

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Návrh a implementace aplikace pro podporu rezervace místností

Jan Vostrý

Vedoucí: Ing. Pavel Náplava, Ph.D.
Obor: Softwarové inženýrství a technologie
Květen 2019

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vostrý** Jméno: **Jan** Osobní číslo: **469921**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Návrh a implementace aplikace pro podporu rezervace místností

Název bakalářské práce anglicky:

Design and implementation of rooms reservation system

Pokyny pro vypracování:

Navrhněte a vytvořte aplikaci pro rezervaci místností, která bude využívána jako doplňkový výukový materiál v předmětu „Informační systémy“. Aplikaci vytvořte na platformě IBM cloud a doplňte ji podpurnými výukovými materiály, demonstrujícími postup při tvorbě aplikace. Postupujte následujícím způsobem:

1. Analyzujte aktuálně využívané výukové materiály pro předmět „Informační systémy“.
2. Navrhněte aplikaci, která naváže na existující případovou studii a popište ji pomocí standardních technik, využívaných v softwarovém inženýrství. Pozornost věnujte především využití wireframů a prototypů.
3. Seznamte se s prostředím IBM Cloud a následně v něm vytvořenou aplikaci implementujte.
4. Vytvořenou aplikaci společně s vyučujícím předmětu uživatelsky otestujte a doplňte výukovou dokumentací, demonstrující přechod od případové studie přes prototyp k výsledné aplikaci.

Seznam doporučené literatury:

- [1] Arlow, J., Neustat, I.: UML 2 a unifikovaný proces vývoje aplikací. Computer Press, ISBN: 978-80-251-1503-9, Praha 2007. (Též jako E-kniha.)
- [2] FOWLER, Martin. Destilované UML. 1. vyd. Praha: Grada, 2009, 173 s. Knihovna programátora (Grada). ISBN 978-80-247-2062-3.
- [3] KNAPP, Jake, John Zeratsky, Braden Kowitz. SPRINT - Jak vyřešit velké problémy a otestovat nové myšlenky v pouhých pěti dnech. Jan Melvil Publishing. 2017. ISBN: 978-80-7555-023-1.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Pavel Náplava, Ph.D., katedra ekonomiky, manažerství a humanitních věd FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **04.02.2019**

Termín odevzdání bakalářské práce: **24.05.2019**

Platnost zadání bakalářské práce: **20.09.2020**

Ing. Pavel Náplava, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Ing. Pavel Ripka, CSc.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

Chtěl bych poděkovat panu Ing. Pavlu Náplavovi Ph.D., za velmi ochotné vedení mé práce, poznatky, tipy, zkušenosti, určení směru a kritiku, díky kterým jsem tuto práci vypracoval.

Dále bych chtěl poděkovat své rodině a přítelkyni Michaelle Klicperové za psychickou podporu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 23. května 2019

Abstrakt

Bakalářská práce se zabývá analýzou výukových materiálů předmětu „Informační systémy“, návrhem, implementací aplikace pro rezervaci místností a nasazením aplikace do IBM Cloud. Popisuje prototypování, prototypovací nástroje, prostředí IBM Cloud a práci s ním. Ukazuje postup, jak od požadavků dojít přes návrh, prototyp, implementaci a testování, až k nasazení aplikace do provozu. Vysvětluje jednotlivé rozhodnutí a kroky při návrhu, implementaci a nasazení aplikace. Výsledkem této práce je rozšíření výukových materiálů předmětů „Informační systémy“ a dle návrhu implementovaná aplikace.

Klíčová slova: rezervační systém, případová studie, analýza, návrh systému, implementace rezervačního systému, wireframe, prototyp, cloud, IBM cloud, nasazení

Vedoucí: Ing. Pavel Náplava, Ph.D.

Abstract

Bachelor thesis deals with analyse of the teaching materials for subject „Informační systémy“, design and implementation of room reservation system and application deploy to IBM Cloud. It describes prototyping, prototyping tools and IBM Cloud environment. It shows process how to get from requirements over design, prototype, implementation and testing to deploying the application to cloud. It explains decisions and steps in design, implementation and application deploy. The results of this thesis are extension of teaching material for subject „Informační Systémy“ and implemented room reservation system based on described design.

Keywords: reservation system, case study, analysis, system design, reservation system implementation, wireframe, prototype, cloud, IBM cloud, deploy

Title translation: Design and implementation of room reservation system

Obsah

1 Úvod	1		
1.1 Motivace	1		
1.2 Cíle	1		
1.3 Struktura	2		
2 Analýza předmětu B6B16INS	3		
2.1 Náplň předmětu	3		
2.2 Práce v předmětu	3		
2.3 Problémy v předmětu	4		
2.4 Požadavky plynoucí z problémů	4		
2.5 Shrnutí analýzy předmětu	5		
3 Rešerše prototypovacích nástrojů	7		
3.1 Prototypování a prototyp	7		
3.1.1 Popis prototypování	7		
3.1.2 Účel prototypování	7		
3.1.3 Výhody prototypování	8		
3.1.4 Nevýhody prototypování	8		
3.1.5 Proces prototypování	8		
3.1.6 Dimenze prototypování	9		
3.1.7 Typy prototypování	9		
3.1.8 Prototyp vs. Mockup vs. Wireframe	9		
3.1.9 Doporučení pro prototypování	11		
3.2 Prototypovací nástroje	11		
3.2.1 Popis prototypovacích nástrojů	11		
3.2.2 Metodika porovnání prototypovacích nástrojů	11		
3.2.3 Volba prototypovacích nástrojů	12		
3.2.4 Porovnání prototypovacích nástrojů	13		
3.2.5 Zhodnocení prototypovacích nástrojů	16		
3.3 Popis práce s prototypovacími nástroji	16		
3.3.1 Popis nástroje: Balsamiq Mockups	16		
3.3.2 Popis nástroje: AdobeXD CC 2018	19		
3.4 Shrnutí rešerše	22		
4 Analýza prostředí IBM Cloud	23		
4.1 Cloud obecně	23		
4.1.1 Rozdělení cloudů podle distribuce	23		
4.1.2 Výhody využití cloudů	24		
4.1.3 Nevýhody využití cloudů	24		
4.1.4 Budoucnost cloudů	24		
4.2 Prostředí IBM Cloud	25		

4.2.1	Popis IBM Cloud	25	5.2.4	Diagram tříd aplikace	40
4.2.2	Možnosti hostování aplikace v IBM Cloud	26	5.3	Prototyp aplikace	41
4.2.3	Prostředí Cloud Foundry v IBM Cloud	27	5.4	Shrnutí analýzy aplikace	44
4.2.4	Postup aktivace studentského účtu IBM Cloud	27	6 Implementace aplikace	45	
4.2.5	Nástroje pro práci s prostředím IBM Cloud	28	6.1	Použité technologie	45
4.3	Shrnutí analýzy prostředí IBM Cloud	29	6.2	Popis technologií	45
5 Analýza a návrh aplikace pro rezervaci místností	31		6.3	Architektura aplikace	46
5.1	Požadavky aplikace	31	6.4	Implementované požadavky	47
5.1.1	Požadavky na funkčnost	32	6.5	Shrnutí implementace	47
5.1.2	Požadavky na použitelnost	33	7 Uživatelské testování aplikace	49	
5.1.3	Požadavky na spolehlivost	34	7.1	Popis testování	49
5.1.4	Požadavky na výkon	34	7.1.1	Dotazník	50
5.1.5	Požadavky na podporovatelnost	34	7.2	Testování aplikace	50
5.2	Diagramy aplikace	35	7.2.1	Účastník 1	50
5.2.1	Diagram případů užití aplikace	35	7.2.2	Účastník 2	51
5.2.2	Procesní diagramy aplikace	37	7.2.3	Účastník 3	51
5.2.3	Stavový diagramy aplikace	39	7.2.4	Účastník 4	51
			7.2.5	Účastník 5	51
			7.2.6	Účastník 6	51
			7.3	Vyhodnocení výsledků testování	52
			8 Nasazení aplikace do prostředí IBM Cloud	53	
			8.1	Možnosti nasazení	53
			8.2	Prerekvizity nasazení	54

8.3	Nasazení do cloudu	55
8.3.1	Struktura postupu nasazení	55
8.3.2	Vytvoření Cloud Foundry aplikace	55
8.3.3	Vytvoření Continuous Delivery Toolchain	56
8.3.4	Přidání NoSQL databáze	56
8.3.5	Příprava prostředí pro vývoj	57
8.3.6	Nastavení Delivery Pipeline	57
8.4	Shrnutí práce s prostředím	58
9	Závěr	61
A	Literatura	63
B	Aplikace: Rezervační systém	67
B.1	Prerekvizity práce s aplikací	67
B.2	Instalace a příprava prostředí	67
B.3	Snímky aplikace	69

Obrázky

3.1	Postup návrhu a tvorby prototypu [17]	10	5.2	Procesní diagram celé aplikace	37
3.2	Wireframe vs. Mockup [17]	10	5.3	Proces: Vytvoření místnosti	38
3.3	Nástroj Adobe XD CC 2018	13	5.4	Proces: Vytvoření rezervace	38
3.4	Nástroj InVision	14	5.5	Proces: Vytvoření rezervace	38
3.5	Nástroj Balsamiq Mockups	15	5.6	Stavový diagram: rezervace	39
3.6	Nástroj Proto.io	16	5.7	Diagram tříd	40
3.7	Balsamiq Mockups: horní panel	17	5.8	Prototyp: Přihlášení	41
3.8	Balsamiq Mockups: panel prvků	17	5.9	Prototyp: Moje rezervace	42
3.9	Balsamiq Mockups: hlavní obrazovka	18	5.10	Prototyp: Vyhledání rezervace	42
3.10	Balsamiq Mockups: panel prvků	18	5.11	Prototyp: Vytvoření rezervace	43
3.11	Balsamiq Mockups: pravý panel	19	5.12	Prototyp: Rezervace vytvořena	43
3.12	Adobe XD: horní lišta	19	5.13	Prototyp: Rezervace potvrzena	44
3.13	Adobe XD: levá lišta	20	5.14	Prototyp: Přehled místností	44
3.14	Adobe XD: levý panel	20	6.1	Model-View-Controller	46
3.15	Adobe XD: hlavní obrazovka	21	8.1	Tvorba aplikace v IBM Cloud [10]	54
3.16	Adobe XD: pravý panel	21	B.1	Aplikace: Tomcat Server	68
3.17	Adobe XD: prototyp	22	B.2	Aplikace: Tomcat Deployment	69
4.1	Trh cloudových služeb [8]	25	B.3	Aplikace: Přihlášení	69
4.2	Přístupový kód IBM Cloud	27	B.4	Aplikace: Registrace	70
5.1	Use-case diagram	36	B.5	Aplikace: Hlavní stránka	70
			B.6	Aplikace: Moje rezervace	71

B.7 Aplikace: Místnosti	71
B.8 Aplikace: Místnost	72
B.9 Aplikace: Rezervace	72

Kapitola 1

Úvod

1.1 Motivace

Práce vychází z předmětu B6B16INS. Tento předmět seznamuje studenty se všemi aspekty výběru, nasazování a údržby informačních systémů. V rámci předmětu studenti tvoří případovou studii, podle které následně tvoří prototyp. Dle prototypu implementují aplikaci, kterou mají nasadit v prostředí IBM Cloud. Momentálně není dostatek materiálů, které popisují prototypování, tvorbu prototypu, nasazení aplikace a práci s prostředím IBM Cloud. A díky těmto nedostatkům vznikla tato práce.

Tuto práci jsem si zvolil, protože jsem předmět B6B16INS absolvoval. Díky nedostatečným materiálům pro výuku jsem s kolegy v předmětu čelil řadě problémů a v rámci této práce bych chtěl tyto problémy rozebrat a vyřešit. Zároveň mi přijde zajímavé prostředí cloudů, které je v dnešní době stále populárnější.

1.2 Cíle

Cílem této práce je rozšíření existujících materiálů předmětu B6B16INS. Tyto výukové materiály se budou zabývat vytvořením přehledu prototypovacích nástrojů, popisem práce s prototypovacími nástroji, popisem prostředí IBM Cloud a nasazením aplikace do tohoto cloudu. Tyto materiály budou tvořeny v kontextu vytvoření aplikace pro rezervaci místností.

■ 1.3 Struktura

Struktura práce je následující:

- Analýza předmětu B6B16INS.
- Vytvoření přehledu a porovnání existujících prototypovacích nástrojů. Popis práce s prototypovacími nástroji.
- Sběr požadavků, tvorba diagramů, návrh aplikace pro rezervaci místností. Tvorba prototypu.
- Implementace aplikace pro rezervaci místností.
- Uživatelské testování aplikace pro rezervaci.
- Analýza prostředí IBM Cloud a ukázka nasazení aplikace do cloudu.

Ve struktuře této práce se odráží postup práce studentů v předmětu B6B16INS.

Kapitola 2

Analýza předmětu B6B16INS

Tato kapitola slouží pro seznámení se s předmětem B616INS, popíšeme si, jaká je náplň předmětu, a co v rámci předmětu studenti musí splnit. Blíže se seznámíme s problémy, které v předmětu momentálně jsou a vyvodíme z nich důsledky.

2.1 Náplň předmětu

Náplň předmětu je seznámit studenty se všemi aspekty výběru, nasazování a údržby informačních systémů. Důraz je zde kladen na schopnost orientovat se v problematice na různých úrovních (tým, zákazník, management) a v různých podobách (diskuze, analýza, prezentace), práci v týmu a na základě vyhodnocení potřeb zákazníka navržení vhodného řešení.

2.2 Práce v předmětu

Na začátku semestru jsou studenti rozděleni do týmů. V první polovině semestru se zákazníkem konzultují problémy, kterým zákazník čelí. Na základě těchto problémů sepisují požadavky, podle kterých pak navrhuji optimální řešení, které popisují v případové studii. Po vytvoření studie, v druhé polovině semestru, začínají tvorbou prototypu, kde se po dvou cyklech konzultace (zhruba tři týdny) se zákazníkem domluví na podobě systému. Funkční část systému mají následně implementovat a nasadit do prostředí IBM Cloud.

Prostředí IBM Cloud se v předmětu využívá díky spolupráci ČVUT a společnosti IBM, která tento cloud provozuje. Myslím si, že mít možnost pracovat s tímto prostředím je příležitost, o kterou by žádný student neměl přijít. Ovšem momentálně je tato část nasazení aplikace do cloudu z předmětu vynechána,

protože studenti neměli dostatek času pro naučení se s tímto prostředím pracovat.

