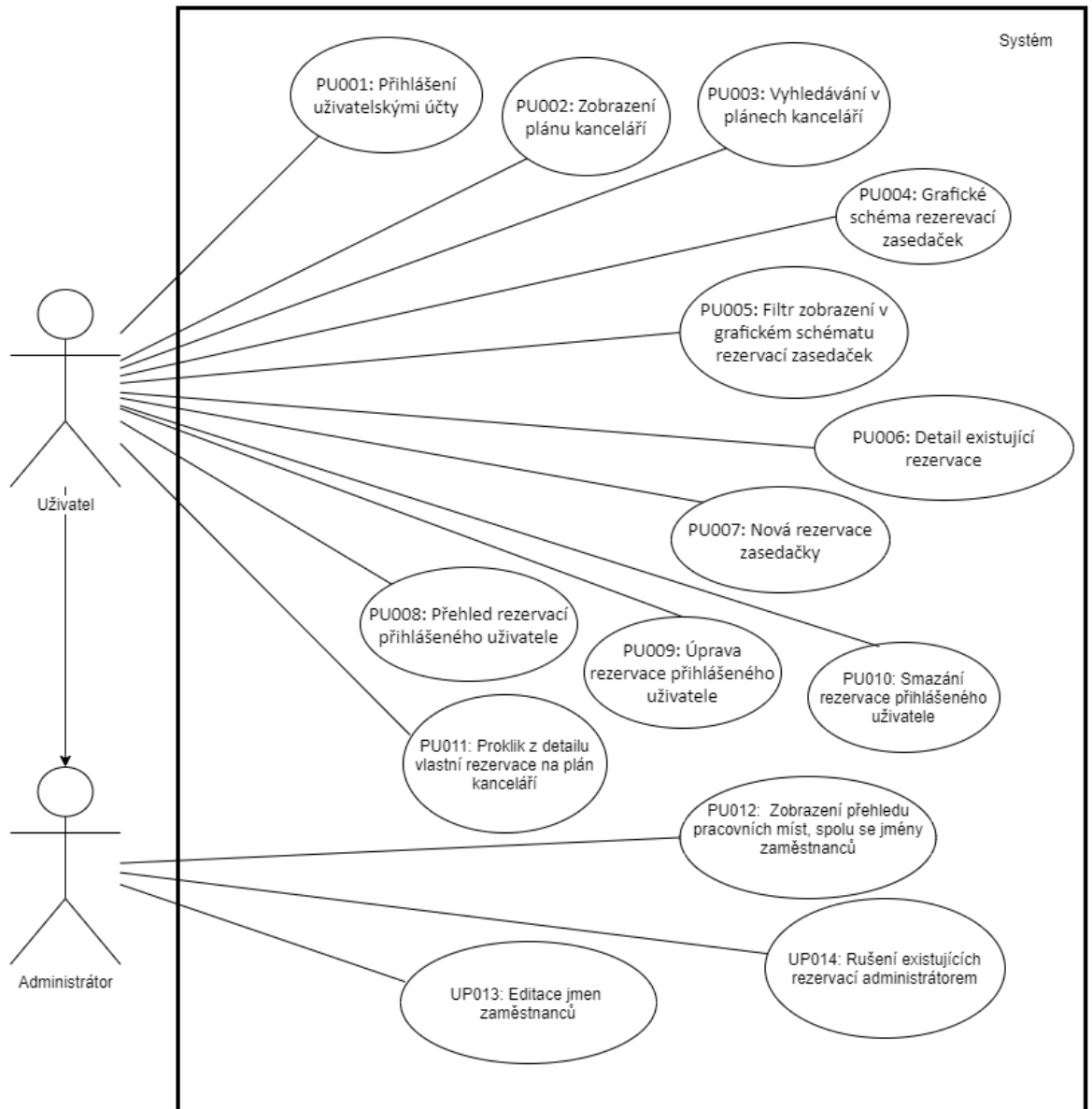
Čáry směrem od případů užití k aktérům označují, který aktér má přístup k příslušným případům užití.

Šipka směřující směrem od aktéra Administrátor k aktérovi Uživatel definuje, že Administrátor dědí všechny přístupy ke všem případům užití, kterými disponuje Uživatel.

Další možností je použití vazeb typu <<include>> a <<extend>>, které ale v této práci využity nebyly. Vazba <<include>> propojuje jeden případ užití s dalším. Když je spuštěn první případ užití, je vždy spuštěn i druhý. Typ vazby <<extend>> opět propojuje dva případy užití. Druhý, takto připojený případ užití se spouští při splnění určité podmínky. Dle odborné literatury je však tento typ vazby využíván velmi zřídka.

3. Diagram Případů užití

Obrázek 11: Diagram případů užití



Z tohoto diagramu je patrné, že je popsáno celkem čtrnáct případů užití. Všechny případy užití jsou prováděny v jednom systému. Existují dvě uživatelské role – Uživatel a Administrátor. Uživatel má přístup celkem k jedenácti případům užití. Administrátor má přístup ke všem případům užití jako Uživatel a navíc disponuje přístupem ještě ke třem dalším.

4.5 Model uživatelského rozhraní

Uživatelské rozhraní je částí aplikace, kterou uživatel používá pro komunikaci s příslušným systémem.

Model uživatelského rozhraní se skládá ze dvou částí - logického modelu (Wire Frame) a grafického designu. V této kapitole je popsáno vytvoření logického modelu příslušné aplikace, který poté posloužil jako podklad pro tvorbu grafického designu.

4.5.1 Postup dle obecné metody návrhu

Nejprve proběhlo studium informací, získaných v předchozích kapitolách – především případů užití. Poté začala tvorba samotného návrhu uživatelského rozhraní, skrze WireFrame.

Další postup je rozdělen do několika fází:

1. Strategie

Cílem této části bylo pochopit cíle a uvažování koncových uživatelů. Pouze tak lze navrhnout uživatelské rozhraní způsobem, aby se uživatelům s aplikací pracovalo pohodlně a efektivně. Toto je velice důležitá část. Dle literatury ale bohužel často opomíjená či odbytá.

Již během požadavkového meetingu, popisovaného v kapitole Sběr uživatelských požadavků, byly definovány vzorové osoby. Jejich příprava byla naplno využita nyní. Analytik si při návrhu uživatelského rozhraní přečetl popis vzorové osoby, a pokusil se přemýšlet tak, jak by přemýšlela tato postava. „Nasazení klobouku“ vzorové postavy je snažší, čím detailněji je vzorová postava popsána. Analytik přemýšlel – věděla by např. Veronika, kde má zde kliknout?

Během návrhu bylo zohledněno, že se jednalo o novou verzi již používané aplikace, na jejíž používání byli uživatelé zvyklí. Pokud by v nové aplikaci bylo uživatelské rozhraní výrazně jiné, uživatelé by mohli být změnou zaskočeni a nepracovalo by se jim, minimálně ze začátku, pohodlně. Proto v návrhu nebyly oproti původní aplikaci vypracovány výrazné změny logiky, všude tam kde to bylo možné.

Zde je příklad jedné ze vzorových osob, které byly definovány na požadavkovém meetingu.

Jméno: Jana Malá

Pohlaví: Žena

Věk: 41

Pracovní pozice: Účetní

Stručný životopis: Po absolvování gymnázia zahájila studium oboru Účetnictví na VŠE. Během studií si našla práci jako účetní a vdala se. Po dokončení studií odešla na sedmiletou mateřskou dovolenou. Od té doby pracuje na pozici účetní, stále ve stejné firmě, ve které pracovala během studií.

Životní cíle: Náplní jejího života je především péče o její rodinu. Je spokojená, pokud jsou spokojeni všichni členové její rodiny.

Zkušenosti s prací na PC: Počítače používá pouze pro svou práci. Ve volném čase má jiné zájmy.

Typický den: Vstává brzy ráno, vyvenčí psy a chystá rodině snídani. Do práce poté dojíždí MHD. Po návratu z práce se věnuje dětem a domácím mazlíčkům. Večer čte knihy nebo sleduje filmy.

Koníčky: Četba knih

Analytik si popis těchto vzorových person nastudoval a pokusil se vžít do jejich zkušeností a chování. U Jany Malé vidí, že nemá příliš velké znalosti práce s PC, jednoduše proto, že ji příliš nezajímá. Je spíše praktická, potřebuje mít vše přehledné a dostupné na jednom místě.

2. Studium požadavkového modelu

V této části posloužil požadavkový model, vypracovaný v kapitole Sběr uživatelských požadavků. Analytik zde viděl požadavky na systém, tak jak jej zadavatel definoval. Získal tak obecný přehled o cílech, kterých bude uživatel skrze aplikaci dosahovat. Vše bylo stále pouze v obecné rovině.

3. Studium případů užití

Nyní byly využity informace, vytvořené v kapitole Případy užití a scénáře užití. Skrze ně analytik získal přehled, jakými kroky uživatelé budou dosahovat svých cílů.

4.5.2 WireFrame

Dalším bodem byla příprava WireFrame. Jedná se o schematický náčrtek, tvořený pouze čarami (wire) a barvami v odstínech šedi (nebo jiné barvy). Cílem WireFrame je zachycení rozmístění prvků uživatelského rozhraní.

Tvorba WireFrame čerpala z poznatků, získaných v předchozích třech krocích. Spojily se tak informace o myšlení uživatelů, o tom jakých cílů je třeba dosáhnout a jakým způsobem těchto cílů bude dosaženo.

Samotná tvorba WireFrame má tři základní části. Všechny tři části spolu úzce souvisí. Pořadí jejich vzniku není přesně dáno, můžou vznikat současně, dle potřeb autora.

1. Interface design

V rámci Interface design byly určeny konkrétní prvky uživatelského rozhraní. Výběr těchto prvků je závislý na potřebách konkrétního použití aplikace. Ze základních prvků byly v aplikaci použity prvky: Action button a Text field.

Další elementy uživatelského rozhraní jsou dohledatelné např. zde: (14, s. 116, 117, 118).

Obrázek 12: Prvek Text field

Text fields allow users to—wait for it—enter text.



Zdroj: (14, s. 116)

Obrázek 13: Prvek Action button

Action buttons can do lots of different things. Typically, they tell the system to take all the other information the user has provided via other interface elements and do something—take action—with it.