Zde jsou uvedeny kroky, kterými si projde každý tým studentů při absolvování předmětu:

1. Sběr požadavků.
2. Návrh systému.
3. Tvorba prototypu.
4. Implementace systému.
5. Nasazení systému.

2.3 Problémy v předmětu

Tyto problémy vychází z práce v předmětu, kde prvním hlavním problémem, kterému studenti čelí, je volba prototypovacího nástroje. V předmětu studenti potřebují, alespoň 15 denní zkušební verzi nebo možnost jednoho prototypu zdarma, aby mohli prototyp průběžně zdokonalovat. Druhý problém pak je nasazení systému do IBM Cloud. Prostředí neposkytuje dostatečnou dokumentaci, jak se s prostředím naučit rychle pracovat, a proto je pro studenty obtížné aplikaci do něho nasadit a tato část je vynechána.

2.4 Požadavky plynoucí z problémů

Požadavky, které plynou z problémů:

- Vytvoření přehledu prototypovacích nástrojů.
 - Nástroj využitelný alespoň na 2 týdny.
 - Nástroj musí poskytovat „free-trial“ verzi.
 - Nástroj by měl být dostupný pro většinu operačních systémů.
- Vytvoření manuálu pro práci s prototypovací nástroji.
- Vytvoření dokumentace prostředí IBM Cloud.
- Vytvoření manuálu pro nasazení aplikace do IBM Cloud.

■ 2.5 Shrnutí analýzy předmětu

V rámci této kapitoly jsme si popsali náplň předmětu B616INS a práci v semestru. Z toho jsme odvodili problémy, se kterými se studenti mohou setkat, a na základě těchto problémů vyplynuly požadavky. Těmito požadavky se v rámci této práce budeme zabývat a budeme se snažit je splnit.

Kapitola 3

Rešerše prototypovacích nástrojů

Tato kapitola kapitola navazuje na požadavky předmětu B6B16INS (viz. kap. 2). Popisuje softwarové prototypování a existující prototypovací nástroje. Tyto nástroje porovnáme a vytvoříme přehled o možnostech a použitelnosti nástrojů v předmětu. Z těchto nástrojů si pak student bude schopen vybrat nástroj, který vyhovuje potřebám předmětu. Nakonec si popíšeme jak se s vybranými prototypovacími nástroji pracuje.

3.1 Prototypování a prototyp

V této sekci si obecně řekneme, co to prototyp a prototypování znamená. Vysvětlíme si za jakým účelem tvoříme prototyp, a jaké nám z toho vyplývají výhody a nevýhody. Na závěr si řekneme, jak v předmětu B6B16INS tyto znalosti využijeme.

3.1.1 Popis prototypování

Prototypování je vytvoření jednotlivých prototypů aplikace. Tato činnost je součástí procesu vývoje softwaru. Prototyp většinou simuluje některé části výsledné aplikace nebo produktu, a proto je na aplikaci nezávislý. Na základě prototypů se aplikace může designově a funkcionálně vyvíjet dle přání zákazníka [1].

3.1.2 Účel prototypování

Dřívější vývoj aplikací bez použití prototypování, kdy byl vyvinut kompletní program a až následně se řešily nalezené odchylky mezi návrhem a imple-

mentací, vedl ke zvýšení nákladů a špatným odhadům času a ceny. Účel prototypu je proto vytvoření grafického návrhu aplikace, ale i modelu základních funkcí aplikace. Umožníme zhodnotit tyto návrhy skutečným používáním namísto pouhého popisování. Koncoví uživatelé následně také mohou přispět svými požadavky, čímž dosáhneme lepších výsledků po vytvoření aplikace [6].

■ 3.1.3 Výhody prototypování

Snížení nákladů a času: Jednoduchým, vystihujícím prototypem dokážeme popsat budoucí stav aplikace, který nám napomůže při tvorbě samotné. Můžeme si tak ušetřit drahocenný čas, který bychom vynaložili na změnu aplikace (změny v prototypu « změny v aplikaci). Zlepšíme tak i kvalitu požadavků, ze kterých budou vývojáři stavět [1].

Zapojení uživatele: Prototypování může zapojovat i uživatele, který si může prototyp otestovat a podat zpětný názor. Tento názor nejlépe popisuje požadavky systému, protože chceme, aby náš zákazník byl spokojený a klient chce, aby jeho uživatelé byli spokojeni. Výsledná aplikace pak vyhovuje představám našich uživatelů [1].

■ 3.1.4 Nevýhody prototypování

Nedostatečná analýza: Zaměření se na prototyp může vést vývojáře od náležité analýzy kompletního projektu. To může vést k přehlédnutí lepších řešení aplikace nebo tvorbě neúplných popisů. Může také nastat, že se vývojáři snaží z prototypu postavit výslednou aplikaci [1].

Nadměrný čas vývoje prototypu: Vývojáři mohou zapomenout na fakt, že prototyp by měl být rychle vytvořený a dostatečně vystihující danou oblast či celek. V případě, že se tak nestane, bude nadbytečný čas strávený na prototypu k ničemu, protože požadavky klienta, po vyzkoušení prototypu, se mohou změnit a celý prototyp se může zahodit [1].

■ 3.1.5 Proces prototypování

Proces prototypování je souhrn kroků k vytvoření prototypu dle požadavků zákazníka a koncového uživatele.

1. Identifikace základních požadavků: Zjištění základních požadavků klienta na funkčnost a vzhled, zjištění vstupních a výstupních informací aplikace.

2. Vývoj prvního prototypu: Vyvinutí prvního prototypu na základě požadavků z předchozího kroku, který bude zaměřený na uživatelské rozhraní

tzn. tvorba Sketchů a Wireframů.

3. Zpětná vazba: Zákazník nám poskytne zpětnou vazbu na náš prvotní návrh.

Poznámka: Předchozí dva kroky se mohou opakovat a tím dosáhneme lepší specifikace prototypu.

4. Úpravy a vylepšení: Na základě zpětné vazby vytvoříme tzv. Mockup nebo Prototyp, který ukážeme zákazníkovi a následně můžeme poskytnout i koncovým uživatelům.

■ 3.1.6 Dimenze prototypování

Prototypování můžeme dělit na základě dimenze. Rozdělujeme zde: horizontální prototyp a vertikální prototyp.

Horizontální prototyp: Takovýto prototyp se zabývá spíše grafickým uživatelským rozhraním. Tento prototyp je klikací a neukazuje na funkcionality aplikace [16].

Vertikální prototyp: Takovýto prototyp je propracování určité části jednoho funkčního bloku aplikace, který poskytuje většinu funkcí tohoto subsystému [16].

■ 3.1.7 Typy prototypování

Prototypování dělíme na 4 hlavní typy: zahazovací prototypování, evoluční prototypování, inkrementální prototypování a extrémní prototypování [7] [6].

Zahazovací prototypování: vytvoříme prototyp, který bude následně zahozen za účelem sběru požadavků od zákazníka.

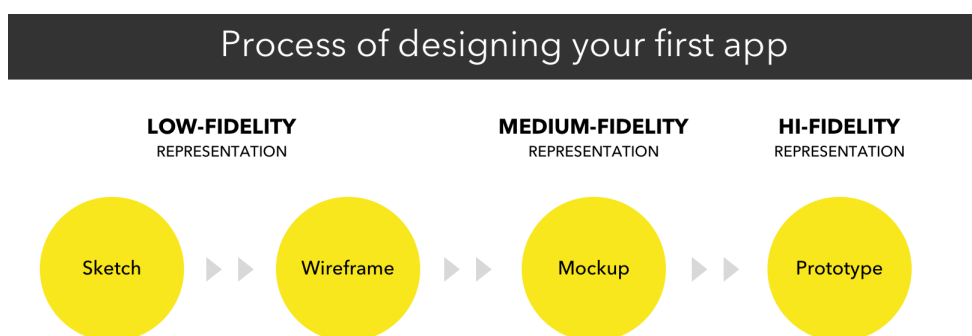
Evoluční prototypování: tvorba robustního modelu, který poslouží jako jádro budoucí aplikace.

Inkrementální prototypování: vytvoření dílčích prototypů jednotlivých částí, které jsou na závěr sloučeny do společného řešení.

Extrémní prototypování: zacílení na vývoj plně funkčního uživatelského rozhraní s minimálním ohledem na jiné služby.

■ 3.1.8 Prototyp vs. Mockup vs. Wireframe

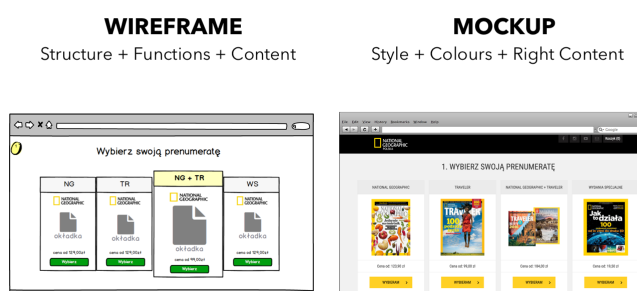
Tyto pojmy jsou často zaměňovány, někdo je dokonce vůbec nerozlišuje. I přes to jsou mezi nimi zásadní rozdíly (viz. obr. 3.1) [16] [18]. Před těmito návrhy by měly být vytvořeny tzv. **Sketch**, neboli prvotní návrh jednotlivých obrazovek nakreslením na papír.



Obrázek 3.1: Postup návrhu a tvorby prototypu [17]

Wireframe: Wireframe je statický model. Slouží k popisu jedné či více vizuálních reprezentací stránek. Můžeme zde využít přechodů mezi stránkami. Umístit na obrazovku jednotlivé prvky, na které když klikneme, tak se provede akce. Wireframe je většinou černobílý a nijak graficky nezpracovaný.

Mockup: Jak můžeme vidět na (obr. 3.2), tak Mockup je více propracovaný wireframe. Mockupy jsou více živé, dávají apel na vzhled, ukazují barvy, fonty, tlačítka, loga zákaznickovy firmy a další vizuální komponenty. Tím dosáhneme přesnějšího určení vzhledu stránky a můžeme tak lépe porozumět požadavkům zákazníka na vzhled.



Obrázek 3.2: Wireframe vs. Mockup [17]

Prototyp: Prototyp by již měl být celkem, který spojuje strukturu, funkčnost a návrh Wireframů se stylem a designem Mockupů. Poskytuje tak interaktivní prvky, přechody mezi stránkami a modeluje do jisté míry funkcionality systému. Prototyp nerovná se aplikaci!

■ 3.1.9 Doporučení pro prototypování

Na závěr této sekce bych doporučil při tvorbě prototypu v předmětu B6B16INS (viz. kap. 2) nejprve vytvořit sketche. Po vytvoření sketchů následně prototypovacím nástrojem navrhnout wireframy aplikace. Tyto wireframy prokonzultovat a nakonec z těchto wireframů vytvořit mockup či prototyp.

■ 3.2 Prototypovací nástroje

V této sekci si popíšeme prototypovací nástroje. Tyto nástroje vychází z požadavků předmětu B6B16INS (viz. kap. 2). Vytvoříme porovnání těchto nástrojů a zhodnotíme pro využití v předmětu.

■ 3.2.1 Popis prototypovacích nástrojů

Prototypovací nástroj je nástroj, který slouží k vytvoření prototypu, a proto by měl obsahovat náležité prvky a funkce, které jsou pro prototyp nezbytné. Prototypovacími nástroji zde rozumíme všechny programy na tvorbu jak prototypů, tak wireframů i mockupů, tyto nástroje se pak liší jen v preciznosti provedení prototypu.

Poznámka: Prototypovacím nástrojem lze nazvat jak základní prezentační software např. Power Point, tak i programovací jazyky, které mají grafickou nadstavbu např. Java nebo C#. Tyto aplikace v semestrálním projektu nezahrnuji.

■ 3.2.2 Metodika porovnání prototypovacích nástrojů

Tyto nástroje budeme porovnávat na základě požadavků předmětu (viz. kap. 2). Aktuální nabízené možnosti prototypovacích nástrojů jsou buď k dispozici jako webové nástroje nebo desktopové. V tomto případě se nijak nebudeme omezovat. Budeme se soustředit na ty nástroje, které poskytují „free-trial“ neboli bezplatné používání po určitou dobu, nebo ty které umožní vytvoření alespoň jednoho prototypu.

Většina prototypovacích nástrojů má podobnou funkcionalitu, a proto budu v jejich popisu uvádět pouze odlišnosti nebo přednosti jednotlivých nástrojů.

Hodnocení nástroje pak bude mít následující strukturu:

1. Základní informace o nástroji:
 - Název
 - Platforma
 - Poskytovatel
 - Webové stránky
 - Zkušební doba
2. Popis nástroje:
 - Popis funkcí a vlastností nástroje.
3. Zhodnocení nástroje:
 - Vlastní zhodnocení nástroje na základě zkušeností a práce s ním.

Poznámka: Vlastní zhodnocení se odvíjí ze zkušeností s prototypováním v nástroji a absolvováním předmětu B6B16INS.

3.2.3 Volba prototypovacích nástrojů

Momentálně existuje mnoho prototypovacích nástrojů, uvádím zde ty, ze kterých jsem vybíral: Lucidchart, InVision, Moqups, Wireframe.cc, MockFlow, Fluid UI, Pencil, Sketch, Mockplus, Proto.io, Adobe XD CC 2018, Axure, Origami Studio, Webflow, Framer, Vectr, UXPin, Balsamiq Mockups, Pidoco, Visio, InDesign CC, Marvel. Tyto nástroje jsem hledal a popisoval ze zdrojů [19] [20] [21].

Na základě omezení a vlastností těchto nástrojů jsem následně vybral tyto nástroje:

- Adobe XD CC
- InVision
- Balsamiq Mockup
- Proto.io

Omezením je myšlena již zmíněná doba zkušební verze, nebo možnost vytvoření nedostatečného množství obrazovek. Vlastnostmi se rozumí platformní omezení (př. využití nástroje pouze na MacOS) a systémové požadavky pro desktopové aplikace, tzn. že pro slabší počítače se starým hardwarem by tato aplikace nemusela pracovat. Tyto omezení a vlastnosti jsou spojené s tvorbou prototypu v předmětu B6B16INS (viz. kap. 2). Ovšem tyto čtyři nástroje jsou dostatečné pro to, aby si každý ze studentů mohl vybrat ten nejvhodnější.