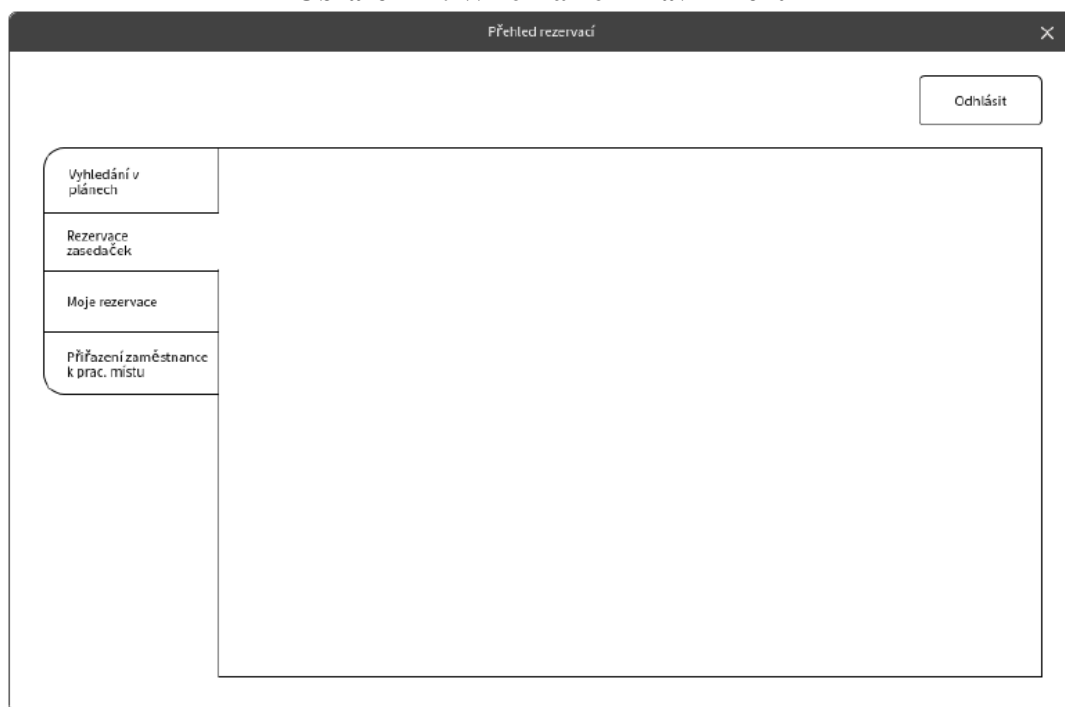
Zdroj: (14, s. 117)

2. Design navigace

Skrze Design navigace bylo třeba určit a popsat, jak se uživatel bude mezi jednotlivými okny aplikace pohybovat.

V případě vypracovávané aplikace byla navigace plánována tak, že po přihlášení bude mít uživatel vždy přístup na Hlavní menu, odkud se může přesunout na zahájení jím požadovaného případu užití. Design lépe popisuje následující náčrt:

Obrázek 14: WireFrame - Hlavní menu



3. Informační design

Informační design určil, jakým způsobem jsou uživateli požadované informace zobrazovány. Cílem bylo, aby informace které aplikace uživateli zobrazuje, byly prezentovány srozumitelně.

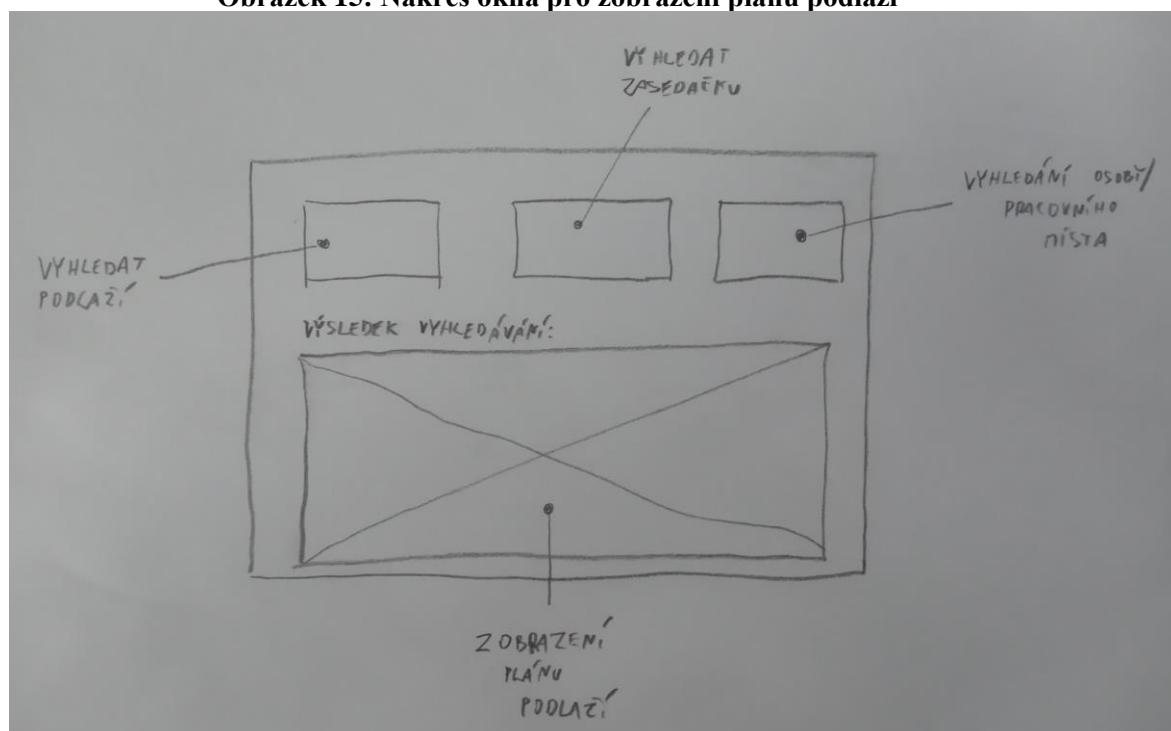
Pokud například chceme zobrazit detail uživateli rezervace, je důležité, aby zobrazené informace byly uspořádané v určitém pořadí. Vzhledem k tomu, že uživatelé informace zobrazené ve sloupci čtou odzhora, budou důležitější informace zobrazeny nahoře. Název zasedačky tedy bude výše, než informace Datum. Následují údaje Od a Do.

4.5.3 Kresba WireFrame

1. Papír a tužka

V první fázi designér tvořil WireFrame pouze s papírem a tužkou. Výhodou tohoto postupu je, že lze změny provádět rychle a efektivně. Někteří lidé se dokáží takovýmto způsobem i lépe soustředit.

Obrázek 15: Nákres okna pro zobrazení plánu podlaží



2. Tvorba v PC

Po vypracování papírových návrhů byly WireFrame vytvořeny i v počítači. Pro jejich tvorbu je dostupné velké množství editorů. Jejich výběr byl podmíněn především jejich možnostmi, hodnoceními v internetových diskuzích a v neposlední řadě licencemi jejich použití.

Při této práci byly využity dvě grafické aplikace:

WireFramePro: Dle internetových recenzí velmi dobře hodnocená aplikace.

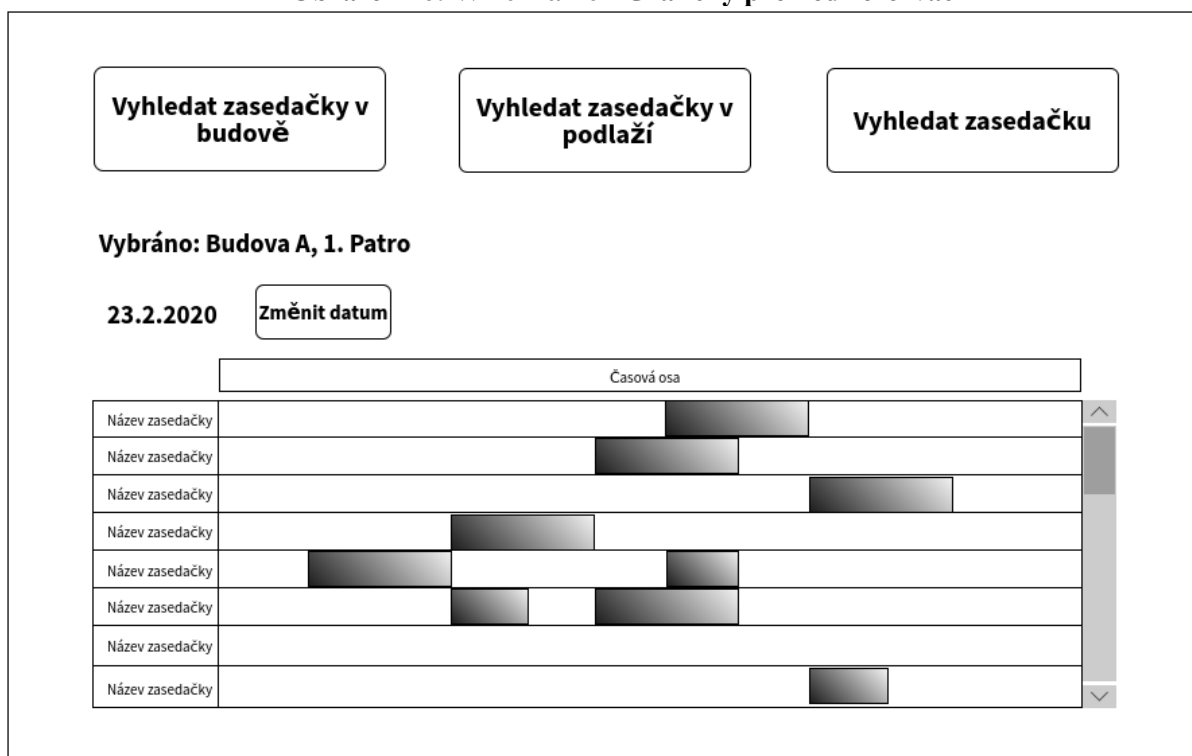
Zdarma nabízí použití její části, která pro tvorbu návrhu WireFrame dostačuje. Obsahuje všechny základní prvky, pro kresbu návrhu Uživatelského rozhraní.

Gimp: Často používaný, volně šiřitelný grafický nástroj. Při práci byl používán jako doplněk aplikace, popsané výše.

4.5.4 Ukázka hotového návrhu wireframe

Na následujícím obrázku je zobrazen hotový návrh wireframe grafického přehledu rezervací, vytvořený dle výše popsaného postupu na základě uživatelských požadavků UP004 a UP005, spolu s případy užití PU004 a PU005.

Obrázek 16: WireFrame - Grafický přehled rezervací



Další vytvořené grafy WireFrame jsou připojeny v příloze této práce.

4.6 Uživatelské akceptační testování

Existuje více druhů testování. Například integrační testy, kde se testuje komunikace mezi jednotlivými službami či moduly, penetrační testování, kde se testuje bezpečnost aplikace či performance testy, které testují výkonnostní zátěž hardwaru, na kterém aplikace běží.

V této práci jsou vypracovány testovací případy, které se stanou podkladem pro akceptační testování, či user acceptance testing, známé i pod zkratkou UAT.

Nejdůležitějším použitým pojmem je testovací případ (test case). Jedná se o soubor podmínek a soupis kroků, pro dosažení určitého cíle. Podrobnější popis se nachází v dalších odstavcích. Testovací scénář (test scenario) je tvořen více testovacími případy, které na sebe navazují. Jejich použití není pro testování nezbytné, v některých případech postačí jen testovací případy. Pojem testovací script (test script) je určen pro automatická testování. Jedná se tedy o soubor programových příkazů.

4.6.1 Tvorba testovacích případů

1. Co je testovací případ?

Testovací případy ideálně vychází z případů užití. V praxi se často stává, že případy užití nejsou plně vypracované (což je ale chyba), takže je nutné testy připravit například jen na základě uživatelských požadavků. Během této práce byly případy užití připraveny dostatečně kvalitně, aby mohli posloužit jako plnohodnotný podklad pro přípravu testovacích případů. Z jednoho případu užití často vznikne více testovacích případů. Smyslem jednotlivých testů může být otestování základního, nejjednoduššího průchodu, kontrola správnosti uživatelského rozhraní či alternativní průchody případem užití a validace vstupních dat. Testuje se i záměrná simulace různých chybových stavů, například pokud je některý z modulů systému nedostupný. Zde se testuje, zda se aplikace s chybou vyrovná a nehavaruje.