3.2.4 Porovnání prototypovacích nástrojů

Adobe XD CC

1. Základní informace o nástroji:

Název: Adobe XD CC 2018

Platforma(desktop): MacOS, Windows

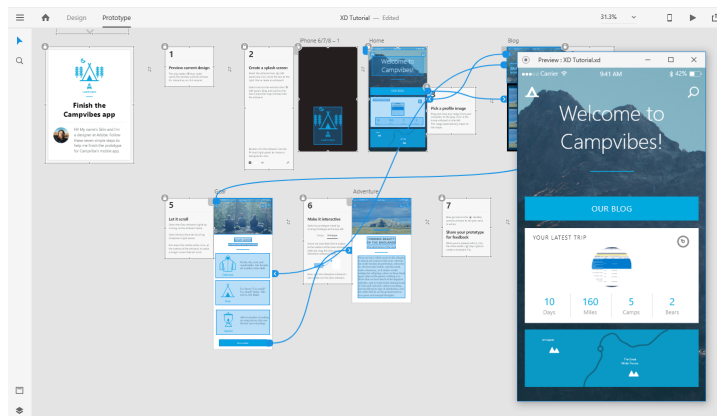
Poskytovatel: Adobe (Adobe Creative Cloud)

Webové stránky: <https://www.adobe.com/cz/products/xd.html>

Zdarma verze: 1 aktivně sdílený prototyp, 1 aktivně sdílený design, 2 GB Cloud úložiště, omezená knihovna fontů

2. Popis nástroje:

Nástroj poskytuje funkcionality jako jsou přechody mezi stránkami tzv. „artboardy“ pomocí pár kliknutí. Umožňuje správu všech artboardů z jednoho okna. Jsou zde možnosti využití importu PhotoShopu a dalších aplikací od Adobe. Poskytuje aplikaci pro mobilní telefony na iOS a Android, tak můžeme na svém mobilu rovnou práci vidět. Umožňuje propojení s aplikacemi jako je Slack a Jira. Na vylepšení tohoto nástroje se aktivně podílejí jeho uživatelé, kteří jsou za to finančně odměňováni.



Obrázek 3.3: Nástroj Adobe XD CC 2018

3. Zhodnocení nástroje:

Nástroj (obr. 3.3) využívá jednoduchého, ale efektivního designu. Umožňuje spoustu funkcí a je jednoduše ovladatelný. Poskytuje úvodní tutoriál, kde je v sedmi krocích vysvětlena práce na mobilním prototypu. Nástroj je dobře optimalizovaný a poskytuje mnoho pokročilých funkcí. Hodí se spíše pro medium-fidety a hi-fidety prototypy.

InVision

1. Základní informace o nástroji:

Název: InVision

Platforma(cloud): nezávislé

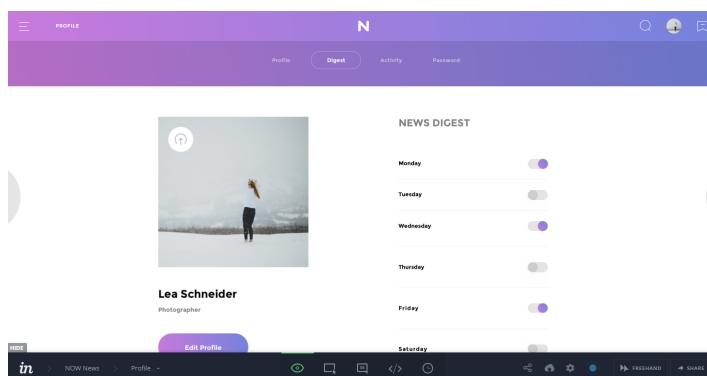
Poskytovatel: InVision

Webové stránky: <https://www.invisionapp.com>

Zdarma verze: jeden prototyp s neomezeným množstvím obrazovek a spolupráce po neomezenou dobu

2. Popis nástroje:

Nástroj poskytuje podaplikace pro malování sketchů, tvorby boardů a umožňuje i následný export sketche, který z něho vytvoří prototyp. Vytvoření prototypu umožňuje pro jakákoliv zařízení, kde poskytuje interaktivní tvorbu těchto prototypů.



Obrázek 3.4: Nástroj InVision

3. Zhodnocení nástroje:

Nástroj (obr. 3.4) mě oslovil svým designem a možnostmi vytváření sktechů, ze kterých je následně možné vytvořit prototyp. Poskytuje jednoduchý přechod ze statického designu do plně animovaného a zároveň umožňuje využití šablon pro tvorbu prototypů (webové aplikace, mobilní aplikace, apod.). Nástroj se hodí spíše pro hi-fidety prototypy.

Balsamiq Mockups

1. Základní informace o nástroji:

Název: Balsamiq Mockups

Platforma(cloud/desktop): nezávislé/MacOS, Windows

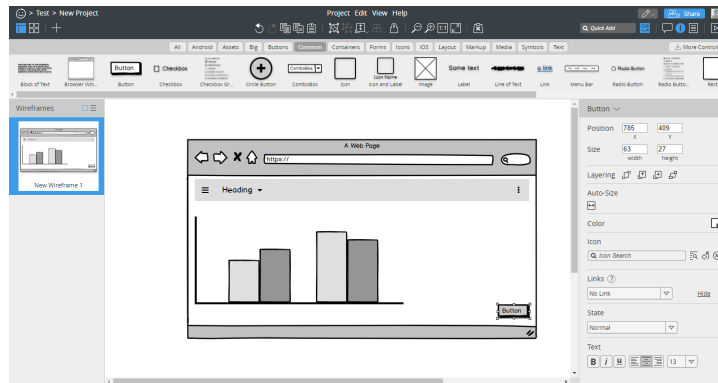
Poskytovatel: Balsamiq

Webové stránky: <https://balsamiq.com>

Zdarma verze: 30denní zkušební doba pro obě platformy

2. Popis nástroje:

Rychlý wireframovací nástroj, který se spíše soustředí na funkcionalitu a získávání zpětné vazby. Design aplikace je druhořadý a není nijak bráný v ohled. Umožňuje přechody mezi jednotlivými obrazovkami, nabízí širokou škálu předem připravených prvků, které pouze zakomponujeme do jednotlivých wireframů.



Obrázek 3.5: Nástroj Balsamiq Mockups

3. Zhodnocení nástroje:

Nástroj (obr. 3.5) je jednoduchý, rychle použitelný a poskytuje dostatečné množství prvků pro vytvoření méně precizního prototypu. Takový prototyp se pak skládá pouze z wireframů a přechodů mezi nimi. Pokud nelpíme na designu, ale na demonstraci funkcionalit, pak tento program bude nejlepší volba.

Proto.io

1. Základní informace o nástroji:

Název: Proto.io

Platforma(cloud): nezávislé

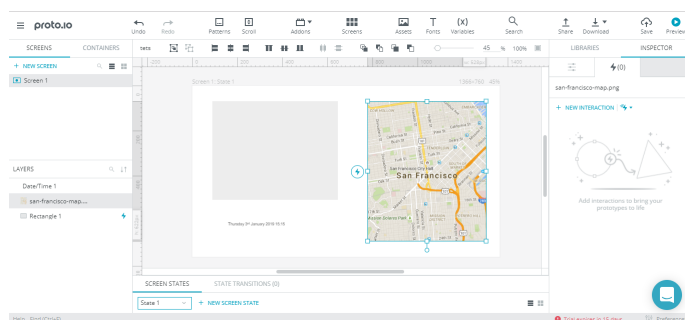
Poskytovatel:Proto.io

Webové stránky: <https://proto.io/>

Zdarma verze: 15 dní, plná verze

2. Popis nástroje:

Jednoduchý a zároveň komplexní prototypovací nástroj, obsahující 6 minutový tutorial, kde je vše potřebné vysvětlo. Platforma, na kterém je postaven prototypovací nástroj, se pokouší pro podporu a správu projektového managementu, aby designeři, developeři i manažeři pracovali na jedné platformě.



Obrázek 3.6: Nástroj Proto.io

3. Zhodnocení nástroje:

Nástroj (obr. 3.6) se hodí pro low-fi a medium-fi prototypy. Umožňuje jednoduché přechody pomocí drag-and-drop. Je zde možné si svůj prototyp i testovat. Obsahuje předpřipravené šablony, které můžeme ve svém prototypu využít. Na tomto nástroji mě zaujalo, že je zakomponovaný v platformě, kde můžete vázat požadavky přímo s prototypem.

3.2.5 Shrnutí prototypovacích nástrojů

Tyto čtyři nástroje byly vybrány na základě požadavků předmětu (viz. kap. 2). Každému, kdo bude vytvářet prototyp bych tak doporučil následující postup. V první řadě si udělat představu a nakreslit návrhy na papír (sketche). Za druhé použít low-fi nástroj a vytvořit wireframy např.: Balsamiq Mockups či Proto.io. Po dostatečném znázornění funkcionalit, využít medium-fi nebo high-fi prototypovací nástroj a vytvořit prototyp s designem např.: Adobe XD CC či. InVision.

3.3 Popis práce s prototypovacími nástroji

V této sekci si popíšeme práci se zvolenými prototypovacími nástroji. Z nástrojů, které vyhovují požadavkům předmětu, jsem se rozhodl pro popsání jednoho, který tvoří low-fi prototyp, a druhého, který tvoří medium-fi a hi-fi a to jsou: Balsamiq Mockups a AdobeXD CC 2018.

3.3.1 Popis nástroje: Balsamiq Mockups

Tento nástroj jsem zvolil kvůli předchozím zkušenostem s prototypováním, kdy jsem tento nástroj používal pro tvorbu prototypu v předmětu B6B16INS. Tento nástroj zde využíváme ve webovém rozhraní: <https://balsamiq.com/wireframes/>. Je zde také možnost si nástroj stáhnout a využívat ho jako

desktopovou aplikací. S desktopovou verzí pak můžeme pracovat bez připojení k internetu. Funkce se ale nikterak neliší.



Obrázek 3.7: Balsamiq Mockups: horní panel

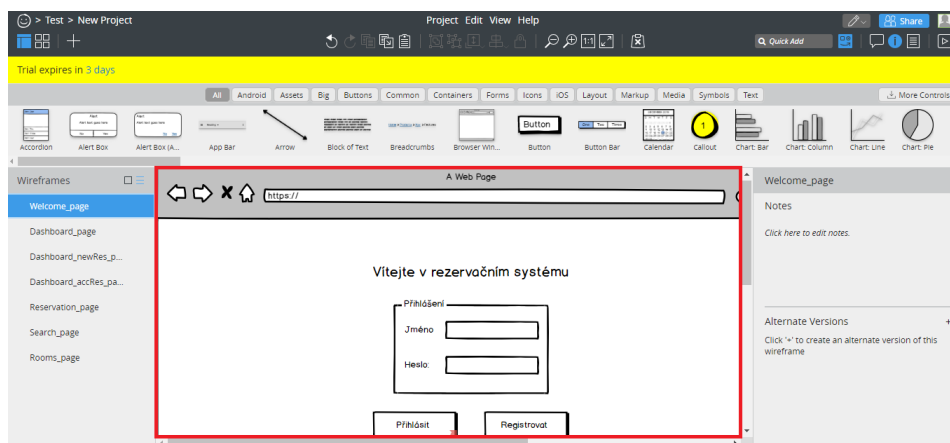
Zde (obr. 3.7) najdeme horní panel nástrojů. Můžeme si zde vybrat formát zobrazení wireframů, operace jako kopírování, krok zpět a vpřed, vkládání, zoom in a out, zobrazení komentářů, informací o projektu a spuštění prezentace.



Obrázek 3.8: Balsamiq Mockups: panel prvků

Pod horním panelem najdeme panel všech prvků (obr. 3.8), které nástroj poskytuje. Můžeme si zde také vybrat zobrazení pouze určitých skupin prvků. Například pro mobilní aplikace typu android nebo iOS.

3. Rešerše prototypovacích nástrojů



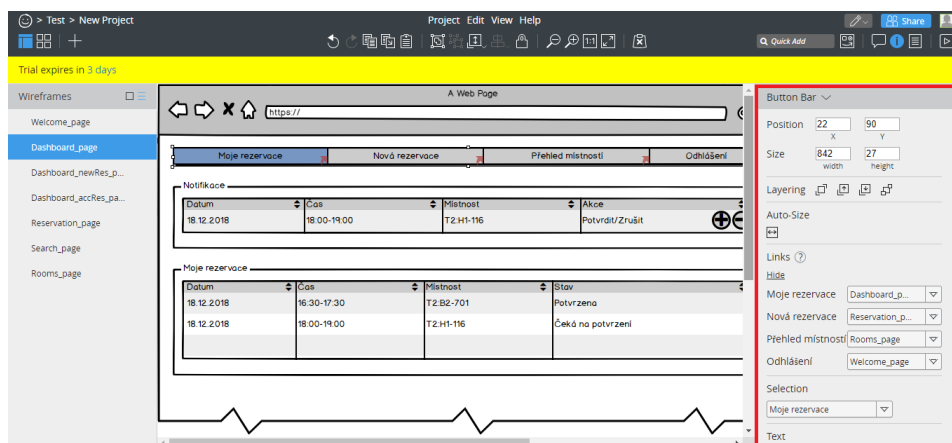
Obrázek 3.9: Balsamiq Mockups: hlavní obrazovka

Uprostřed se nachází hlavní obrazovka (obr. 3.9), kde můžeme manipulovat s aktuálním wireframem, přetahovat zde prvky z lišty a manipulovat se vzhledem aplikace.



Obrázek 3.10: Balsamiq Mockups: panel prvků

Panel na levé straně (obr. 3.10) nám ukazuje všechny wireframy aktuálního projektu, můžeme zde vidět na jakém aktuálně pracujeme a máme zde možnosti jak upravit jméno, kopírovat a smazat wireframe.

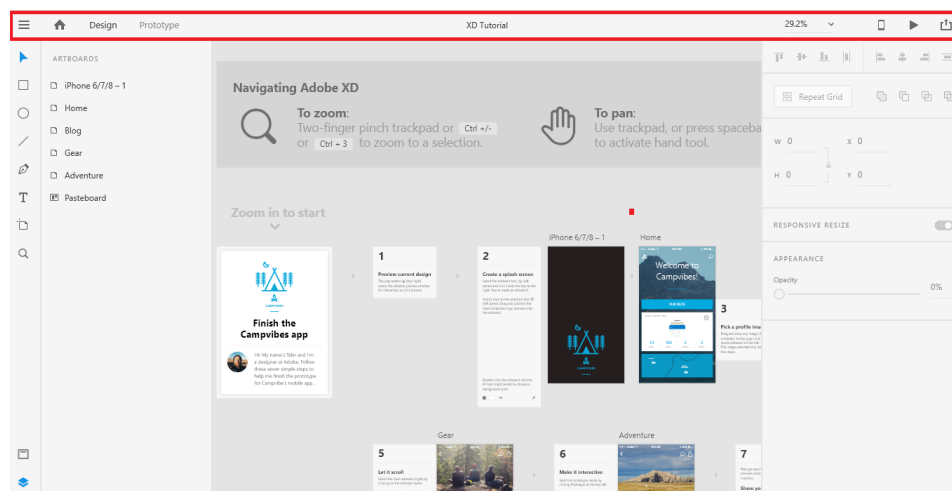


Obrázek 3.11: Balsamiq Mockups: pravý panel

Pravý panel (obr. 3.11) zobrazuje vlastnosti aktuálně vybraného prvku. Pokud je tedy označen některý z prvků tlačítek, můžeme zde nastavit i parametr link, kde vybereme, na jaký wireframe nás tlačítko přeměruje.

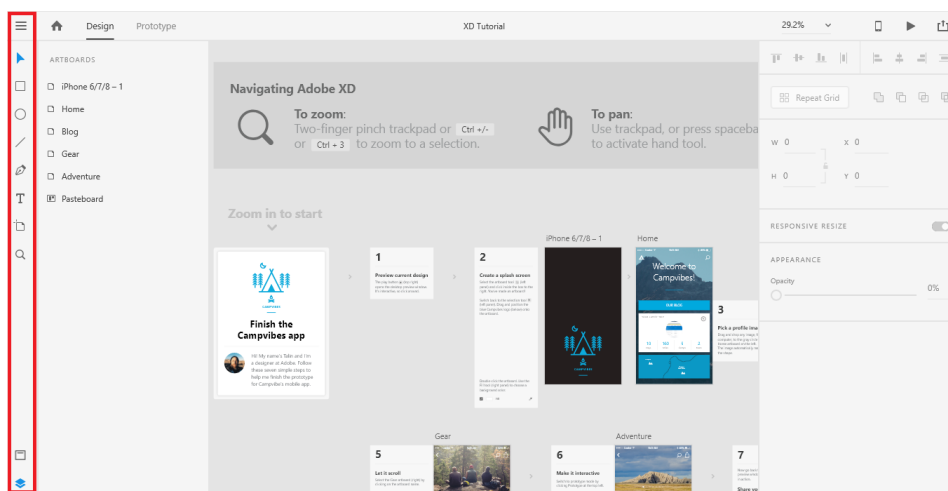
3.3.2 Popis nástroje: AdobeXD CC 2018

S tímto nástrojem jsem neměl žádné předchozí zkušenosti, zvolil jsem ho z důvodu, že je v podstatě nový a zajímalo mě, jaké možnosti tento nástroj poskytuje jak se s ním pracuje.



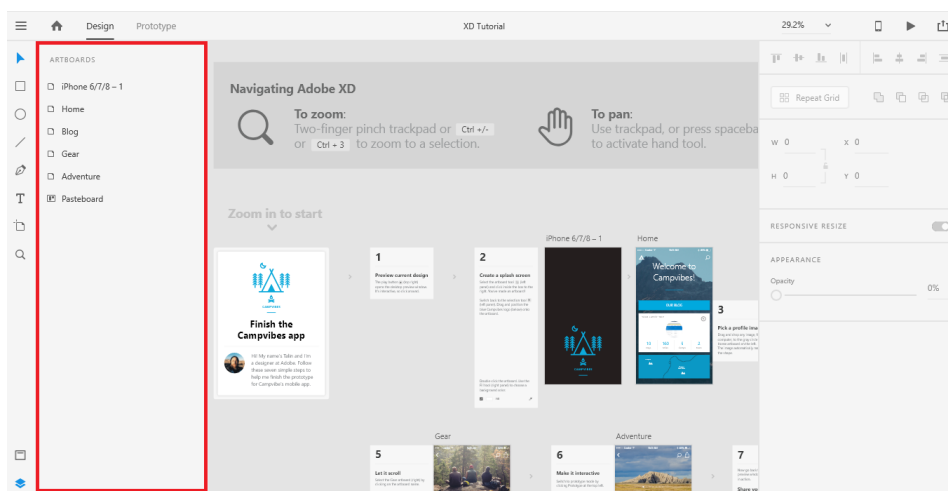
Obrázek 3.12: Adobe XD: horní lišta

Horní lišta (obr. 3.12) projektu obsahuje ikonu pro správu projektu, jako je vytvoření nového projektu, uložení, otevření stávajícího a podobné. Na levé straně se nachází záložka Design a Prototype, záložka Design nám slouží pro tvorbu a návrh jednotlivých obrazovek, záložka Prototype pak pro propojení těchto obrazovek, zvolení přechodů a jiné. Na pravé straně najdeme ikony pro prezentaci jak na zvoleném zařízení, tak na aktuálním.



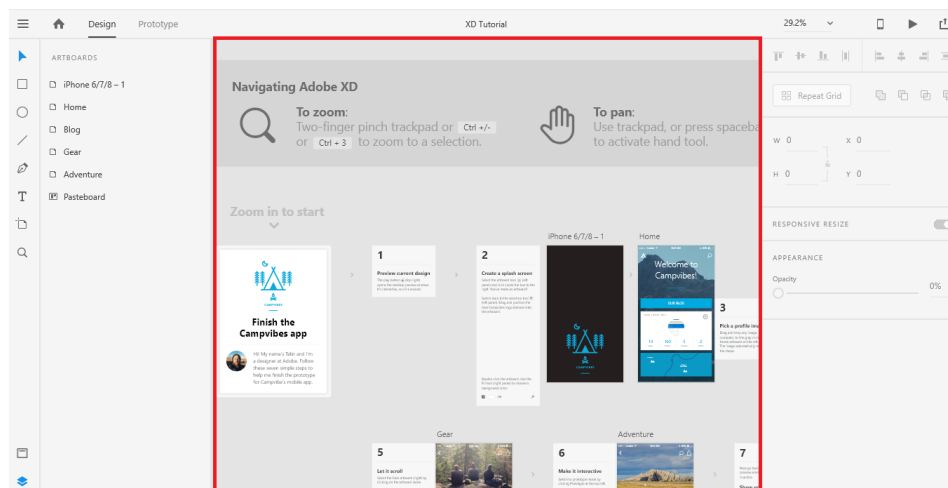
Obrázek 3.13: Adobe XD: levá lišta

Po levé straně najdeme lištu s nástroji (obr. 3.13), ty slouží pro přidání textu, obdélníků, elips, čar. Nachází se zde také nástroj pro přidání artboardů, to znamená předpřipravených oken pro zařízení jako je iPad, tablety nebo web stránky. Ve spodní části najdeme dvě tlačítka, kde přepínáme mezi artboardy a assets, to bude popsáno na dalším obrázku.



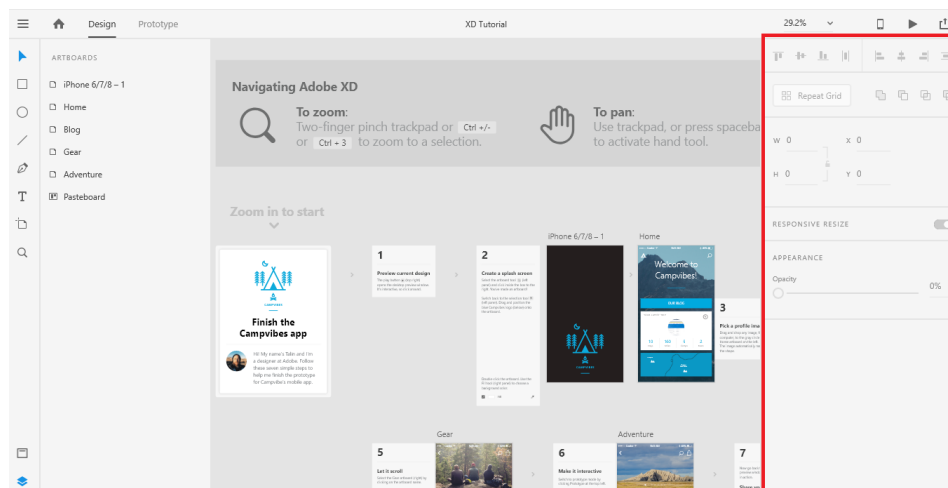
Obrázek 3.14: Adobe XD: levý panel

Tento panel (obr. 3.14) nám zobrazí všechny artboards, které jsou momentálně v našem projektu, máme k nim tak lehčí přístup a ve velkých projektech přehlednost. To samé zde platí i pro druhou možnost a to jsou assets, neboli styly, které v projektu používáme.



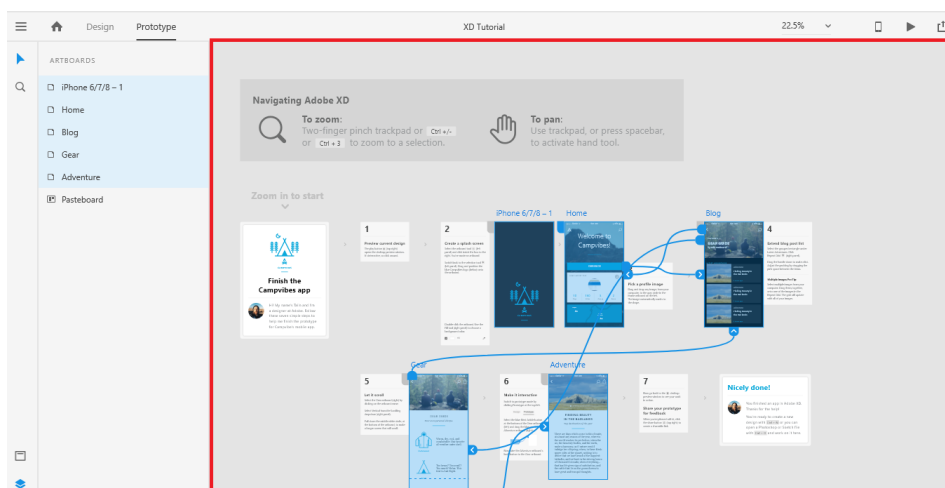
Obrázek 3.15: Adobe XD: hlavní obrazovka

Hlavní obrazovka (obr. 3.15) pro design. Zde můžeme manipulovat s našimi artboardy, tvořit je a upravovat.



Obrázek 3.16: Adobe XD: pravý panel

Po pravé straně se nachází panel (obr. 3.16) s vlastnostmi selektovaného prvku. Tyto vlastnosti zde můžeme upravovat, popřípadě nastavit opakování.



Obrázek 3.17: Adobe XD: prototyp

Hlavní obrazovka pro prototypování (obr. 3.17). Zde můžeme nastavovat přechody mezi artboardy, které jsme vytvořili, neplatí to pouze pro jednotlivé artboardy, ale i nějaký prvek nás může přesměrovat na jiný artboard.

3.4 Shrnutí rešerše

V rámci rešerše jsme si popsali, co to prototyp a prototypování znamená, řekli jsme si z jakých důvodů se prototypování využívá, porovnali jeho výhody a nevýhody a doporučili jsme si postup, jakým máme při tvorbě prototypu postupovat.

Následně jsme provedli rešerši existujících prototypovacích nástrojů, které vyhovují požadavkům předmětu (viz. kap. 2). Tyto nástroje jsme si popsali a porovnali. Z těchto nástrojů jsme vybrali dva, se kterými jsme si ukázali postup práce.

Nástrojem Balsamiq Mockup v rámci práce následně vytvořím prototyp (viz. kap. 5), který bude součástí návrhu aplikace.

Kapitola 4

Analýza prostředí IBM Cloud

Tato kapitola navazuje na požadavky předmětu B6B16INS (viz. kap. 2), kde v rámci výuky studenti nasazují svoje aplikace do cloudu od společnosti IBM. My si zde řekneme co to cloud znamená, popíšeme si jeho základní rozdělení a porovnáme výhody a nevýhody jejich použití. Popíšeme si zde prostředí IBM Cloud, jeho využití pro předmět a připravíme si nástroje, které pro práci s prostředím budeme potřebovat.

4.1 Cloud obecně

Obecný název cloud nese mnoho významů, ale nejčastěji je považován za model vývoje a používání počítačových technologií. Můžeme si pod tím také představit poskytování služeb či programů na cloudovém serveru. Principem těchto služeb je poskytování výpočetního výkonu uživatelům, kteří za něj následně platí. V této sekci čerpám ze zdrojů [2] [3] [14] [23] [10].

4.1.1 Rozdělení cloudů podle distribuce

- **IaaS** - Infrastructure/Integration as a Service (infrastruktura/integrace jako služba) tj. poskytovatel poskytne klientovi virtualizovaný hardware na základě požadavků, kde si klient bude moci řešit například výpočetní úlohy a podobně. Příklady poskytovatelů: Amazon WS a IBM Cloud.
- **PaaS** - Platform as a Servive (platforma jako služba) tj. poskytovatel poskytne klientovi služby pro podporu celého životního cyklu aplikace a služby související s poskytováním webových služeb. Každý poskytovatel, ale nabízí pouze omezené prostředky jako jsou programovací jazyky. Příklady poskytovatelů: Google App, IBM Cloud a Microsoft Azure.

- **SaaS** - Service/Software as a Service (služba/software jako služba) tj. poskytovatel poskytne klientovi hotový software s licencí rovnou na používání. Klient tedy může mít k aplikaci přístup odkudkoliv a kdekoliv. Příklady poskytovatelů: Google Apss, Cargopass.

■ 4.1.2 Výhody využití cloudů

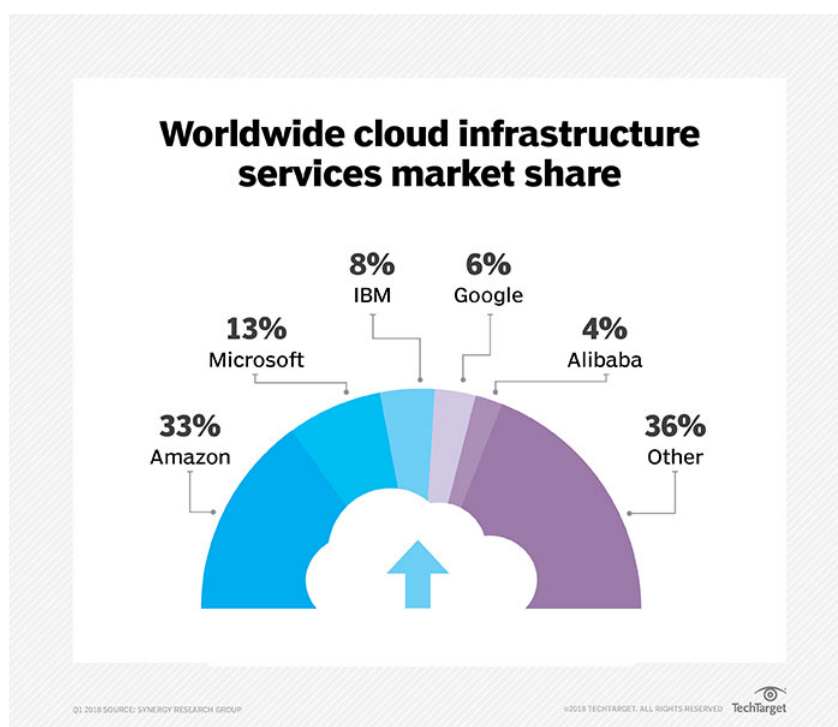
- Bez potřeby vlastnit HW, či znát potřebný SW.
- Dostupnost aplikace odkudkoliv.
- Možnost okamžitého upgradu HW, SW.
- Vyšší zabezpečení dat, ve většině případech.
- Možnost zvýšení výkonu.
- Možnost investic do jiných příležitostí než do IT.