2. Volba způsobu zápisu testovacích případů

Obecně odborná literatura popisuje tři druhy zápisu UAT testovacích případů. První z nich jsou formalizované testy. Ty jsou psány nejpodrobněji, předpokládá se, že osoba která dle nich bude testovat, nemá hluboké znalosti o fungování systému. Nevýhodou tohoto druhu testů je jejich časově náročnější příprava. Neformalizované testy již nejsou tak podrobné. V rámci těchto testovacích případů je popsán cíl, jehož má uživatel dosáhnout, samotný postup je popsán pouze rámcově. Předpokládá se, že tester bude mít hlubší znalosti testovaného softwaru. Třetím typem jsou badatelské testy. Tyto testy může provádět pouze tester s mnoha zkušenostmi a hlubokými znalosti příslušného softwaru. Testy neobsahují v podstatě žádný popis, tester testuje na základě vlastního uvážení. V praxi se tyto testy nazývají „free testy“.

Formalizované testy

Na základě možností popsaných výše, byl zvolen typ zápisu formalizované testy, jelikož tímto způsobem bylo dosaženo nejpodrobnějšího popisu testovaného procesu. Postup byl zapisován v jednotlivých krocích, jejichž první část je akce uživatele, druhá část je reakce aplikace. Vznikl takto jakýsi dialog mezi uživatelem a systémem.

Jednotlivé kroky byly zapisovány stručně, ale tak aby byly jednoznačné a snadno pochopitelné – tester, který dle nich měl testovat, nemusel mít hluboké znalosti fungování aplikace, především z hlediska backendu. Bylo třeba počítat i s možností, že se během projektu zadání (případy užití) mohly měnit.

4.6.2 Obsah testovacího případu

Standard ansi/ieee 829 definuje přesné informace, které má testovací případ obsahovat. Odborná literatura ale dodává, že se jedná o doporučení, které není nutné vždy přesně dodržet. Analytik má tedy možnost počet těchto popisných atributů upravit tak, aby při testování na konkrétním projektu dávaly smysl.

Na základě odborné literatury byly definovány tyto parametry, které musí být součástí každého případu užití:

Název – Stručný název testovacího případu.

Jednoznačný identifikátor – Krátký kód, díky kterému bylo možné každý jednotlivý testovací případ dohledat. Není nikde definováno pravidlo, jak se tyto identifikátory mají tvořit. V této práci byla použita jednoduchá metoda prostého číslování. První testovací případ nese označení TP001, druhý TP002 atd.

Stručný popis – Zde bylo v přirozeném jazyce stručně vysvětleno, co má testovací případ za cíl otestovat. Nebylo třeba problém popisovat příliš podrobně, jelikož v případě strukturovaných testů byl podrobně popsán jednotlivými kroky.

Vazba na případ užití - Odkaz na identifikátor případu užití, ze kterého tento testovací případ vychází.

Vstupní podmínky – Skutečnosti, které musí být splněny pro zahájení testovacího případu. Důležitou vstupní podmínkou byla role, kterou uživatel disponuje. Některé testovací případy bylo možné provést pouze s rolí administrátor.

Výstupní podmínky – Podmínky, které musely být splněny, aby mohl být testovací případ uzavřen jako úspěšný, bez identifikování defektu.

Priorita - Prioritu určil analytik. Vycházela z druhu testu, případně mohla být ovlivněna i prioritou případu užití, ze kterého vycházela.

Není přesně předepsáno, kolika stupni priority mají být testovací případy definovány. V praxi se většinou využívají stupně tři nebo pět. V této práci byly použity tři stupně – Low, Normal a High.

Rozdělení priorit bylo velmi důležité z několika důvodů.

Při přípravě testů počítal analytik s tím, že testování mohlo být prováděno v několika fázích, například na různých testovacích prostředích. Za určitých okolností dává smysl v některých fázích testovat jen některé testy – ty s vyšší prioritou. Mohla nastat i situace, kdy bylo třeba aplikaci testovat na více platformách, např. na různých internetových prohlížečích. Většinou není z časových důvodů možné otestovat vždy všechny testovací případy, testují se tedy všechny testy na jedné platformě a na ostatních pouze testovací případ s vysokou prioritou.

Stupeň High značil nejvyšší prioritu. V případě této práce se jednalo o nejjednodušší cestu k dosažení cíle.

Stupeň Normal představoval střední prioritu. V této práci byly tímto stupněm nejčastěji označeny testovací případy, sloužící ke kontrole uživatelského rozhraní.

Stupeň Low označoval testovací případy, které měly za cíl záměrně dosáhnout chybového stavu. Testovalo se, zda aplikace nehavala. Jednalo se o test, který bylo nutné provést. V případě, kdy bylo testování prováděno například na více platformách, bylo možné tento test v některých případech vynechat.

4.6.3 Podrobný postup tvorby testovacích případů

Naposledy se vraťme k případům užití, vytvořených v jedné z minulých kapitol. Ty nyní posloužily jako podklad pro vytvoření testovacích případů.

Testovací případ vytvořený dle PU004

Dle PU004 byl vytvořen pouze jeden testovací případ, jelikož případ užití je poměrně jednoduchý, není zde možnost simulace záměrné chyby, testující zda aplikace nehavaruje.

Vstupní i výstupní podmínky jsou v tomto případě stejné, jako u příslušného případu užití.

Priorita je zde ale hodnocena jinak než v případě PU. Jak je popsáno výše, prioritu určil analytik dle druhu testu. Tento test je nejjednodušší cestou k dosažení cíle, proto mu byla přiřazena priorita High.

Samotné kroky testovacího případu jsou dva. Uživatel zahájí případ volbou zobrazení grafického schématu, což aplikace provede. Druhým krokem je uživatelská volba požadovaného dne. Aplikace poté zobrazí schéma pro datum, dle uživatelského výběru.

Výsledný testovací případ:

TP007: Grafické schéma rezervací zasedaček

Stručný popis: Test zobrazení grafického schéma rezervací zasedaček pro určité datum a následné změny tohoto data.

Vazba na případ užití: UP004

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Uživateli je zobrazeno grafické schéma rezervací zasedaček pro datum dle uživatelského výběru.

Priorita: High

	Uživatel:	Systém:
1.	Zvol možnost pro zobrazení Grafického schématu.	Systém provede zobrazení Grafického schématu pro aktuální den.
2.	Vyber možnost pro změnu data.	Systém zobrazí Grafické schéma pro datum dle uživatelského výběru.

Testovací případ vytvořený dle PU005

Případ užití UP005 se stal podkladem také pouze pro jeden testovací případ.

Vstupní podmínky jsou opět stejné, jako v případě užití.

Byla zde ale přidána podmínka, pro existenci testovacích dat. Těmi je několik zasedacích místností v rámci patra a v rámci budovy, na kterých je vytvořeno několik rezervací. Bez těchto dat by nebylo možné testovací případ provést.

Výstupní podmínkou je, aby systém úspěšně provedl filtrování dle všech vyhledávacích možností.

Priorita byla nastavena na High, jelikož se stejně jako v minulém případě jedná o nejjednodušší cestou k dosažení cíle.

Uživatel má dle vstupních podmínek již zobrazen přehled rezervací.

Jsou zde tři možnosti filtrování, v testovacím případě byl pro každý způsob vytvořen samostatný krok. Pro každou z možností uživatel nejprve volí příslušnou možnost, aplikace poté zobrazuje možnost zadání vyhledávacích kritérií – na základě výběru filtrovací možnosti.

Po jejich zadání a potvrzení uživatelem aplikace provede filtrování a zobrazí požadovaný přehled.

Výsledný testovací případ:

TP008: Filtr zobrazení v grafickém schématu rezervací zasedaček

Stručný popis: Test filtrování v grafickém schématu rezervací zasedaček skrze rezervace jedné zasedačky, všech zasedaček v určité budově a v podlaží.

Vazba na případ užití: UP005

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace. Existuje několik zasedacích místností v rámci patra a v rámci budovy, na kterých je vytvořeno několik rezervací. Uživatel má zobrazen přehled rezervací.

Výstupní podmínky: Systém provedl filtrování zobrazených zasedaček, dle všech možností.

Priorita: High

	Uživatel:	Systém:
1.	Zvol zobrazení všech zasedaček v určité budově.	Systém, dle uživatelova výběru, zobrazí možnosti zadání vyhledávacích kritérií.
2.	Uživatel zadá vyhledávací kritéria a vyhledání potvrdí.	Aplikace zobrazí grafický přehled obsazenosti zasedaček či zasedačky dle uživatelova výběru, defaultně pro aktuální den.
3.	Zvol zobrazení všech zasedaček v určitém podlaží.	Systém, dle uživatelova výběru, zobrazí možnosti zadání vyhledávacích kritérií.

4.	Uživatel zadá vyhledávací kritéria a vyhledání potvrdí.	Aplikace zobrazí grafický přehled obsazenosti zasedaček či zasedačky dle uživatelova výběru, defaultně pro aktuální den.
5.	Zvol zobrazení pouze jedné konkrétní zasedačky.	System, dle uživatelova výběru, zobrazí možnosti zadání vyhledávacích kritérií.
6.	Uživatel zadá vyhledávací kritéria a vyhledání potvrdí.	Aplikace zobrazí grafický přehled obsazenosti zasedačky dle uživatelova výběru, defaultně pro aktuální den.

U testovacích případech, které byly dostupné pro obě role - Uživatel i Administrátor, bylo nutné provést test pro každou roli zvlášť. Toto se při plánování řešilo rozvržením testů do složek. Vznikla tak série testů zvlášť pro obě role.

4.6.4 Kontrola pokrytí případů užití testovacími případy

Důležitým krokem bylo provedení kontroly, zda každý případ užití je pokryt alespoň jedním testovacím případem. Tato kontrola byla provedena skrze jednoduchý nákres, vytvořený skrze program Excel.

Obrázek 17: Přehled pokrytí případů užití testovacími případy

Případy užití	Testovací případy
PU001	TP001
— „ —	TP002
— „ —	TP003
— „ —	TP004
PU002	TP005
PU003	TP006
PU004	TP007
PU005	TP008
PU006	TP009
PU007	TP010
— „ —	TP011
PU008	TP012
PU009	TP013
— „ —	TP014
PU010	TP015
— „ —	TP016
PU011	TP017
PU012	TP018
PU013	TP019
PU014	TP020
— „ —	TP021

4.6.5 Software

Bylo více možností, v jakém softwaru testovací případy evidovat a kde provádět jejich exekuci. V případě této práce nebyl použit žádný specializovaný software, testovací případy byly evidovány v programu MS Word.

O použití specializovaného software ale bylo uvažováno, jelikož nabízí řadu výhod. Skrze tento software je možno skrze odkazy propojit konkrétní testovací případy s příslušnými případy užití, ze kterých vychází, což usnadňuje kontrolu pokrytí všech případů užití alespoň jedním testovacím případem.

Bugy lze evidovat také v tomto systému a je možné je založit rovnou skrze konkrétní krok, v testovacím případě. Další možností může být lepší organizace a využití znovupoužívání testů v případě, kdy se testovací případy testují znovu v různých fázích projektu. Pokud exekuci testů provádí méně zkušenější testéři, ocení intuitivní značení, které kroky v testovacím případě byly úspěšně provedené, které obsahují bug, či které jsou právě

exekuvány – toto je velmi praktické, pokud testy provádí více testerů. Software počítá i s možnostmi, kdy kroky v testovacím případě obsahují chybu.

Nejpoužívanějším softwarem je Jira od společnosti Atlassian a HP Quality Center, aktuálně vyvíjen firmou Hewlett-Packard. V obou případech se jedná o placený software, který nemá žádnou verzi pro použití zdarma.

4.6.6 Výsledky kapitoly

Jak již bylo řečeno v závěru kapitoly Případy užití a scénáře užití, testovací případy byly spolu s případy užití zaslány zadavatelské firmě ke kontrole.

Firma byla spokojená a souhlasila se zahájením samotného vývoje aplikace, založeném na těchto analýzách.

5 Výsledky a doporučení

5.1 Výsledky

Dle zadání byl připraven postup pro získání uživatelských požadavků, do kterého byly zahrnuti zástupci vedení zadavatele i zástupci řadových zaměstnanců. Jako metody byly zvoleny "požadavkový meeting" a "pozorování uživatelů při práci". Takto bylo popsáno celkem čtrnáct uživatelských požadavků.

Tyto požadavky posloužily pro přípravu případů užití. V práci je popsán způsob, kterým bylo vytvořeno čtrnáct případů užití.

Dále se případy užití staly podkladem pro tvorbu návrhu uživatelského rozhraní a testovacích případů uživatelského akceptačního testování (UAT).

Dle zadání zakázky, byly nyní případy užití a testovací případy odeslány zadavateli, ke schválení. Ten posoudil, zda analýzy odpovídají jeho představě. Díky případům užití mohl posoudit, jakých cílů bude skrze aplikaci možné dosáhnout a díky testovacím případům získal představu, jakým způsobem bude posuzována konečná kvalita vytvořené aplikace. Zadavatel dodané analýzy akceptoval.

Případy užití se staly základním kamenem modelu požadavků. Poté byl zahájen samotný vývoj aplikace, kterého se již autor této práce přímo neúčastnil. Model požadavků, byl předán, k vytvoření technických analýz a zahájení programování. Návrh uživatelského rozhraní v podobě wireframes, vytvořené autorem této práce, byly předány designerovi k vytvoření grafického designu - tedy dokumentu popisujícím finální vzhled aplikace.

Uživatelské akceptační testování bylo založeno na testovacích případech, připravených též v rámci této práce. Při testování byly odhaleny buggy, blokující některé funkce. Tyto buggy byly následně programátory opraveny a testery úspěšně retestovány. Nakonec testování potvrdilo, že aplikace je v akceptovatelném stavu, jelikož žádný z testovacích případů nebyl blokován bugem.

Následně byla aplikace předána klientovi, který ji akceptoval.

Výsledkem této práce je úspěšné vytvoření návrhu modelu požadavků a testovací analýzy, která byla akceptována zadavatelem a na které byl založen vývoj aplikace.

5.2 Doporučení

Jak již bylo řečeno v kapitole Cíle, účelem aplikace v této fázi není řešení databáze zaměstnanců. V budoucnu doporučuji tuto aplikaci na databázi zaměstnanců napojit. Důvodem je výrazně efektivnější dlouhodobá udržitelnost. Velmi užitečnou funkcí by také byla možnost evidence zaměstnanců, pozvaných na rezervovaný termín. K tomu je třeba výše zmíněné napojení na databázi zaměstnanců. Dalším doporučením je napojení na systém, který by odesílal uživatelům e-mailové notifikace ohledně událostí v aplikaci. Takovými událostmi může být přiřazení k nově vytvořené rezervaci, změny ve stávajících rezervacích či zrušení uživateli rezervace Administrátorem a podobně. Kvalitu použitelnosti aplikace by též zvýšilo napojení na systém kalendáře, jenž by uživatelům poskytoval časový přehled rezervací, které uživatel vytvořil, nebo na které je pozván. Aplikace taktéž v této fázi neřeší front endový způsob registrace nového uživatele, či editace jeho přihlašovacích údajů. Doporučujeme, aby registrace a editace přihlašovacích údajů byly propojeny s výše zmíněnými systémy zaměstnaneckých e-mailů a kalendáře. Posledním doporučením, je vytvoření anglické jazykové mutace.

6 Závěr

Záměrem této práce bylo provést studii uživatelských požadavků, vytvořit jejich evidenci skrze případy užití, navrhnout schematický model uživatelského užití a připravit testovací analýzu, pro akceptační testování.

Teoretická část této práce, tvoří základ pro praktickou část. Je zde obsaženo představení problematiky jednotlivých kapitol. Tato část také obsahuje citace a odkazy na zdroje, tvořené odbornou literaturou, které obsahují mnoho dalších informací, případně i popisy jiných možností, jak dané postupy provést.

Praktická část byla rozdělena do pěti hlavních kapitol. V první z nich byly popsány dvě použité metody sběru uživatelských požadavků a jejich způsob jejich zápisu pro další použití. Ve druhé kapitole byly z těchto požadavků vytvořeny případy užití. Následující kapitola popisuje tvorbu diagramu případů užití, pro lepší orientaci v popsáných případech užití a uživatelských rolí. Čtvrtá kapitola se věnuje tvorbě schematického návrhu modelu uživatelského rozhraní. V poslední kapitole je popsán způsob tvorby testovací analýzy. Výsledky, vytvořené v rámci těchto kapitol, tvoří přílohu této práce.

Skrze nové řešení aplikace byly odstraněny nedostatky aplikace původní. Byla přidána asistence pro případ zapomenutí hesla, možnost zobrazení plánu požadovaného podlaží bez nutnosti vyhledání konkrétní zasedací místnosti a vytvoření grafického přehledu rezervací. Díky metodě pozorování uživatelů při práci, byly odhaleny a vyřešeny další dva nedostatky – byla přidána možnost přímého zobrazení plánu podlaží skrze detail rezervace a absence nutnosti zadat navíc název podlaží při vyhledávání dle názvu zasedačky.

Tato práce může být využita jako zdroj informací či inspirace při vytváření analýzy, pro vývoj menších podnikových aplikací – zejména způsob získávání uživatelských požadavků a jejich evidence skrze případy užití. Text obsahuje odkazy na zdroje, které poskytují i jiné možnosti řešení, než byly nakonec použity v této práci. Příkladem může být použití jiné metody pro získání uživatelských požadavků, lze využít například metody prototypování, nebo dotazníky.

7 Seznam použitých zdrojů

1. Rascasone.com. INFORMAČNÍ SYSTÉMY V KOSTCE: ERP, CRM, IMPLEMENTACE. [online]. (html). [cit. 2021-02-28] Dostupné z <https://www.rascasone.com/cs/blog/informacni-systemy-erp-crm-implemetace>
2. SODOMKA, Petr, KLČOVÁ, Hana. Informační systémy v podnikové praxi. 2., aktualiz. a rozš. vyd. Brno: Computer Press. (2010) ISBN 978-80-251-2878-7
3. Rascasone.com. CO JE PODNIKOVÝ SOFTWARE A PROČ BYSTE O NĚM MĚLI PŘEMÝŠLET? [online]. (html). [cit. 2021-02-28] Dostupné z <https://www.rascasone.com/cs/blog/podnikovy-enterprise-software>
4. WIEGERS, Karl Eugene. Požadavky na software. Brno: Computer Press, 2008. ISBN 978-80-251-1877-1.
5. FOWLER, Martin. Destilované UML. Praha: Grada, 2009. Knihovna programátora (Grada). ISBN 978-80-247-2062-3.
6. COCKBURN, Alistair. Writing effective use cases. Boston: Addison-Wesley, c2001. ISBN 02-017-0225-8.
7. Moodle ČVUT, Ing. Martin Komárek. Model případu užití. (PDF). [cit. 2020-03-20] Dostupné z WWW: https://moodle.fel.cvut.cz/pluginfile.php/47115/mod_label/intro/PripadyUziti_UC_prednaska_4a5.pdf
8. Tutorialspoint.com. UML - Building Blocks [online]. (htm) [cit. 2020-03-20] Dostupné z WWW: https://www.tutorialspoint.com/uml/uml_building_blocks.htm
9. ARLOW, Jim a Ila NEUSTADT. UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.

10. Itnetwork.cz. UML - Use Case Specifikace. (html). [cit. 2020-03-20]. Dostupné z <https://www.itnetwork.cz/navrh/uml/uml-use-case-specifikace-diagram>.

11. Česká zemědělská univerzita v Praze, Praha – Suchbát. Ing. Josef Pavlíček Ph.D. , Bc. Karolína Solanská a kol. Učebnice Interakce člověk počítač

12. PATTON, Ron. Testování softwaru. Praha: Computer Press, 2002. Programování. ISBN 80-722-6636-5.

13. ROMAN, Adam. The ISTQB Foundation level 2018 syllabus: test techniques and sample exams. Verze 2018. New York, NY: Springer Berlin Heidelberg, 2018. ISBN 978-331-9987-392.

14. GARRETT, Jesse James. The elements of user experience: user-centered design for the web and beyond. 2nd Edition. Berkeley, CA: New Riders, c[2011]. ISBN 9780321683687.

8 Přílohy

8.1	Uživatelské požadavky.....	67
8.2	Vzorové osoby	69
8.2.1	Primární vzorové osoby	69
8.2.2	Sekundární vzorové osoby.....	70
8.3	Případy užití a scénáře užití	71
8.4	Model uživatelského rozhraní	80
8.5	Testovací případy uživatelského akceptačního testování	82

8.1 Uživatelské požadavky

UP001: Přihlášení uživatelskými účty

Priorita: Vysoká

Popis: Potřebuji se do aplikace přihlásit skrze uživatelské jméno a heslo, jelikož každý uživatel má svůj osobní účet.

UP002: Zobrazení plánu kanceláří

Priorita: Vysoká

Popis: Potřebuji si zobrazit plány kanceláří.

UP003: Vyhledávání v plánech kanceláří

Priorita: Střední

Popis: Na plánech kanceláří potřebuji mít možnost vyhledávání určitého patra, zasedačky a místa, kde sedí určitý zaměstnanec. Po vyhledání patra potřebuji, aby se mi zobrazil plán příslušného patra. Pokud vyhledám zasedačku či místo, kde sedí určitý zaměstnanec, zobrazí se mi plán příslušného patra s graficky zvýrazněným hledaným objektem.

UP004: Grafické schéma rezervací zasedaček

Priorita: Vysoká

Popis: Potřebuji mít možnost zobrazení grafického schématu znázorňujícího existující rezervace jedné či více zasedaček pro určitý den. Přehled je možné posouvat po dnech.

UP005: Filtr zobrazení v grafickém schématu rezervací zasedaček

Priorita: Střední

Popis: Potřebuji, aby v rámci grafického schématu rezervací zasedaček byla možnost výběru zobrazení všech zasedaček v určité budově, v podlaží či pouze jedné zasedačky.

UP006: Detail existující rezervace

Priorita: Střední

Popis: Skrze grafické schéma si potřebuji zobrazit detail rezervace, kde je uveden zaměstnanec, který rezervaci vytvořil.

UP007: Nová rezervace zasedačky

Priorita: Střední

Popis: Potřebuji mít možnost vytvořit novou rezervaci zasedačky.

UP008: Přehled rezervací přihlášeného uživatele

Priorita: Nízká

Popis: Potřebuji mít možnost zobrazení přehledu všech mých aktuálních rezervací.

UP009: Úprava rezervace přihlášeného uživatele

Priorita: Nízká

Popis: Potřebuji, aby mé vlastní rezervace bylo možné upravovat.

UP010: Smazání rezervace přihlášeného uživatele

Priorita: Nízká

Popis: Potřebuji, aby mé vlastní rezervace bylo možné mazat.