■ 4.1.3 Nevýhody využití cloudů

- Závislost na internetovém připojení.
- Závislost na poskytovateli – zákazník se musí podvolit, které verze SW používat. Horší komunikace s poskytovatelem v případě větších cloudů. Poskytovatel může změnit ceny případně zbankrotovat.
- Cloud je stále nový pojem. Neexistuje žádné dlouhodobé, spolehlivé řešení.
- Migrační náklady – pro některé firmy, může být krok přechodu do cloudu ztrátový.
- Méně funkcí a horší stabilita.
- Odlišný právní řád poskytovatele a klienta.

■ 4.1.4 Budoucnost cloudů

V blízké budoucnosti vývoj cloudových řešení poroste exponenciálním růstem. Podle zdroje [5] ale stále na trhu zůstanou čtyři hlavní hráči (viz. obr. 4.1) a to je: Amazon Web Services, Microsoft, Google a IBM. Další krok pro cloud computing bude, že cloudy nebudou privátní, veřejné nebo dokonce kombinace těchto dvou, ale dosáhne se dynamické provázanosti mezi nimi, co znamená, že privátní cloud bude využívat různé služby několika veřejných cloudů.



Obrázek 4.1: Trh cloudových služeb [8]

4.2 Prostředí IBM Cloud

V rámci této sekce si popíšeme jak prostředí IBM Cloud funguje a jaké možnosti studentům nabízí. Následně si ukážeme, jak v tomto prostředí aktivovat účet a co budeme potřebovat pro nasazení a správu aplikace v tomto prostředí.

Prostředí IBM Cloud se v rámci předmětu B6B16INS (viz. kap. 2) využívá pro nasazení prototypu aplikace studenty. V rámci spolupráce školy a IBM, tato společnost nabízí využití zdarma po dobu 6 měsíců.

4.2.1 Popis IBM Cloud

Platforma IBM Cloud je kombinace platformy jako služby (PaaS) a infrastruktury jako služba (IaaS). Tato platforma podporuje nejen malé developerské týmy a organizace, ale také i velké Enterprise byznysy. Datová centra cloudu jsou rozmístěna všude po světě, a tak je možné přistupovat k datům odkudkoliv. Tyto data jsou vždy dostupná a řádně zabezpečená [8] [10].

Platforma se skládá z několika hlavních komponent tvořících celek, který poskytuje konzistentní cloudovou službu:

- Katalog nabídek, který poskytuje stovky servisů.
- Konzole, která slouží jako front end pro vytváření, zobrazení a správu cloudových zdrojů.
- Identifikační a přístupový management komponent, které uživatelé přidělují přístup k jednotlivým servisům.
- Vyhledávací a tagovací mechanismus pro filtrování a identifikaci zdrojů.
- Management účtu a platebních záležitostí poskytujících informace o přesném používání zdrojů a následném placení za tyto zdroje.

■ 4.2.2 Možnosti hostování aplikace v IBM Cloud

IBM Cloud nabízí mnoho možností pro hostování aplikace, poskytuje možnosti, které dají tolik kontroly nad prostředím nebo HW, kolik potřebujeme [10].

Na výběr máme z následujících možností:

- **Servlet function** – FaaS (Funkce jako služba) poskytuje programovací platformu založenou na Apache OpenWhisk, kde můžete pomocí servletů generovat dynamický obsah webových aplikací.
- **Cloud Foundry** – prostředí umožňující vytvoření více instancí, je izolované a je založené na PaaS více v následující sekci.
- **Docker container in Kubernetes** – kombinace kontajnerů s technologií Kubernetes, přímo poskytující bezpečnost, automatické nasazení, operace, škálování a sledování kontajnerových aplikací.
- **VMware** – prostředí sloužící pro virtualizaci jednoho či více počítačových jednotek.
- **Virtual machine** – slouží pro škálování serverů, u kterých si volíme jádra a paměť.
- **High-performance Bare Metal Servers** – Servery určené pouze pro jednotlivce, s nikým jiným se nesdílí, veškeré nastavení máte povoleno. Většinou placené hodinově.

Vzhledem k tomu, že nemám žádné přechozí zkušenosti s těmito možnostmi, jsem se rozhodl pro demonstraci nasazení aplikace do tohoto prostředí (viz. kap. 8) pomocí prostředí Cloud Foundry. V následujícím kroku si toto prostředí popíšeme.

■ 4.2.3 Prostředí Cloud Foundry v IBM Cloud

Cloud Foundry je prostředí postavené na tak zvané „container-based“ architektuře, kde jako vývojaři spojujeme určité funkční bloky (back-end, front-end, databáze), které jsou na sobě nezávislé a dohromady pak tvoří aplikaci [13].

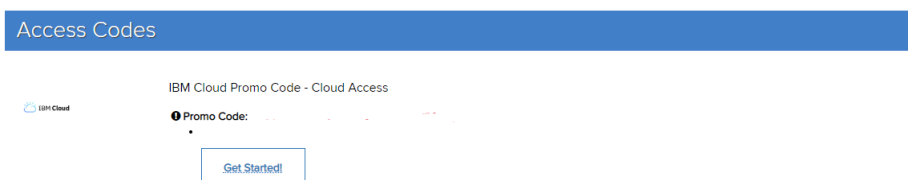
Díky této architektuře, každá cloudová aplikace by měla být psána podle „The Twelve Factors“ [11]. Tento návod popisuje architekturu a vlastnosti každé SaaS, která po nasazení bude:

- dobře škálovatelná bez velkých změn
- pochopitelná pro vývojáře, kteří se nepodíleli na tvorbě
- připravená pro nasazení do moderních cloudů

■ 4.2.4 Postup aktivace studentského účtu IBM Cloud

Zde si ukážeme postup, pomocí kterého si studenti předmětu B6B16INS (viz. kap. 2) aktivují svůj účet v prostředí IBM Cloud.

- Vytvoříme si účet na <https://cloud.ibm.com/login>.
- Zaregistrujeme se na <https://ibm.onthehub.com/>.
 - Při verifikaci vybereme, že se chceme registrovat s emailem organizace.
 - Zadáme svůj školní email.
 - Dokončíme registraci.
 - V záložce „Students“ vybereme „Cloud“ a v ní vybereme „IBM Cloud“.
 - Klikneme na „IBM Cloud Promo Code“ a přidáme tuto službu do košíku „Add to Cart“.
 - Potvrdíme formulář a vyplníme dotazník.
 - Následně když se podíváme na náš účet v záložce „Orders & Downloads“ bychom měli vidět aktivovanou službu.
 - Po zobrazení detailu této služby najdeme promo kód pro IBM Cloud (viz. obr. 4.2).



Obrázek 4.2: Přístupový kód IBM Cloud

- Aktivace služby.
 - Přihlásíme se na <https://cloud.ibm.com/login>.
 - V pravém horním rohu v záložce „Manage“ vybereme „Account“.
 - V levém panelu vybereme „Account settings“.
 - V sekci „Subscription and feature codes“ klikneme na „Apply Code“.
 - Vložíme kód získaný ve druhém kroku tohoto postupu a stiskneme „Apply“.

Po dokončení tohoto postupu máme vytvořený a aktivovaný účet v prostředí IBM Cloud.

■ 4.2.5 Nástroje pro práci s prostředím IBM Cloud

Pro práci s prostředím IBM Cloud budeme využívat nástroj IBM Cloud CLI, který prostředí poskytuje a nástroj GIT, který nám práci s prostředím usnadní. Popíšeme si zde využití těchto nástrojů a jejich instalaci.

- IBM Cloud CLI
Příkazová řádka pro IBM Cloud, je jeden z nejsilnějších nástrojů tohoto cloudu. Pomocí něho můžeme ovládat všechny zdroje a komponenty, které budeme potřebovat.

Popis instalace IBM Cloud CLI:

- Na adrese <https://cloud.ibm.com/docs/cli?topic=cloud-cli-install-ibmcloud-cli> si stáhneme instalační soubor pro náš operační systém.
- Nainstalujeme IBM Cloud CLI.
- V příkazové řádce ověříme funkčnost příkazem [bx --version].

Následně ověřte dostupnost nástrojů Cloud Foundry příkazem [bx cf --version] a developerských nástrojů [bx dev --version]. Pokud příkaz [dev] nebude rozpoznán, použijte příkaz: [bx plugin install dev -r blue-mix] to samé platí pro příkaz [cf] použijte: [bx plugin install cf -r bluemix].

Poznámka: Pokud po instalaci nefunguje žádný z uvedených příkazů, v proměnných prostředí vašeho systému zadejte cestu, kde je nástroj nainstalován.

■ GIT

V rámci vývoje aplikace budeme pracovat verzovacím nástrojem - Gitem. Git je jeden z nejrozšířenějších nástrojů využívaný programátory, který umožňuje správu zdrojových kódů. Díky tomuto faktu prostředí umožňuje správu zdrojových kódů a následné nasazování pomocí GitHubu.

Popis instalace nástroje GIT:

- Jděte na adresu <https://git-scm.com/downloads>.
- Stáhněte verzi pro váš operační systém.
- Nainstalujte Git.
- V příkazové řádce ověřte funkčnost příkazem [git --version].

■ 4.3 Shrnutí analýzy prostředí IBM Cloud

V rámci analýzy prostředí jsme si dle požadavků předmětu B6B16INS (viz. kap. 2) popsali toto prostředí a cloudy obecně. Řekli jsme si možnosti, které v rámci hostování aplikace máme a možnost hostování pomocí Cloud Foundry jsme si blíže popsali. Nakonec jsme si ukázali nástroje, které budeme potřebovat pro práci s prostředím při nasazení aplikace do cloudu.

V rámci nasazení aplikace (viz. kap. 8) si v prostředí Cloud Foundry tuto aplikaci nasadíme.

Kapitola 5

Analýza a návrh aplikace pro rezervaci místností

V této kapitole se budeme zabývat návrhem aplikace pro rezervaci místností. Popíšeme zde požadavky, které by měla aplikace pro rezervaci místností splňovat. Podle požadavků vytvoříme diagramy případů užití, procesní a stavové. Následně vytvoříme diagram tříd a prototyp této aplikace pomocí prototypovacího nástroje (viz. kap. 3).

5.1 Požadavky aplikace

Požadavky aplikace pro rezervaci místností popisují, co má daná aplikace dělat, aby mohla rezervovat místnosti. Pro popis požadavků využijeme FURPS+ analýzu vytvořenou společností Hawlett-Packard, protože v rámci předmětu B6B16INS (viz. kap. 2) studenti vytváří FURPS+ analýzu v případové studii.

FURPS+ analýza se dívá na kvalitu softwaru či informačního systému z pěti základních hledisek: funkčnost, použitelnost, spolehlivost, výkon a podporovatelnost [4].

V této sekci se zaměříme na požadavky aplikace související s rezervací místností. Na základě těchto požadavků vytvoříme diagramy stavové, procesní a případů užití.

Vysvětlivky požadavků: FR – funkčnosti, TR – spolehlivost, UR – použitelnost, PR – výkon, SR – podporovatelnost.

■ 5.1.1 Požadavky na funkčnost

FR01.1 Vytvoření rezervace

- Aplikace umožní uživateli si vytvořit rezervaci na libovolnou nerezervovanou místnost v určitý čas a určitý den minimálně 1hod předem. Místnost následně nebude v systému dostupná pro jiné uživatele po dobu této rezervace.

FR01.2 Zrušení rezervace

- Aplikace umožní uživateli zrušit nepotvrzenou rezervaci.

FR01.3 Změna rezervace

- Aplikace umožní uživateli změnit datum a čas nepotvrzené rezervace.

FR01.4 Přehled rezervací

- Aplikace zobrazí uživateli všechny jím aktuálně rezervované místnosti. Aplikace zobrazí i rezervace, které již proběhly.

FR01.5 Potvrzení rezervace

- Po obdržení notifikace, musí uživatel potvrdit svou rezervaci v přehledu rezervací, a tím se stane rezervace potvrzenou.

FR01.6 Notifikace o rezervaci

- Aplikace upozorní uživatele 1hod předem o jeho rezervaci. V přehledu rezervací uživatele se objeví upozornění.

FR01.7 Vypršení rezervace

- Po obdržení notifikace, když uživatel nepotvrdí svou rezervaci do 0,5hod, tak rezervace vyprší.

FR02.1 Přidání místností

- Aplikace umožní administrátorovi přidat novou místnost do systému.

FR02.2 Odebrání místností

- Aplikace umožní administrátorovi odebrat místnost ze systému.

FR02.3 Úprava místností

- Aplikace umožní administrátorovi upravit již existující místnost v systému.

FR02.4 Přehled místností

- Aplikace zobrazí přehled všech místností. V přehledu zobrazí, kdy jsou jednotlivé místnosti zarezervované

FR03.1 Registrace do systému

- Nový uživatel se bude moci zaregistrovat. Po validaci se v systému vytvoří nový uživatel s údaji z registrace.

FR03.2 Přihlášení do systému

- Validní uživatel se bude moci přihlásit do systému a využívat systémové zdroje.

FR03.3 Odhlášení ze systému

- Aplikace odhlásí přihlášeného uživatele.

FR03.4 Validace

- Aplikace zkontroluje všechny uživatelské vstupy a pokud jsou vstupy validní, odešle požadavek. Pokud vstupy validní nejsou, pošle uživateli upozornění na nevalidní vstupy.

FR04.1 Vyhledávání

- Aplikace vyhledá všechny místnosti, které jsou k dispozici pro zadaný datum a čas.

■ 5.1.2 Požadavky na použitelnost

UR01 Jazyk

- Aplikace bude psána v anglickém jazyce.

UR02 Prohlížeč

- Aplikace bude dostupná v internetových prohlížečích IE, Firefox, Chrome, Opera, Edge, Safari a ve verzích k době realizace aktuálních.

■ 5.1.3 Požadavky na spolehlivost

TR01 Dostupnost

- Dostupnost aplikace dvacet čtyři hodin sedm dní v týdnu mimo aktualizace aplikace.

TR02 Aktualizace

- Data aplikace musí být v pravidelných intervalech aktualizována, to znamená minimálně jedenkrát za měsíc.