UP011: Proklik z detailu vlastní rezervace na plán kanceláří

Priorita: Nízká

Popis: Potřebuji, aby bylo možné z detailu vlastní rezervace jedním klikem zobrazit plán kanceláří se zvýrazněním příslušné zasedačky.

UP012: Zobrazení přehledu pracovních míst spolu se jmény zaměstnanců

Priorita: Nízká

Popis: S administrátorskou rolí potřebuji mít možnost zobrazení přehledu pracovních míst spolu se jmény zaměstnanců, kteří jsou k těmto místům přiřazeni.

UP013: Editace jmen zaměstnanců

Priorita: Nízká

Popis: S administrátorskou rolí potřebuji mít možnost editovat jména zaměstnanců v rámci přehledů pracovních míst.

UP014: Rušení existujících rezervací administrátorem

Priorita: Nízká

Popis: S administrátorskou rolí potřebuji mít možnost rušit již existující rezervace - například z důvodu konání prioritního meetingu.

8.2 Vzorové persony

8.2.1 Primární vzorové persony

Jméno: Jana Malá

Pohlaví: Žena

Věk: 41

Pracovní pozice: Účetní

Stručný životopis: Po absolvování gymnázia zahájila studium oboru Účetnictví na VŠE. Během studií si našla práci jako účetní a vdala se. Po dokončení studií odešla na sedmiletou mateřskou dovolenou. Od té doby pracuje na pozici účetní, stále ve stejné firmě, ve které pracovala během studií

Životní cíle: Náplní jejího života je především péče o její rodinu. Je spokojená, pokud jsou spokojeni všichni členové její rodiny.

Zkušenosti s prací na PC: Počítače používá pouze pro svou práci. Ve volném čase má jiné zájmy.

Typický den: Vstává brzy ráno, vyvenčí psy a chystá rodině snídani. Do práce poté dojíždí MHD. Po návratu z práce se věnuje dětem a domácím mazlíčkům. Večer čte knihy nebo sleduje filmy.

Koníčky: Četba knih

Jméno: Tomáš Kratochvíl

Pohlaví: Muž

Věk: 38

Pracovní pozice: Projektový manažer

Stručný životopis: Již od raného věku byl Tomáš workoholikiem. Studia a později práci vždy kladl ve svém životě na první místo. Absolvoval několik vysokých škol v tuzemsku i v zahraničí. Vystřídal několik pracovních pozic včetně IT analytika, databázisty a programátora. V současnosti pracuje na pozici Projektového manažera.

Životní cíle: Pro Tomáše je jeho životní náplní práce. Chce stále získávat nové zkušenosti a dosahovat stále vyšších úspěchů. V budoucnu by si rád založil vlastní IT firmu.

Zkušenosti s prací na PC: Díky jeho zkušenostem na různých pracovních pozicích jsou jeho zkušenosti s prací na počítači hluboké a rozsáhlé.

Typický den: Do zaměstnání odjíždí před úsvitem a domů se vrací po setmění. Na nic jiného, než na práci, není v jeho životě prostor.

Koníčky: Jeho práce je zároveň i jeho koníčkem.

8.2.2 Sekundární vzorové osoby

Jméno: Jakub Vyšohlíd

Pohlaví: Muž

Věk: 44

Pracovní pozice: Údržbář

Stručný životopis: Po ukončení základního vzdělání Jakub vystřídal mnoho různých zaměstnání. Kladl velký důraz na to, aby ho práce bavila a naplňovala. Nejdéle pracoval jako technik v divadle, či jako údržbář v pražské MHD. Důležitou částí jeho života je fotbal, kterému se dříve aktivně věnoval. V současné době jej pouze sleduje v televizi.

Životní cíle: Jakub je se svým současným životem spokojen a nemá v plánu nic měnit.

Zkušenosti s prací na PC: Počítače Jakuba nezajímají, se základními způsoby použití ale problém nemá.

Typický den: Brzy ráno odchází do zaměstnání. Po práci rád navštěvuje svou oblíbenou restauraci, kde popíjí s přáteli.

Koníčky: Sledování fotbalu, návštěvy restaurací.

Jméno: Jiří Koutný

Pohlaví: Muž

Věk: 21

Pracovní pozice: Brigádník – junior tester

Stručný životopis: Jiří pochází z Jižních Čech, kde studoval na střední průmyslové škole. Po maturitě se přestěhoval do Prahy, kde nyní studuje informatiku na soukromé vysoké škole. Během studií si brigádně přivydělává jako softwarový junior tester.

Životní cíle: Po dokončení svých studií plánuje několik let pracovat v zahraničí. Má ale v plánu se později vrátit a v České republice si založit rodinu.

Zkušenosti s prací na PC: Práce s počítači je Jirkovým denním chlebem. Pracuje s nimi ve škole, v práci i ve volném čase. Svými schopnostmi a zkušenostmi převyšuje běžné uživatele.

Typický den: Jiřího dny jsou nepravidelné. Některé dny je nucen vstávat brzy do školy, jiné dny vstává brzy do práce. V případě, kdy ani z jednoho důvodu vstávat nemusí, rád prospí celé dopoledne. Po splnění jeho denních povinností rád hraje počítačové hry, často až do pozdních nočních hodin.

Koníčky: Hraní počítačových her, četba fantasy literatury.

8.3 Případy užití a scénáře užití

PU001: Přihlášení uživatelskými účty

Priorita: Vysoká

Uživatelský požadavek: UP001

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor.

Výstupní podmínky: Uživatel je přihlášen do aplikace.

Případ užití:

Uživatel očekává možnost přihlásit se do aplikace skrze uživatelské jméno a heslo, jelikož každý uživatel má svůj osobní účet.

- a. Očekává možnost zadání přihlašovacího jména a hesla.
- b. Očekává možnost obnovení hesla, v případě jeho zapomenutí .
- c. Očekává možnost potvrdit zadání přihlašovacího jména a hesla a vstoupit tak to aplikace.

Scénář užití:

1. Uživatel zadá přihlašovací jméno a heslo.
2. Systém zaktivní možnost potvrzení přihlášení.
3. Uživatel zvolí možnost pro resetování hesla v případě, kdy jej zapomněl.
4. Systém uživatele upozorní, že mu na mail poslal instrukce k obnovení hesla.
5. Uživatel zvolí možnost pro vstup do aplikace.
6. V případě, že uživatel zadal špatné přihlašovací údaje, systém na to uživatele upozorní. Klient má možnost zadat údaje znovu. V případě, kdy jsou uživatelské údaje správné, systém klienta přihlásí a zobrazí mu úvodní obrazovku.

PU002: Zobrazení plánu kanceláří**Priorita: Vysoká****Uživatelský požadavek: UP002**

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Systém uživateli zobrazil plány kanceláří.

Případ užití:

Uživatel očekává možnost zobrazení plánu kanceláří. Najednou je zobrazeno pouze jedno podlaží, uživatel může zobrazené podlaží změnit.

Scénář užití:

1. Uživatel zvolí možnost pro zobrazení plánu kanceláří.
2. Systém zobrazí plány kanceláří.
3. Uživatel vybere možnost pro změnu zobrazeného podlaží.
4. Systém zobrazí podlaží dle uživatelova výběru.

PU003: Vyhledávání v plánech kanceláří**Priorita: Střední****Uživatelský požadavek: UP003**

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Uživateli je zobrazen plán podlaží, dle výsledku jeho vyhledávání.

Případ užití:

Uživatel očekává možnost vyhledávání a následné grafické zvýraznění v plánech kanceláří.

- a. Očekává možnost vyhledání a zobrazení určitého podlaží.
- b. Očekává možnost vyhledání a zobrazení určitého podlaží dle požadované zasedačky. Zasedačka bude barevně zvýrazněna.
- c. Očekává možnost vyhledání a zobrazení určitého podlaží dle jména zaměstnance, který pracuje na určitém pracovním místě. Pracovní místo bude barevně zvýrazněno.

Scénář užití:

1. Uživatel zvolí možnost pro zobrazení plánu kanceláří.
2. Systém zobrazí plány kanceláří.
3. Uživatel zvolí, zda chce zobrazit určité podlaží či chce zobrazit podlaží, kde je určitá zasedačka či podlaží, kde se nachází pracovní místo některého zaměstnance.
4. Systém dle uživatelova výběru zobrazí možnosti zadání vyhledávacích kritérií.
5. Uživatel zadá vyhledávací kritéria a vyhledání potvrdí.
6. Aplikace zobrazí plán příslušného patra. Pokud uživatel vyhledával pracovníka či zasedačku, příslušný objekt je graficky zvýrazněn.

PU004: Grafické schéma rezervací zasedaček

Priorita: Vysoká

Uživatelský požadavek: UP004

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Uživateli je zobrazeno grafické schéma rezervací zasedaček pro datum dle uživatelova výběru.

Případ užití:

Uživatel očekává možnost zobrazení grafického schéma znázorňujícího existující rezervace zasedaček pro určitý den. Přehled je možné posouvat po dnech.

Scénář užití:

1. Uživatel zvolí možnost pro zobrazení grafického schématu.
2. Systém provede zobrazení grafického schématu pro aktuální den.
3. Uživatel vybere možnost pro změnu data.
4. Systém zobrazí grafické schéma pro datum, dle uživatelova výběru.

PU005: Filtr zobrazení v grafickém schématu rezervací zasedaček**Priorita: Střední****Uživatelský požadavek: UP005**

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace. Uživatel má zobrazen přehled rezervací.

Výstupní podmínky: Systém provedl filtrování zobrazených zasedaček, dle všech možností.

Případ užití:

Uživatel v grafickém schématu očekává možnost filtrování zobrazených zasedaček.

- a. Očekává možnost zobrazení všech zasedaček v určité budově
- b. Očekává možnost zobrazení pouze jedné zasedačky
- c. Očekává možnost zobrazení všech zasedaček v určitém podlaží.

Scénář užití:

1. Uživatel zvolí, zda chce v grafickém přehledu zobrazit zasedačky v určité budově, v podlaží či pouze jedné zasedačky.
2. Systém, dle uživatelova výběru, zobrazí možnosti zadání vyhledávacích kritérií.
3. Uživatel zadá vyhledávací kritéria a vyhledání potvrdí.
4. Aplikace zobrazí grafický přehled obsazenosti zasedaček či zasedačky, dle uživatelova výběru, defaultně pro aktuální den.

PU006: Detail existující rezervace**Priorita: Střední****Uživatelské požadavky: UP006**

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace. V systému existuje alespoň jedna rezervace.

Výstupní podmínky: Uživateli je zobrazen detail požadované rezervace, obsahující všechny požadované atributy.

Případ užití:

Uživatel očekává možnost zobrazení detailu existující rezervace.

Scénář užití:

1. Uživatel zvolí možnost pro zobrazení Grafického schématu.
2. Systém provede zobrazení Grafického schématu.
3. Uživatel provede vyhledání/filtrování tak, aby mu byly zobrazeny rezervace požadované zasedačky, v požadovaný den.
4. Systém v přehledu zobrazí rezervace požadované zasedačky v požadovaný den.
5. Uživatel zvolí možnost zobrazení detailu rezervace.
6. Systém zobrazí detail rezervace. V tomto detailu jsou uvedeny tyto informace: Jméno zasedačky, Organizátor, Důvod rezervace, Datum, Čas od a Čas do.

PU007: Nová rezervace zasedačky

Priorita:

Uživatelský požadavek: UP007

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Byla vytvořena nová rezervace.