■ 5.1.4 Požadavky na výkon

PR01 Škálovatelnost

- Aplikace obslouží stovky uživatelů a musí garantovat škálovatelnost v řádu stovek uživatelů.

PR02 Odezva

- Aplikační doba odezvy bude v desetinách sekundy.

■ 5.1.5 Požadavky na podporovatelnost

SR01 Databáze

- Aplikace bude využívat databázi pro ukládání veškerých dat.

SR02 Prostředí

- Aplikace bude nasazena v prostředí IBM Cloud.

FR04 Zdrojový kód

- Zdrojový kód aplikace bude psaný v anglickém jazyce a okomentovaný.

V rámci implementace aplikace (viz. kap. 6) nebudeme implementovat všechny z těchto požadavků, protože budeme vytvářet pouze funkční prototyp aplikace pro rezervaci místností. Některé z těchto požadavků jsou zde pouze, aby specifikace požadavků byla úplná (viz. škálovatelnost a funkčnosti admina).

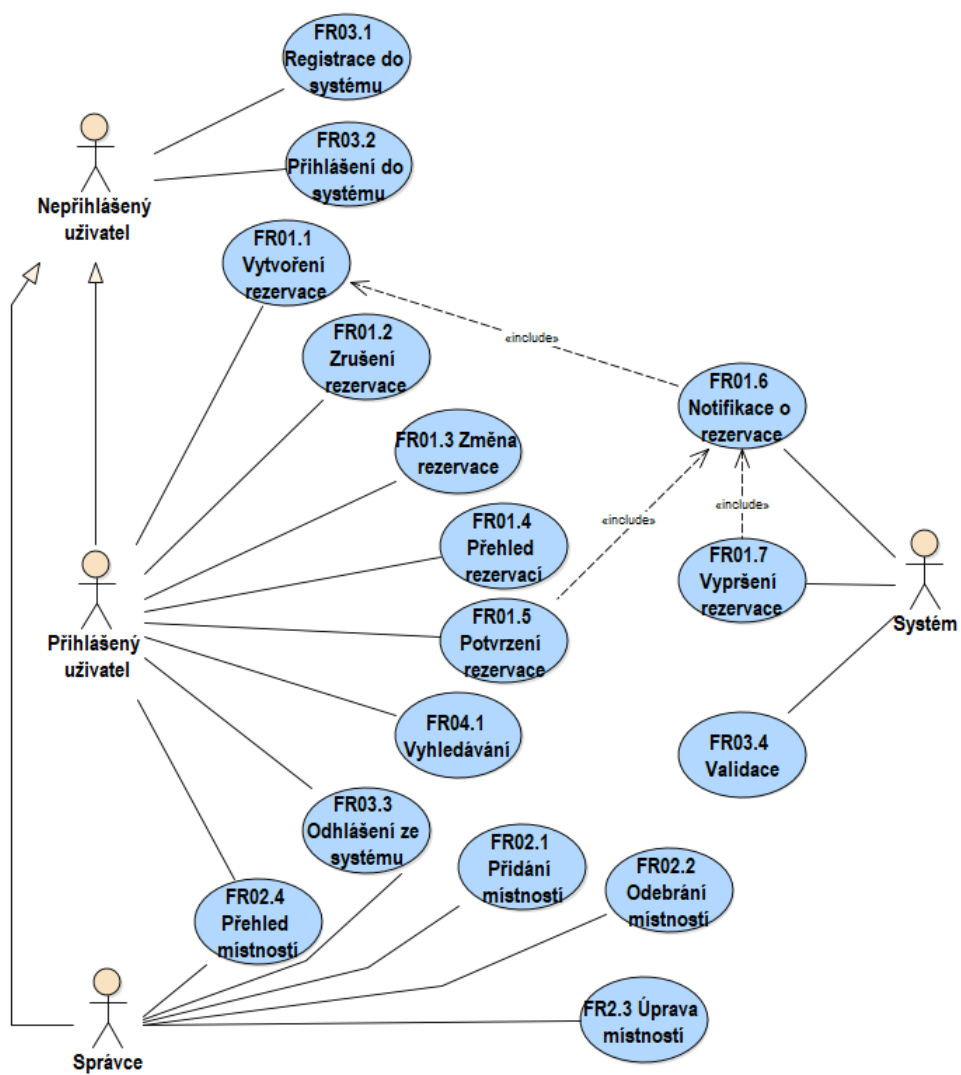
■ 5.2 Diagramy aplikace

Pro modelování a tvorbu diagramů zde využíváme Unified Modeling Language (UML), který definuje standarty pro jednotnou strukturu diagramů. Slouží k specifikaci, vizualizaci, konstrukci a dokumentaci částí softwarového systému [27]. Pro modelování v tomto jazyku budeme využívat nástroj Enterprise Architect verze 12 se školní licenci. Všechny diagramy zde popisovat nebudu, ale popíšeme si zde jen ty, které využijeme.

■ 5.2.1 Diagram případů užití aplikace

Diagram případů užití (use-case diagram) obsahuje dvě komponenty, kterými jsou aktéři a případy užití. Aktéři nám říkají, které osoby se systémem mohou pracovat v rámci určité role. Případy užití pak jsou funkce systému, které aktéři vyvolávají a dostávají viditelný výsledek jako zpětnou vazbu [26].

V aplikaci se budou nacházet čtyři role aktérů a to jsou: nepřihlášený uživatel, přihlášený uživatel, správce a systém. Jednotlivé případy užití (viz. obr. 5.1) vychází z požadavků aplikace (viz. sekce požadavky).

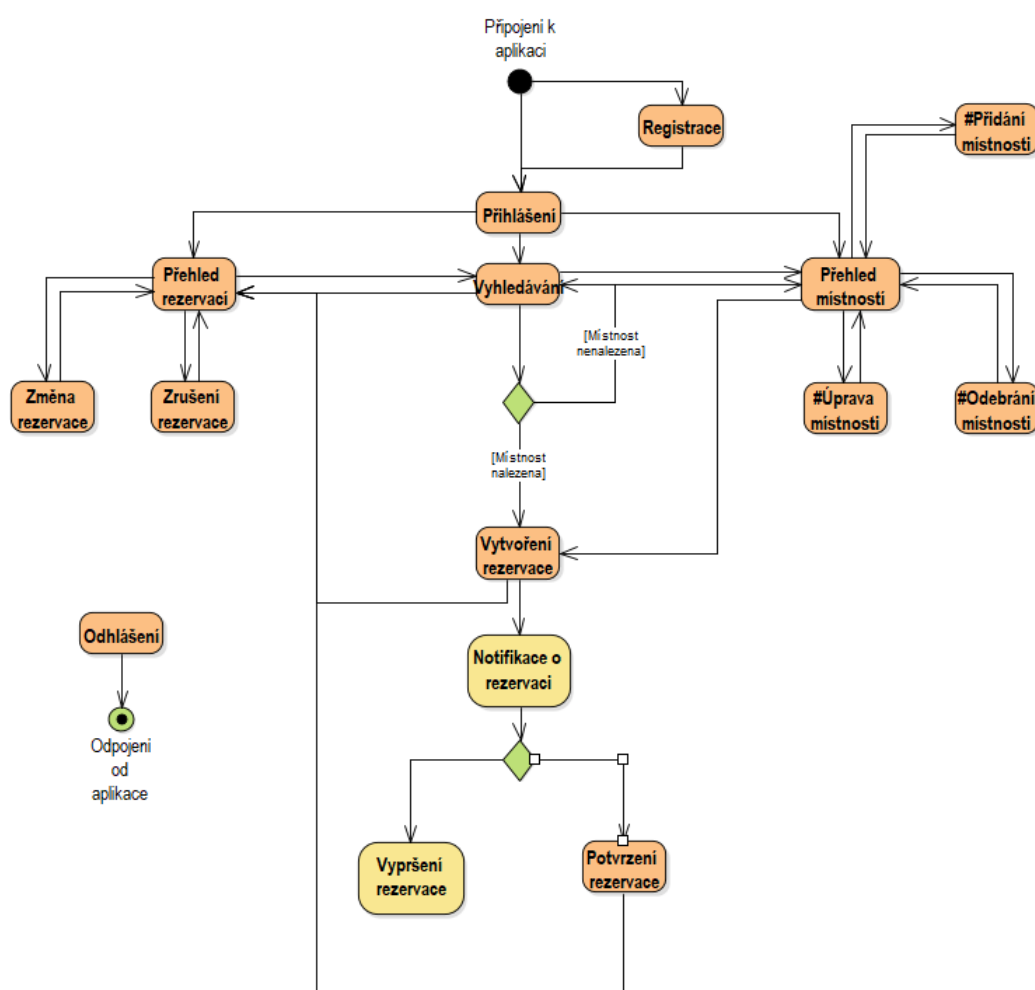


Obrázek 5.1: Use-case diagram

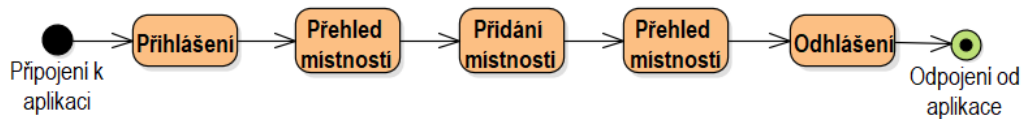
5.2.2 Procesní diagramy aplikace

Procesní diagram slouží k lepšímu pochopení textového vyjádření procesů. Říká, v jaké posloupnosti mají být realizovány jednotlivé činnosti procesů, a jaké vstupy a výstupy procesy potřebují pro správné fungování [27].

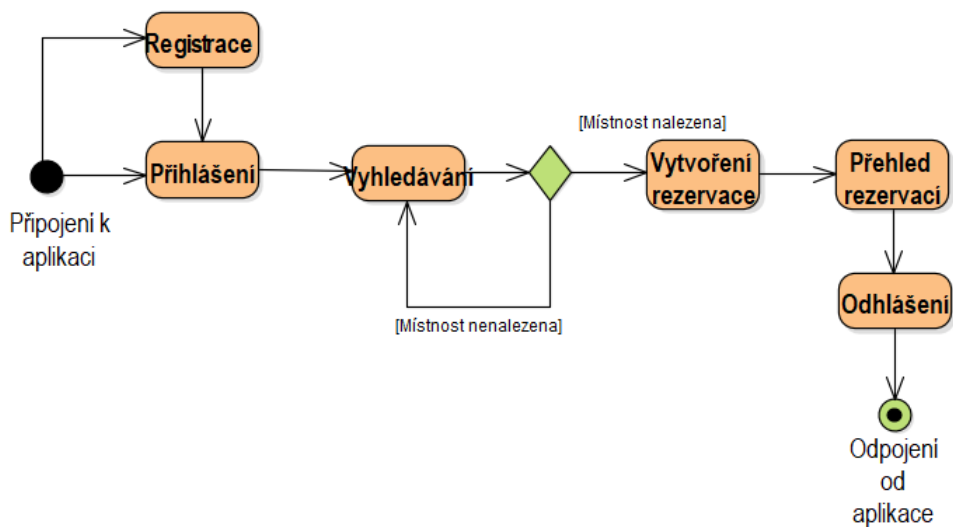
Procesní diagram (obr. 5.2) obsahuje veškeré procesy, které se v aplikaci vyskytují. Rozhodl jsem se tento diagram vytvořit, abych tak nejlépe zobrazil všechny procesy a vazbu mezi nimi. Na obrázcích (obr. 5.3-5.5) pak můžeme vidět některé z procesů samostatně.



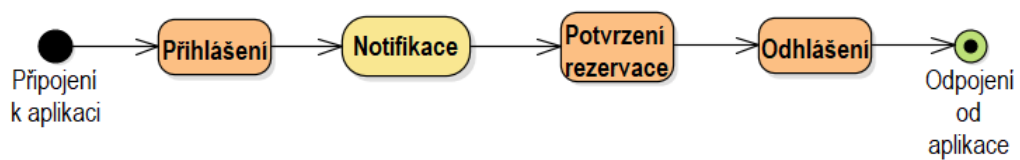
Obrázek 5.2: Procesní diagram celé aplikace



Obrázek 5.3: Proces: Vytvoření místnosti



Obrázek 5.4: Proces: Vytvoření rezervace

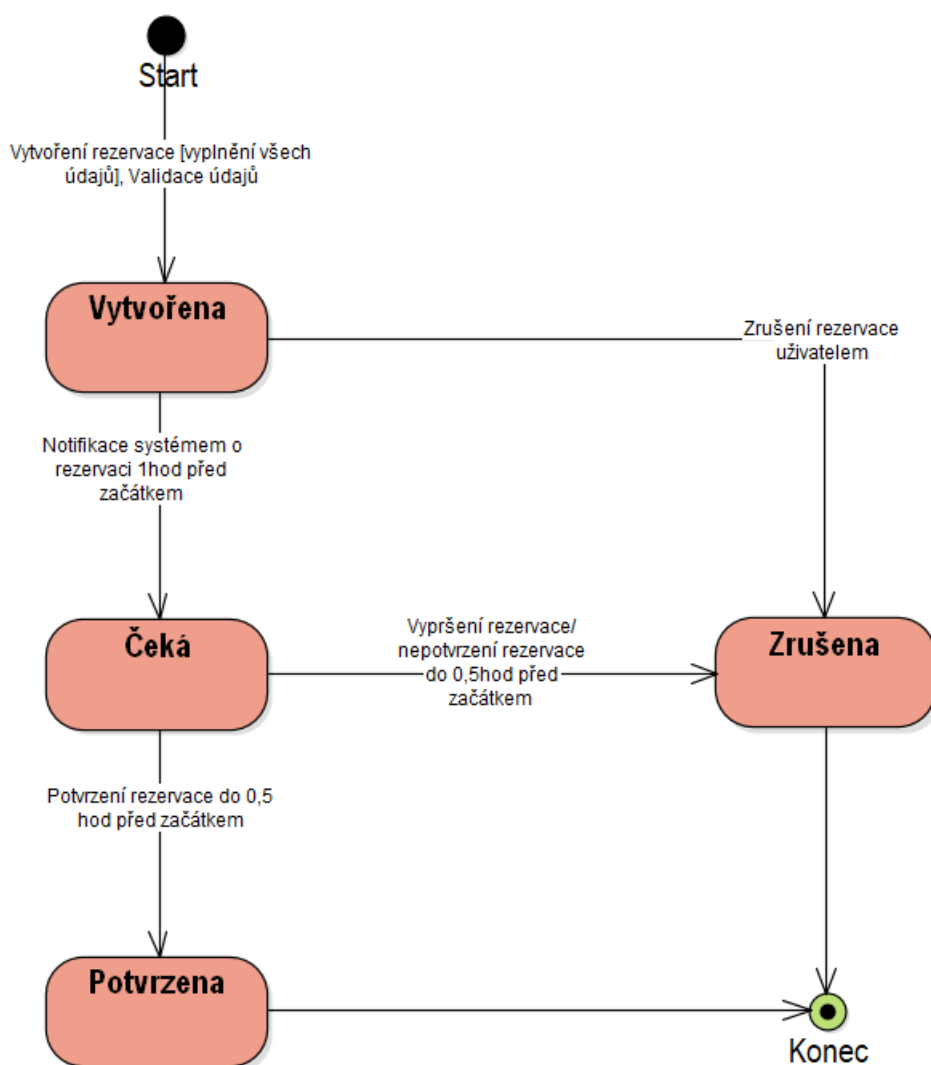


Obrázek 5.5: Proces: Vytvoření rezervace

5.2.3 Stavový diagramy aplikace

Stavový diagram je grafického znázornění komponenty a jejich stavů v systému. Popisuje vývoj této komponenty a říká, čím je přechod mezi stavy způsoben [26].

Stavový diagram rezervace (obr. 5.6) popisuje stavy, ve kterých se rezervace může nacházet a přechody mezi nimi.

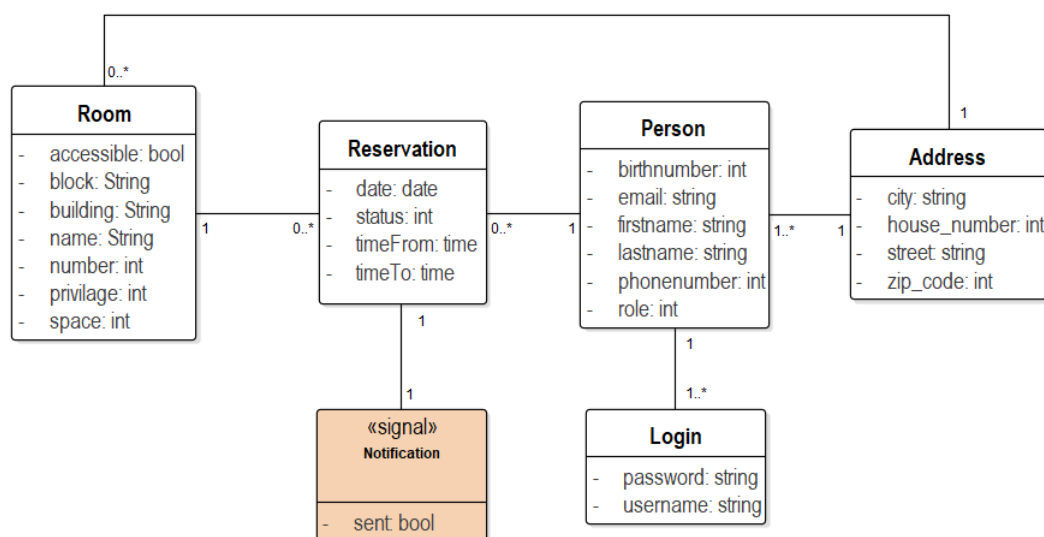


Obrázek 5.6: Stavový diagram: rezervace

5.2.4 Diagram tříd aplikace

Diagram tříd se používá k popisu struktury systému. Hlavní elementy jsou třídy a vztahy mezi nimi. Třídy mohou poskytovat atributy a operace. Vztahy nám říkají, jak jsou spolu třídy provázány. Cílem je vytvoření všech tříd, které se v systému nacházejí a podle nich systém implementovat [26].

V diagramu tříd (obr. 5.7) jsem zachytil hlavní jádro aplikace pro rezervaci místností. Z těchto tříd tak vznikl model komponent, které budeme v aplikaci implementovat (viz. kap. 6). Tento diagram je vytvořen na základě požadavků (viz. sekce Požadavky aplikace) a případů užití (viz. sekce Diagram případů užití aplikace).



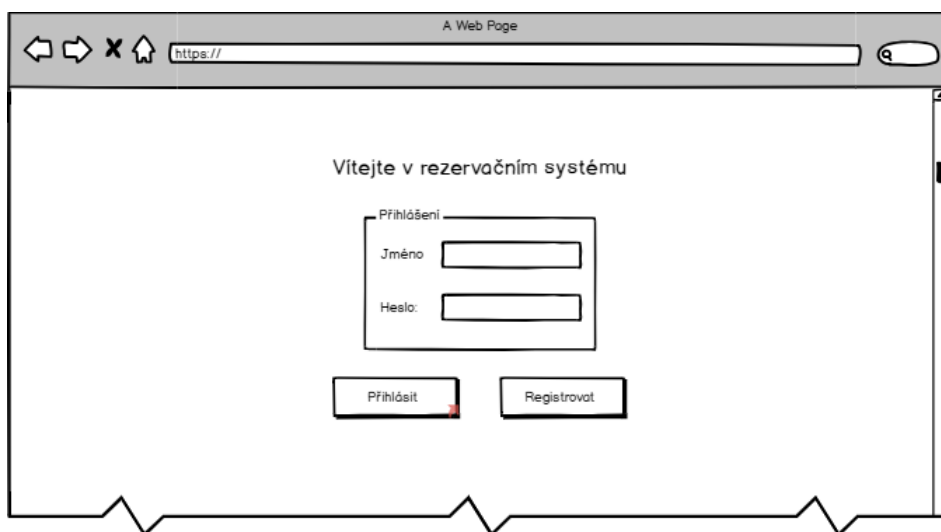
Obrázek 5.7: Diagram tříd

Poznámka: V doménovém modelu se nachází i notifikace (viz. sekce Požadavky aplikace), přidal jsem jí, protože mi to přijde jako důležitá část pro vyobrazení, díky které víme, jak se stav rezervace bude měnit.

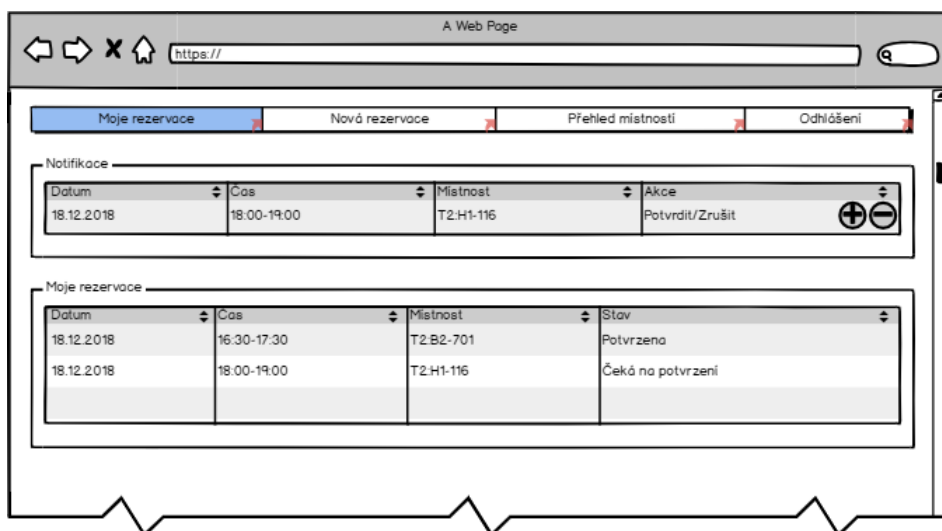
5.3 Prototyp aplikace

Na základě požadavků předmětu (viz. kap. 2) v rámci návrhu vytvoříme také prototyp aplikace (viz. kap. 3), kde jsme si řekli, co to prototyp je a k čemu slouží.

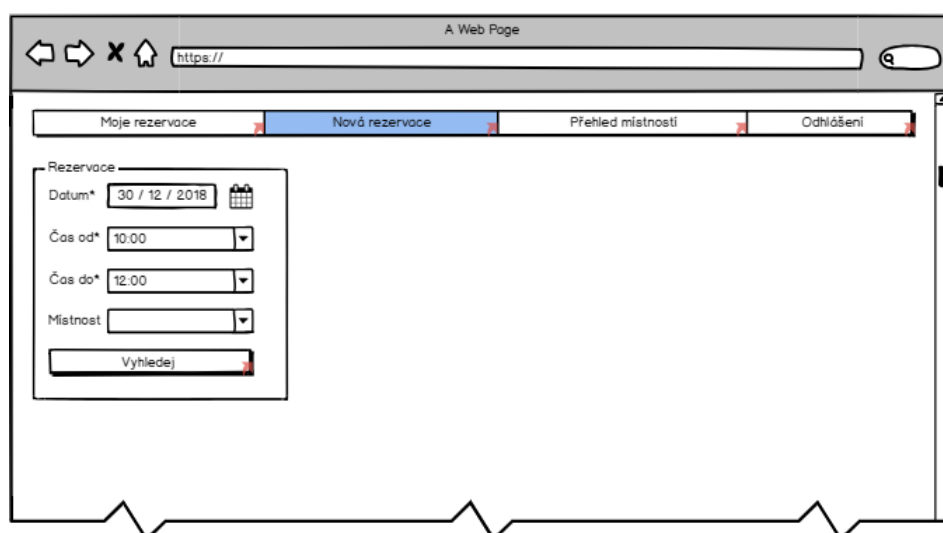
Snímky vytvořeného prototypu (viz. obr. 5.8 - 5.14) v prototypovacím nástroji Balsamiq Mockups (viz. kap. 3), které jsem vytvořil na základě konzultací s vedoucím práce.



Obrázek 5.8: Prototyp: Přihlášení



Obrázek 5.9: Prototyp: Moje rezervace



Obrázek 5.10: Prototyp: Vyhledání rezervace

Rezervace

Datum* 30 / 12 / 2018

Čas od* 10:00

Čas do* 12:00

Místnost

Vyhledej

Dostupné místnosti

Místnost	Datum	Čas	Akce
T2.H1-116	30.12.2018	10:00-12:00	Rezervovat
T2.C3-205	30.12.2018	10:00-12:00	Rezervovat
KNE-103	30.12.2018	10:00-12:00	Rezervovat
KNE-311	30.12.2018	10:00-12:00	Rezervovat

Obrázek 5.11: Prototyp: Vytvoření rezervace

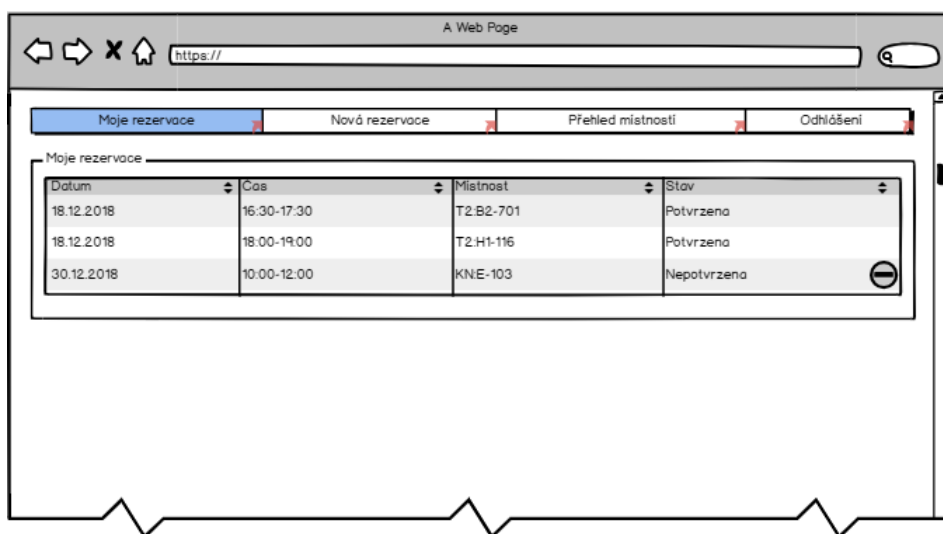
Notifikace

Datum	Čas	Místnost	Akce
18.12.2018	18:00-19:00	T2.H1-116	Potvrdit/Zrušit

Moje rezervace

Datum	Čas	Místnost	Stav
18.12.2018	16:30-17:30	T2.B2-701	Potvrzena
18.12.2018	18:00-19:00	T2.H1-116	Čeká na potvrzení
30.12.2018	10:00-12:00	KNE-103	Nepotvrzena

Obrázek 5.12: Prototyp: Rezervace vytvořena



Obrázek 5.13: Prototyp: Rezervace potvrzena



Obrázek 5.14: Prototyp: Přehled místností

5.4 Shrnutí analýzy aplikace

V rámci této kapitoly jsme namodelovali a navrhli systém na základě požadavků (viz. sekce Požadavky aplikace), které plynou z podstaty rezervačního systému a požadavků předmětu B6B16INS (viz. kap. 2). Popsali jsme si, k čemu jednotlivé typy diagramů používáme a co nám popisují. Nakonec jsme vytvořili prototyp aplikace, který bude sloužit jako vzor implementace (viz. kap. 6) aplikace pro rezervaci místností.

Kapitola 6

Implementace aplikace

V této kapitole si popíšeme implementaci aplikace. Tato implementace probíhala na základě návrhu aplikace (viz. kap. 5). Popíšeme si jaké technologie jsme zvolili pro implementaci, k čemu nám slouží a jakou má naše aplikace architekturu.

Pro implementaci rezervačního systému jsem se rozhodl použít šablonu z předmětu B6B33EAR – „Enterprise Architektury“. Tato šablona poskytuje základní kostru pro tvorbu webových aplikací postavených na frameworku Spring. Tato šablona pak spadá pod licenci GNU a škola mi dala veškerá práva pro práci s touto šablonou.

6.1 Použité technologie

Back-end (BE) aplikace je psaná v jazyce Java a využívá framework Spring. Tento framework je zde využit pro implementaci modelu (viz. kap. 5), autentizaci, autorizaci a přístup k datům v databázi pomocí Object Relation Mapping (ORM) využitím Java Persistence API (JPA). Propojení BE aplikace s front-endem (FE) je zařízeno pomocí REST API. Na straně FE stojí systém Node.js, pod kterým běží JavaScript psaný v knihovně React.js. Pro ukládání dat jsem zvolil PostgreSQL databázi, se kterou mám nejvíce zkušeností.