Případ užití:

Uživatel očekává možnost vytvoření nové rezervace zasedačky.

Scénář užití:

1. Uživatel zvolí možnost pro zobrazení Grafického schématu.
2. Systém provede zobrazení Grafického schématu.
3. Uživatel provede vyhledání/filtrování tak, aby mu byly zobrazeny rezervace požadované zasedačky, v požadovaný den.
4. Systém v přehledu zobrazí rezervaci požadované zasedačky v požadovaný den.

5. Uživatel vybere možnost pro vytvoření nové rezervace. Zadá Jméno zasedačky, Důvod rezervace, Datum, Čas od a Čas do. Vybere možnost pro uložení.

6. Systém provede validaci dat, která zadal uživatel.

Pokud jsou vstupní data nevalidní, nebo je termín již obsazen, systém na to uživatele upozorní.

Pokud jsou data validní a termín je volný, systém uloží novou rezervaci.

PU008: Přehled rezervací přihlášeného uživatele

Priorita: Nízká

Uživatelský požadavek: UP008

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Systém zobrazuje přehled rezervací přihlášeného uživatele.

Případ užití:

Uživatel očekává možnost zobrazení přehledu všech jeho aktuálních rezervací.

Scénář užití:

1. Uživatel vybere možnost pro zobrazení jeho aktuálních rezervací.
2. Systém uživateli zobrazí přehled jeho aktuálních rezervací.

PU009: Úprava rezervace přihlášeného uživatele

Priorita: Nízká

Uživatelský požadavek: UP009

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace. Uživatel disponuje alespoň jednou existující rezervací.

Výstupní podmínky: Proběhla úspěšná editace uživatelské rezervace.

Případ užití:

Uživatel očekává možnost editace jeho vlastních rezervací.

Scénář užití:

1. Uživatel vybere možnost pro zobrazení jeho aktuálních rezervací.

2. Systém uživateli zobrazí přehled jeho aktuálních rezervací.
 3. Uživatel vybere možnost pro editaci některé ze svých rezervací.
 4. Systém umožní uživateli provést editaci parametrů Důvod rezervace, Čas od a Čas do.
 5. Uživatel provede změnu a vybere možnost pro potvrzení.
 6. Systém provede validaci dat, která zadal uživatel.
- Pokud jsou vstupní data nevalidní, nebo je termín již obsazen, systém na to uživatele upozorní.

Pokud jsou data validní a termín je volný, systém uloží novou rezervaci.

PU010: Smazání rezervace přihlášeného uživatele

Priorita: Nízká

Uživatelský požadavek: UP010

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace. Uživatel disponuje alespoň jednou existující rezervací.

Výstupní podmínky: Proběhlo smazání uživatelovy rezervace.

Případ užití:

Uživatel očekává možnost smazání jeho vlastní rezervace.

Scénář užití:

1. Uživatel vybere možnost pro zobrazení jeho aktuálních rezervací.
2. Systém uživateli zobrazí přehled jeho aktuálních rezervací.
4. Uživatel vybere možnost pro smazání některé rezervace.
5. Systém se uživatele zeptá, zda si opravdu přeje tuto rezervaci smazat. Nabídne uživateli možnost pro potvrzení smazání rezervace a pro zrušení tohoto požadavku.
6. Uživatel zvolí možnost pro potvrzení smazání rezervace, nebo možnost pro zrušení tohoto požadavku.
7. Pokud uživatel zvolil možnost pro potvrzení smazání rezervace, systém rezervaci smaže.

Pokud uživatel zvolil možnost pro zrušení tohoto požadavku, systém rezervaci nesmaže.

PU011: Proklik z detailu vlastní rezervace na plán kanceláří

Priorita: Nízká

Uživatelský požadavek: UP011

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace. Uživatel disponuje alespoň jednou existující rezervací.

Výstupní podmínky: Proběhlo zobrazení plánu kanceláří, kde je zobazeno patro, na kterém se nachází příslušná zasedačka, jenž je barevně zvýrazněna.

Případ užití:

Uživatel očekává možnost prokliku z detailu vlastní rezervace na plán kanceláří, kdy bude graficky zvýrazněna příslušná místnost.

Scénář užití:

1. Uživatel vybere možnost pro zobrazení jeho aktuálních rezervací zasedaček.
2. Systém uživateli zobrazí přehled jeho aktuálních zasedaček.
4. Uživatel zvolí možnost pro zobrazení rezervované zasedačky na plánu kanceláří.
5. Systém uživateli zobrazí plány kanceláří, kde je zobazeno patro, na kterém se nachází příslušná zasedačka. Ta je graficky zvýrazněna.

PU012: Zobrazení přehledu pracovních míst spolu se jmény zaměstnanců

Priorita: Nízká

Uživatelský požadavek: UP012

Vstupní podmínky: Uživatel disponuje rolí Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Systém uživateli zobrazil seznam pracovních míst spolu se jmény zaměstnanců, kteří na daných místech pracují.

Případ užití:

Uživatel očekává možnost zobrazení přehledu pracovních míst, spolu se jmény zaměstnanců, kteří na daném místě pracují.

Scénář užití:

1. Uživatel vybere možnost pro zobrazení seznamu pracovních míst.

2. Systém zobrazí uživateli seznam pracovních míst. V rámci tohoto seznamu jsou uvedena i jména zaměstnanců, kteří na daných místech pracují.

PU013: Editace jmen zaměstnanců

Priorita: Nízká

Uživatelský požadavek: UP013

Vstupní podmínky: Uživatel disponuje rolí Administrátor. Uživatel je přihlášen do aplikace. Uživatel má zobrazen seznam pracovních míst.

Výstupní podmínky: Proběhla změna jména zaměstnance, který je přiřazen k určitému pracovnímu místu.

Případ užití:

Uživatel očekává možnost editovat jména zaměstnanců v rámci přehledů pracovních míst.

Scénář užití:

1. Uživatel vybere možnost pro editaci pracovního místa.
2. Systém umožní uživateli provést editaci parametrů Jméno a Příjmení.
3. Uživatel provede změnu a vybere možnost pro potvrzení.
4. Systém uloží změnu.

PU014: Rušení existujících rezervací administrátorem

Priorita: Nízká

Uživatelský požadavek: UP014

Vstupní podmínky: Uživatel disponuje rolí Administrátor. Uživatel je přihlášen do aplikace. Existuje alespoň jedna rezervace, kterou vytvořil jiný uživatel. Proběhl úspěšný průchod procesem PU006: Detail existující rezervace.

Výstupní podmínky:

Na základě uživatelova výběru proběhlo či neproběhlo smazání rezervace, vytvořené jiným uživatelem.

Případ užití:

Uživatel očekává možnost zrušení existující rezervace, kterou vytvořil jiný uživatel.

Scénář užití:

1. Uživatel na detailu rezervace vybere možnost pro smazání této rezervace.
2. Systém se uživatele zeptá, zda si opravdu přeje tuto rezervaci smazat. Nabídne uživateli možnost pro potvrzení smazání rezervace a pro zrušení tohoto požadavku.
3. Uživatel zvolí možnost pro potvrzení smazání rezervace, nebo možnost pro zrušení tohoto požadavku.
4. Pokud uživatel zvolil možnost pro potvrzení smazání rezervace, systém rezervaci smaže.

Pokud uživatel zvolil možnost pro zrušení tohoto požadavku, systém smazání rezervace neprovede.

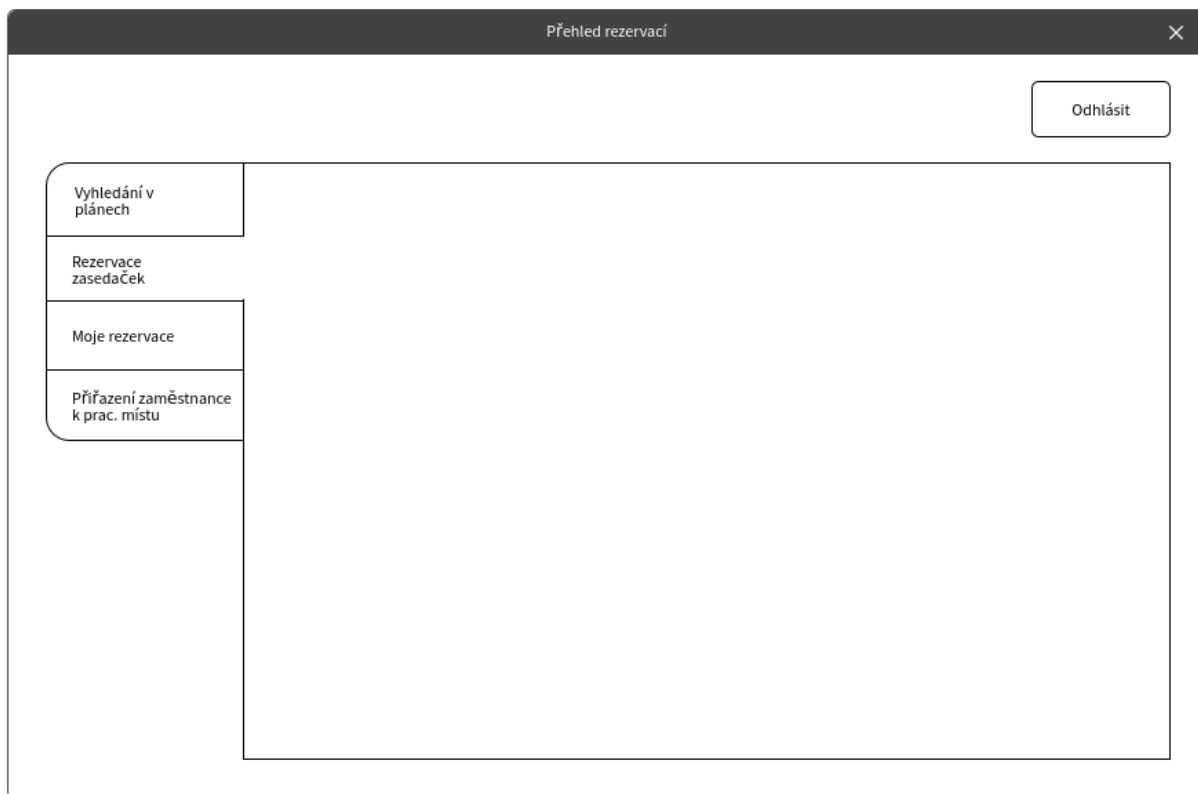
8.4 Model uživatelského rozhraní

Obrázek 18: WireFrame - Přihlašovací okno

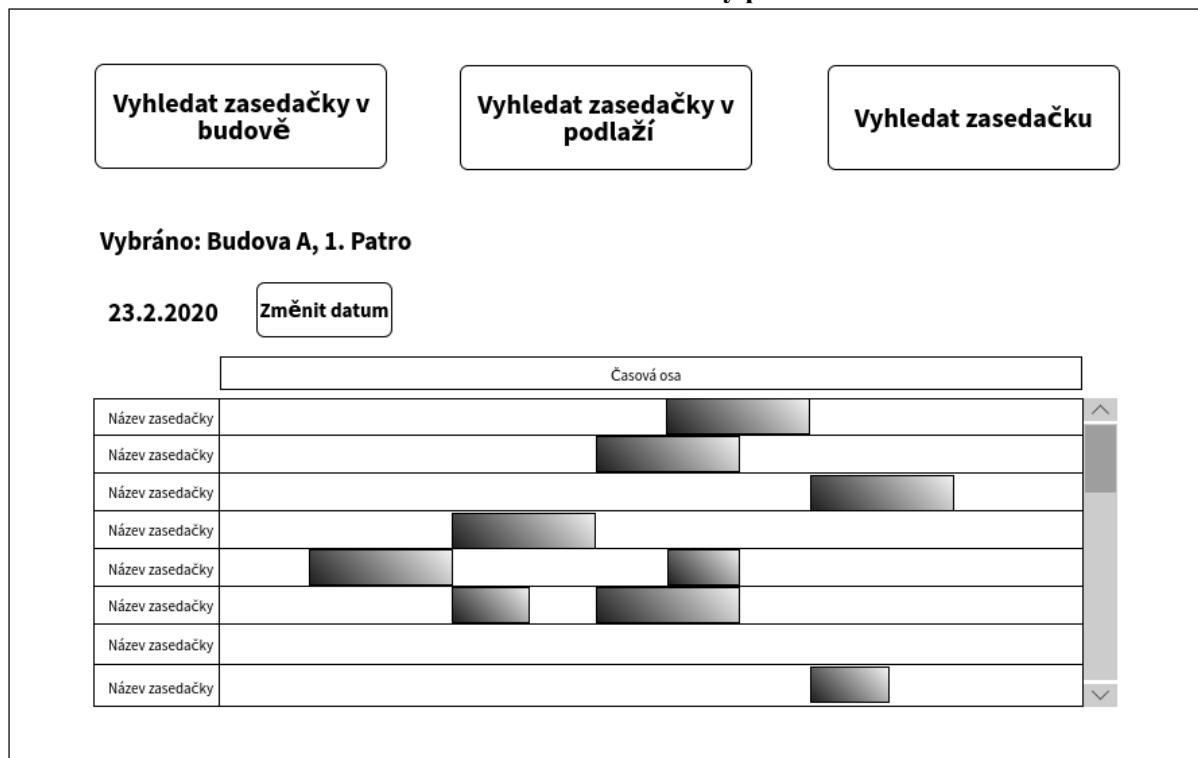
The wireframe shows a login window with a title bar containing three window control buttons and the text 'Přihlášení'. The main content area contains the following elements:

- A label 'Přihlašovací jméno:' followed by a text input field with the placeholder text 'Vyplnit přihlašovací jméno'.
- A label 'Heslo:' followed by a text input field with the placeholder text 'Vyplnit heslo'.
- A button labeled 'Zapomněl jsem heslo' located below the password field.
- A button labeled 'Přihlásit' located to the right of the 'Zapomněl jsem heslo' button.

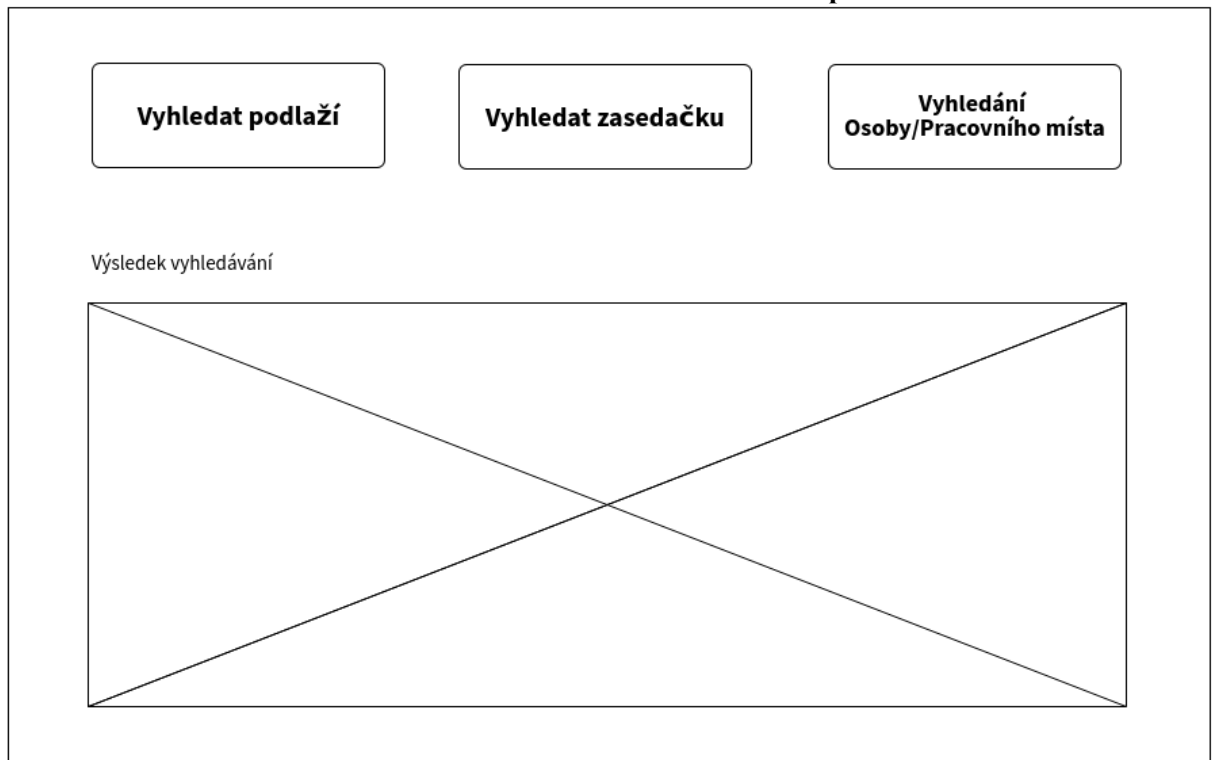
Obrázek 19: WireFrame - Hlavní menu



Obrázek 20: WireFrame - Grafický přehled rezervací



Obrázek 21: WireFrame - Zobrazení plánů



8.5 Testovací případy uživatelského akceptačního testování

TP001: Přihlášení uživatele s rolí Uživatel

Stručný popis: Test procesu přihlášení do aplikace s rolí Uživatel.

Vazba na případ užití: UP001

Vstupní podmínky: Uživatel disponuje rolí Uživatel. Uživatel není přihlášen do aplikace.

Výstupní podmínky: Uživatel je přihlášen do aplikace

Priorita: High

	Uživatel:	System:
1.	Zadej přihlašovací jméno a heslo, pro roli Uživatel.	System zaktivní možnost potvrzení přihlášení.
2.	Zvol možnost pro vstup do aplikace.	System klienta přihlásí a zobrazí mu úvodní obrazovku.

TP002: Přihlášení uživatele s rolí Administrátor

Stručný popis: Test procesu přihlášení do aplikace s rolí Administrátor.

Vazba na případ užití: UP001

Vstupní podmínky: Uživatel disponuje rolí Administrátor. Uživatel není přihlášen do aplikace.

Výstupní podmínky: Uživatel je přihlášen do aplikace.

Priorita: High

	Uživatel:	System:
1.	Zadej přihlašovací jméno a heslo, pro roli Administrátor.	System zaktivní možnost potvrzení přihlášení.
2.	Zvol možnost pro vstup do aplikace.	System klienta přihlásí a zobrazí mu úvodní obrazovku.

TP003: Nesprávně zadané přihlašovací údaje

Stručný popis: Test možnosti, kdy uživatel zadá nesprávné přihlašovací údaje.

Vazba na případ užití: UP001

Vstupní podmínky: Uživatel není přihlášen do aplikace.

Výstupní podmínky: Uživatel nebyl do aplikace přihlášen, je mu zobrazena chybová hláška a má možnost zadat přihlašovací údaje znovu.

Priorita: Normal

	Uživatel:	System:
1.	Zadej neexistující přihlašovací jméno a heslo.	System zaktivní možnost potvrzení přihlášení.
2.	Zvol možnost pro vstup do aplikace.	V případě, že uživatel zadat špatné přihlašovací údaje, system na to uživatele upozorní. Klient má možnost zadat údaje znovu.

TP004: Zapomenuté heslo

Stručný popis: Test asistence v případě, kdy uživatel zapomněl své heslo.

Vazba na případ užití: UP001

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel není přihlášen do aplikace.

Výstupní podmínky: Uživatel obnovil své heslo.

Priorita: High

	Uživatel:	Systém:
1.	Zvol možnost pro resetování hesla v případě, kdy jej zapomněl.	Systém uživatele upozorní, že mu na mail poslal instrukce k resetování hesla.
2.	Postupuj dle pokynů v mailu.	Heslo bylo obnoveno.

TP005: Zobrazení plánu kanceláří

Stručný popis: Test zobrazení a změny plánu podlaží.

Vazba na případ užití: UP002

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Uživateli je zobrazen plán podlaží dle výsledku jeho vyhledávání.

Priorita: High

	Uživatel:	Systém:
1.	Zvol možnost pro zobrazení plánu kanceláří	Systém zobrazí plány kanceláří.
2.	Vyber možnost pro změnu zobrazeného podlaží.	Systém zobrazí podlaží, dle uživatelova výběru.

TP006: Vyhledávání v plánech kanceláří

Stručný popis: Test vyhledávání v plánech kanceláří skrze vyhledání patra, zasedačky či pracovního místa, dle jména zaměstnance.

Vazba na případ užití: PU003

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace. Uživatel provedl zobrazení plánu kanceláří.

Výstupní podmínky: Uživateli je zobrazen plán podlaží dle výsledku jeho vyhledávání.

Priorita: High

	Uživatel:	System:
1.	Zvol možnost pro vyhledání a zobrazení určitého podlaží.	System, dle uživatelova výběru, zobrazí možnosti zadání vyhledávacích kritérií.
2.	Zadej vyhledávací kritéria a vyhledání potvrď.	Aplikace zobrazí plán příslušného patra.
3.	Zvol možnost pro vyhledání určité zasedačky.	System, dle uživatelova výběru, zobrazí možnosti zadání vyhledávacích kritérií.
4.	Zadej vyhledávací kritéria a vyhledání potvrď.	Aplikace zobrazí plán příslušného patra. Vyhledaná zasedačka je graficky zvýrazněna.
5.	Zvol možnost pro vyhledání pracovního místa dle jména zaměstnance.	System, dle uživatelova výběru, zobrazí možnosti zadání vyhledávacích kritérií.
6.	Zadej vyhledávací kritéria a vyhledání potvrď.	Aplikace zobrazí plán příslušného patra. Vyhledané pracovní místo je graficky zvýrazněné.

TP007: Grafické schéma rezervací zasedaček

Stručný popis: Test zobrazení grafického schéma rezervací zasedaček pro určité datum a následné změny tohoto data.

Vazba na případ užití: UP004

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Uživateli je zobrazeno grafické schéma rezervací zasedaček pro datum dle uživatelova výběru.

Priorita: High

	Uživatel:	Systém:
1.	Zvol možnost pro zobrazení Grafického schématu.	Systém provede zobrazení Grafického schématu pro aktuální den.
2.	Vyber možnost pro změnu data.	Systém zobrazí Grafické schéma pro datum dle uživatelova výběru.

TP008: Filtr zobrazení v grafickém schématu rezervací zasedaček

Stručný popis: Test filtrování v grafickém schématu rezervací zasedaček skrze rezervace jedné zasedačky, všech zasedaček v určité budově a v podlaží.

Vazba na případ užití: UP005

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace. Existuje několik zasedacích místností v rámci patra a v rámci budovy, na kterých je vytvořeno několik rezervací. Uživatel má zobrazen přehled rezervací.

Výstupní podmínky: Systém provedl filtrování zobrazených zasedaček, dle všech možností.

Priorita: High

	Uživatel:	Systém:
1.	Zvol zobrazení všech zasedaček v určité budově.	Systém, dle uživatelova výběru, zobrazí možnosti zadání vyhledávacích kritérií.
2.	Uživatel zadá vyhledávací kritéria a vyhledání potvrdí.	Aplikace zobrazí grafický přehled obsazenosti zasedaček či zasedačky dle uživatelova výběru, defaultně pro aktuální den.
3.	Zvol zobrazení všech zasedaček v určitém podlaží.	Systém, dle uživatelova výběru, zobrazí možnosti zadání vyhledávacích kritérií.

4.	Uživatel zadá vyhledávací kritéria a vyhledání potvrdí.	Aplikace zobrazí grafický přehled obsazenosti zasedaček či zasedačky dle uživatelova výběru, defaultně pro aktuální den.
5.	Zvol zobrazení pouze jedné konkrétní zasedačky.	Systém, dle uživatelova výběru, zobrazí možnosti zadání vyhledávacích kritérií.
6.	Uživatel zadá vyhledávací kritéria a vyhledání potvrdí.	Aplikace zobrazí grafický přehled obsazenosti zasedačky dle uživatelova výběru, defaultně pro aktuální den.

TP009: Detail existující rezervace

Stručný popis: Test zobrazení detailu rezervace aktuálně přihlášeného uživatele.

Vazba na případ užití: UP006

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Uživateli je zobrazen detail požadované rezervace, obsahující všechny požadované atributy.

Priorita: High

	Uživatel:	Systém:
1.	Zvol možnost zobrazení detailu rezervace.	Systém zobrazí detail rezervace. V tomto detailu jsou uvedeny tyto informace: Jméno zasedačky, Organizátor, Důvod rezervace, Datum, Čas od a Čas do.

TP010: Nová rezervace zasedačky

Stručný popis: Test vytvoření nové rezervace zasedačky.

Vazba na případ užití: UP007

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Byla vytvořena nová rezervace.

Priorita: High

	Uživatel:	System:
1.	Vyber možnost pro vytvoření nové rezervace. Zadej Jméno zasedačky, Důvod rezervace, Datum, Čas od a Čas do. Zadej takový termín, aby nebyl doposud obsazen jinou rezervací. Vyber možnost pro uložení.	System provede validaci dat, která zadal uživatel. Pokud jsou data validní a termín je volný, system uloží novou rezervaci.

TP011: Nová rezervace zasedačky - neúspěšná validace

Stručný popis: Test možnosti neúspěšné validace, během vytváření nové rezervace zasedačky.

Vazba na případ užití: UP007

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Uživatel byl systémem upozorněn na nevalidní data ve vstupním poli.

Priorita: Low

	Uživatel:	System:
1.	Vyber možnost pro vytvoření nové rezervace. Zadej Jméno zasedačky, Důvod rezervace, Datum, Čas od a Čas do. Zadej takový termín, aby nebyl doposud obsazen jinou rezervací. Vyber možnost pro uložení.	System provede validaci dat, která zadal uživatel. Pokud jsou vstupní data nevalidní, nebo je termín již obsazen, system na to uživatele upozorní.

TP012: Přehled rezervací přihlášeného uživatele

Stručný popis: Test zobrazení přehledu rezervací, aktuálně přihlášeného uživatele.

Vazba na případ užití: UP008

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Systém zobrazuje přehled rezervací přihlášeného uživatele.

Priorita: High

	Uživatel:	Systém:
1.	Vyber možnost pro zobrazení aktuálních rezervací přihlášeného uživatele.	Systém uživateli zobrazí přehled jeho aktuálních rezervací.

TP013: Úprava rezervace přihlášeného uživatele

Stručný popis: Test úpravy rezervace, aktuálně přihlášeného uživatele. Lze upravit parametry Důvod rezervace, Čas od a Čas do.

Vazba na případ užití: UP009

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Proběhla úspěšná editace existující rezervace.

Priorita: High

	Uživatel:	Systém:
1.	Vyber možnost pro zobrazení aktuálních rezervací přihlášeného uživatele.	Systém uživateli zobrazí přehled jeho aktuálních rezervací.
2.	Vyber možnost pro editaci některé ze zobrazených rezervací.	Systém umožní uživateli provést editaci parametrů Důvod rezervace, Čas od a Čas do.

3.	Proveď změnu a vyber možnost pro potvrzení. Zadaná data jsou validní a nový termín zatím není obsazen.	Systém provede validaci dat, která zadal uživatel. Pokud jsou data validní a termín je volný, systém uloží novou rezervaci.
----	--	---

TP014: Úprava rezervace přihlášeného uživatele - Neúspěšná validace

Stručný popis: Test možnosti neúspěšné validace během úpravy rezervace, aktuálně přihlášeného uživatele.

Vazba na případ užití: UP009

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Uživatel byl systémem upozorněn na nevalidní data ve vstupním poli.

Priorita: Low

	Uživatel:	Systém:
1.	Vyber možnost pro zobrazení aktuálních rezervací přihlášeného uživatele.	Systém uživateli zobrazí přehled jeho aktuálních rezervací.
2.	Vyber možnost pro editaci některé ze zobrazených rezervací.	Systém umožní uživateli provést editaci parametrů Důvod rezervace, Čas od a Čas do.
3.	Proveď změnu a vyber možnost pro potvrzení. Zadaná data jsou nevalidní a nový termín je obsazen.	Systém provede validaci dat, která zadal uživatel. Pokud jsou data validní a termín je volný, systém uloží novou rezervaci.

TP015: Smazání rezervace přihlášeného uživatele

Stručný popis: Test smazání rezervace, aktuálně přihlášeného uživatele.

Vazba na případ užití: UP010

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Proběhlo smazání uživatelovy rezervace.

Priorita: High

	Uživatel:	Systém:
1.	Vyber možnost pro zobrazení aktuálních rezervací přihlášeného uživatele.	Systém uživateli zobrazí přehled jeho aktuálních rezervací.
2.	Vyber možnost pro smazání některé rezervace.	Systém se uživatele zeptá, zda si opravdu přeje tuto rezervaci smazat. Nabídne uživateli možnost pro potvrzení smazání rezervace a pro zrušení tohoto požadavku.
3.	Zvol možnost pro potvrzení smazání rezervace.	Pokud uživatel zvolil možnost pro potvrzení smazání rezervace, systém rezervaci smaže.

TP016: Smazání rezervace přihlášeného uživatele - zrušení smazání

Stručný popis: Test situace, kdy uživatel zrušil požadavek na smazání jeho rezervace.

Vazba na případ užití: UP010

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Uživatel zrušil požadavek na smazání rezervace, systém rezervaci nemesmazal.

Priorita: Normal

	Uživatel:	Systém:
1.	Vyber možnost pro zobrazení aktuálních rezervací přihlášeného uživatele.	Systém uživateli zobrazí přehled jeho aktuálních rezervací.
2.	Vyber možnost pro smazání některé rezervace.	Systém se uživatele zeptá, zda si opravdu přeje tuto rezervaci smazat. Nabídne uživateli možnost pro potvrzení smazání

		rezervace a pro zrušení tohoto požadavku.
3.	Zvol možnost pro zrušení tohoto požadavku.	Pokud uživatel zvolil možnost pro zrušení tohoto požadavku, systém rezervaci nesmaže.

TP017: Proklik z detailu vlastní rezervace na plán kanceláří

Stručný popis: Test prokliku z detailu vlastní rezervace na plán kanceláří

Vazba na případ užití: UP011

Vstupní podmínky: Uživatel disponuje rolí Uživatel nebo Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Proběhlo zobrazení plánu kanceláří, kde je zobrazeno patro, na kterém se nachází příslušná zasedačka, která je barevně zvýrazněna.

Priorita: High

	Uživatel:	Systém:
1.	Vyber možnost pro zobrazení aktuálních rezervací přihlášeného uživatele.	Systém uživateli zobrazí přehled jeho aktuálních rezervací.
2.	Zvol možnost pro zobrazení některé rezervované zasedačky na plánu kanceláří.	Systém uživateli zobrazí plány kanceláří, kde je zobrazeno patro, na kterém se nachází příslušná zasedačka. Ta je barevně zvýrazněna.

TP018: Zobrazení přehledu pracovních míst spolu se jmény zaměstnanců

Stručný popis: Test zobrazení přehledu všech pracovních míst spolu se jmény zaměstnanců

Vazba na případ užití: UP012

Vstupní podmínky: Uživatel disponuje rolí Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Systém uživateli zobrazil seznam pracovních míst spolu se jmény zaměstnanců, kteří na daných místech pracují.

Priorita: High

	Uživatel:	Systém:
1.	Vybere možnost pro zobrazení seznamu pracovních míst.	Systém zobrazí uživateli seznam pracovních míst. V rámci tohoto seznamu jsou uvedena i jména zaměstnanců, kteří na daných místech pracují.

TP019: Editace jmen zaměstnanců

Stručný popis: Test editace jmen zaměstnanců přiřazených k určitému pracovnímu místu.

Vazba na případ užití: UP013

Vstupní podmínky: Uživatel disponuje rolí Administrátor. Uživatel je přihlášen do aplikace. Uživatel má zobrazen přehled pracovních míst spolu se jmény zaměstnanců.

Výstupní podmínky: Proběhla změna jména zaměstnance, který je přiřazen k určitému pracovnímu místu.

Priorita: High

	Uživatel:	Systém:
1.	Vyber možnost pro editaci pracovního místa.	Systém umožní uživateli provést editaci parametrů Jméno a Příjmení.
2.	Uživatel provede změnu a vybere možnost pro potvrzení.	Systém uloží změnu.

TP020: Rušení existujících rezervací administrátorem

Stručný popis: Test zrušení existující rezervace, kterou vytvořil jiný uživatel.

Vazba na případ užití: UP014

Vstupní podmínky: Uživatel disponuje rolí Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Na základě uživatelova výběru proběhlo smazání rezervace vytvořené jiným uživatelem.

Priorita: High

	Uživatel:	System:
1.	Na detailu vyber možnost pro smazání příslušné rezervace.	System se uživatele zeptá, zda si opravdu přeje tuto rezervaci smazat. Nabídne uživateli možnost pro potvrzení smazání rezervace a pro zrušení tohoto požadavku.
2.	Zvol možnost pro potvrzení smazání rezervace.	Pokud uživatel zvolil možnost pro potvrzení smazání rezervace, system rezervaci smaže.

TP021: Rušení existujících rezervací administrátorem - zrušení požadavku

Stručný popis: Test případu, kdy uživatel zrušil požadavek, na zrušení existující rezervace, kterou vytvořil jiný uživatel.

Vazba na případ užití: UP014

Vstupní podmínky: Uživatel disponuje rolí Administrátor. Uživatel je přihlášen do aplikace.

Výstupní podmínky: Na základě zrušení uživatelova požadavku neproběhlo smazání rezervace vytvořené jiným uživatelem.

Priorita: Normal

	Uživatel:	System:
1.	Na detailu vyber možnost pro smazání příslušné rezervace.	System se uživatele zeptá, zda si opravdu přeje tuto rezervaci smazat. Nabídne uživateli možnost pro potvrzení smazání rezervace a pro zrušení tohoto požadavku.
2.	Zvol možnost pro zrušení tohoto požadavku.	Pokud uživatel zvolil možnost pro zrušení tohoto požadavku, system smazání rezervace neprovede.