6.2 Popis technologií

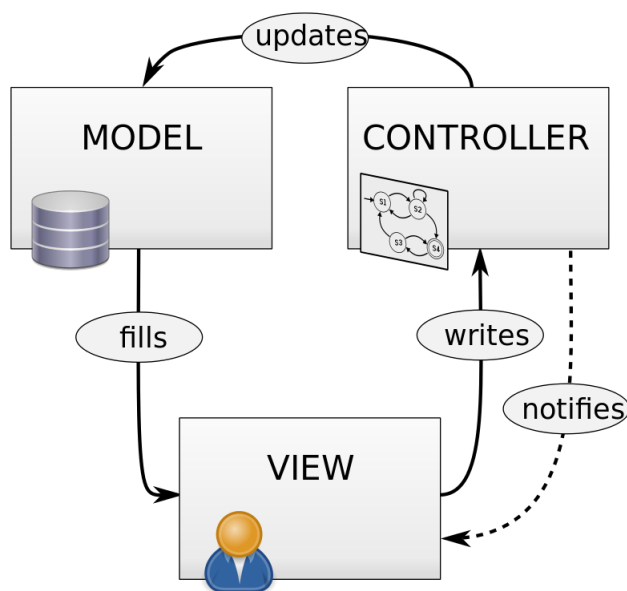
- Spring - framework sloužící pro jednoduchý vývoj J2EE aplikací [25].

- J2EE - součást platformy Java určená pro vývoj webových aplikací, vkládání závislostí, přístup k relačním databázím (ORM), vývoj sdílené logiky (EJB) a další [24].
- ORM - programovací technika v softwarovém inženýrství, která zajišťuje autamtickou konverzi dat mezi relační databází a objektově orientovaným programovacím jazykem [15].
- JPA - standard jazyka Java umožňující ORM [12].
- Node.js - softwarový systém pro tvorbu JavaScriptových aplikací [9].
- React.js - JavaScriptová knihovna vytvořená Facebookem.

6.3 Architektura aplikace

Aplikace se drží webové architektury Model-View-Controller (MVC), která rozděluje datový model aplikace, uživatelské rohraní a řídicí logiku do tří nezávislých komponent tak, že modifikace kterékoliv z nich by měla mít minimální vliv na ostatní komponenty (viz. obr. 6.1) [22].

- Model - slouží k reprezentaci informací, s nimiž aplikace pracuje.
- View - převádí data z modelu do podoby vhodné pro uživatele.
- Controller - reaguje na události od uživatele a zajišťuje změny v modelu a pohledu.



Obrázek 6.1: Model-View-Controller

6.4 Implementované požadavky

Při implementaci požadavků (viz. kap. 5), jsem se soustředil primárně na ty, které jsou nezbytné pro rezervaci místností. Požadavky, které jsem v aplikaci neimplementoval jsou: FR2.1, FR2.2 a FR2.3. S vedoucím práce jsme se dohodli, že tyto požadavky, můžeme postrádat při demonstraci rezervace místností, kvůli časové tísni.

6.5 Shrnutí implementace

V této kapitole jsme si popsali implementovanou aplikaci, vytvořenou dle návrhu (viz. kap. 5). Řekli jsme si jaké tato aplikace používá technologie a popsali jsme si architekturu aplikace. V sekci požadavky jsme si řekli, které požadavky (viz. kap. 5) splňuje, a které ne. Snímky a práce s aplikací jsou popsány v příloze (B).

Implementovanou aplikaci je potřeba následně podrobit uživatelskému testování (viz. kap. 7), abychom dostali zpětnou vazbu pro práci s aplikací předtím, než aplikaci nasadíme do provozu.

Kapitola 7

Uživatelské testování aplikace

Po implementaci (viz. kap 6) aplikaci podrobíme uživatelskému testování, abychom zjistili zpětnou vazbu uživatelů, kteří si práci s touto aplikací vyzkouší. Zde si popíšeme, jak testování probíhalo a vyhodnotíme, zda aplikace splňuje požadavky pro rezervaci místností, a zda je tato aplikaci použitelná pro reálný provoz.

7.1 Popis testování

Testování aplikace proběhlo za účelem získání zpětné vazby uživatelů. Během testování účastník plní zadané úkoly, ve kterých otestuje běžné používání aplikace pro rezervaci místností. U účastníka sedí moderátor, který toto testování vede.

Kroky testování vychází z požadavků (viz. kap. 5). a byly následující:

1. Zaregistrujte se do systému (pozn. použijte libovolné ID number).
2. Vytvořte novou rezervaci.
 - a. Zadejte libovolný čas od 9 – 17hod.
 - b. Zadejte zítřejší datum.
 - c. Vyberte libovolnou místnost.
3. Jděte do přehledu místností.
 - a. Zkontrolujte, zda místnost, pro kterou jste rezervaci vytvořili, je v daný čas a datum opravdu zarezervována.
4. Rezervaci odstraňte.

5. Vytvořte novou rezervaci.
 - a. Zadejte čas: Aktuální čas + (30–60 min), tak aby vyhovoval kontrole.
 - b. Zadejte aktuální datum.
 - c. Zvolte libovolnou místnost.
6. Rezervaci potvrďte.
7. Odhlaste se ze systému.

7.1.1 Dotazník

Každý uživatel po absolvování testování dostal dotazník, na základě kterého zhodnotil aplikaci a její používání.

V rámci dotazníku jsme se účastníků ptali na následující otázky:

- Pohlaví
- Věk
- Zkušenosti s rezervačním systémem
- Jak hodnotí práci se systémem
- Zda je systém přehledný
- S čím si v rámci testování nevěděli rady
- Co by uvítali pro zjednodušení systému
- Jaké další funkce by uvítali

7.2 Testování aplikace

Testování se účastnilo 6 respondentů z toho 3 studenti ČVUT a 3 další osoby. Všichni účastníci byli informováni, že se jedná pouze o prototyp aplikace pro rezervaci místností.

7.2.1 Účastník 1

Účastník měl problém s registrací, kde nemohl správně napsat identifikační číslo, účastník našel BUG v aplikaci, který jsem následně opravil. Testování probíhalo bez větších problémů. U kroku (3.b.) nevěděl, jak zadat datum

ve správném formátu do formuláře. Zároveň měl poznámky k vylepšení přehlednosti aplikace, kde navrhl vytvoření interaktivního přehledu místností, myšleno mapou, kterou je možné se proklikat k rezervaci.

■ 7.2.2 Účastník 2

Účastník měl problém s anglickým jazykem aplikace, kde jsem mu musel pomáhat s překladem. Kroky procházel pomalu, protože neměl zkušenost s jiným rezervačním systémem. Vždy se ujistil, zda to co dělá je správné. Díky nezkušenosti s rezervačními systémy následně nevěděl, jak v kroku (3) vložit formáty jednotlivých vstupů. Poté co se mu to povedlo, neměl jiné problémy.

■ 7.2.3 Účastník 3

Účastník neměl problém s žádným z úkolů, které měl splnit. Měl pouze připomínky k některým funkcím a vlastnostem. Například: proč systém nevyhledá volnou rezervaci podle zvolené místnosti, nebo podle zvoleného data a času místnost.

■ 7.2.4 Účastník 4

Tento účastník byl velmi aktivní, ale když měl splnit nějaký úkol, většinou kliknul na první, co ho napadlo. Proto se hned ze začátku místo registrace do systému snažil přihlásit. V kroku (3.b.) mu připadalo neobvyklé zadávání formátu data DD-MM-YY.

■ 7.2.5 Účastník 5

Účastník si dal při průchodu aplikace na čas, ale jako jediný využil nápovědy u textových polí. Tím si na rozdíl od jiných ulehčil práci. Testováním prošel bez problémů, přičemž poukazoval na věci, které by se do systému hodilo přidat.

■ 7.2.6 Účastník 6

Účastník na v prvním kroku místo aby se do systému registroval, se rovnou snažil o přihlášení. V rámci testování mu nesešel anglický jazyk. Při testování si nevěděl rady s krokem (4.a), kde nemohl najít, jak si zkontrolovat, zda je aplikace opravdu zarezervována.

7.3 Vyhodnocení výsledků testování

V průběhu testování se vyskytovaly dva opakující se problémy, které většině účastníků zabraly více času. Bylo to zadání správného formátu data do formuláře a následná kontrola, zda místnost je opravdu zarezervována v přehledu místností. Každý uživatel potom využil následné nápovědy u textového pole. Uživatelé by uvítali lepší design ať už barevný nebo interaktivní a následně předepsané hodnoty jednotlivých položek ve formuláři. V případě zadávání data by uvítali možnost kalendáře, kde by si sami mohli zvolit datum a nemuseli ho psát.

Na základě těchto připomínek můžeme říci, že aplikace plní požadavky na rezervaci místnost a zároveň dává uživatelům potřebnou přehlednost, která není zcela dokonalá. Vzhledem k tomu, že se jedná o první implementovanou verzi, jsem s výsledkem nadmíru spokojený a pokládám to za úspěch.

V rámci další implementace bych doplnil kalendář pro přidání data, funkce pro správu aplikace jako správce a vizuální zobrazení místností v budově.

Kapitola 8

Nasazení aplikace do prostředí IBM Cloud

V této kapitole jsme původně měli navázat a nasadit implementovanou a otestovanou aplikaci (viz. kap. 6). Ovšem kvůli tomu, že prostředí IBM Cloud nepodporuje framework Spring, na kterém je aplikace postavena (viz. kap. 6), tak jsme se s vedoucím práce dohodli na vytvoření step-by-step manuálu práce s prostředím.

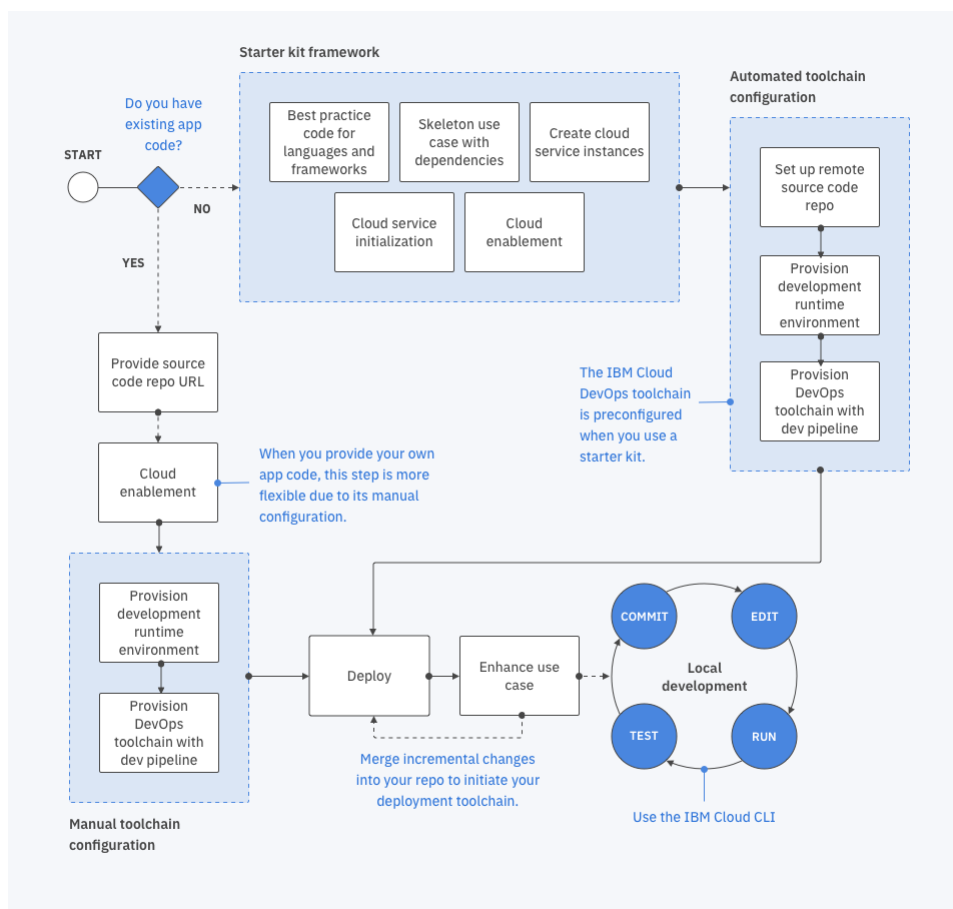
Ukážeme si zde tedy vytvoření a nasazení Node.js aplikace do prostředí Cloud Foundry (viz. kap. 4). Vytvoříme Continuous Delivery Toolchain, přidáme NoSQL databázi a připravíme si prostředí pro následný vývoj této aplikace.

Node.js aplikaci jsem zvolil, protože jsem Node.js využíval v implementované aplikaci (viz. kap. 6).

8.1 Možnosti nasazení

Možnosti nasazení aplikace (viz. obr. 8.1) do Cloud Foundry v IBM Cloud máme dvě:

- Nová aplikace – vytvoření aplikace cloudem, prostředí za nás vygeneruje potřebné konfigurační soubory, které při nasazení budeme potřebovat.
- Existující aplikace – pokud se rozhodneme pro tuto volbu **velmi doporučuji**, abychom si nejdříve zjistili, jaké toto prostředí podporuje frameworky, následně pak vytvořili potřebné kontainery a podle nich, modifikovali existující části aplikace.



Obrázek 8.1: Tvorba aplikace v IBM Cloud [10]

8.2 Prerekvizity nasazení

Každý kdo chce postupovat dle tohoto návodu, musí mít stažené a nainstalované nástroje pro práci s prostředím a přístup do prostředí IBM Cloud (viz. kap. 4).

Protože jsme si zvolili aplikaci v Node.js, budeme také potřebovat nainstalovat i tento nástroj:

- Na adrese <https://nodejs.org/en/download/> stáhneme verzi pro náš operační systém.
- Nainstalujeme Node.js.
- V rámci instalace se ujistíme že instalujeme „node package manager“ (npm). Defaultně je součástí instalace.
- V příkazové řádce ověříme funkčnost příkazem `[npm --version]`.

Poznámka: Pokud využíváte nějaké IDE pro vývoj aplikací. Doporučuji se kouknout, zda vaše IDE poskytuje doplňky pro jednodušší nasazení aplikace do IBM Cloud. Například zde je doplněk pro IntelliJ: <https://www.ibm.com/blogs/bluemix/2017/02/deploy-manage-java-applications-intellij/>.

8.3 Nasazení do cloudu

Zde si ukážeme postup, kterým vytvoříme prostředí Cloud Foundry a kontainer pro Node.js aplikaci, následně k němu přidáme databázi a vytvoříme řetězec akcí, které se budou provádět vždy při změně na GITu.

V rámci nasazení jsem vytvořil video, které naleznete v přílohách nebo případně na YouTube: <https://www.youtube.com/watch?v=wWekU0tAK8A>.

8.3.1 Struktura postupu nasazení

- Vytvoříme Cloud Foundry aplikaci.
- Vytvoříme Continuous Delivery Toolchain.
- Přidáme NoSQL databáze.
- Přípravíme si prostředí pro vývoj.
- Nastaveníme Delivery Pipeline.

8.3.2 Vytvoření Cloud Foundry aplikace

Prostředí Cloud Foundry jsme si popsali (viz. kap. 4). Nyní v tomto prostředí apliaci vytvoříme. Postup vytvoření aplikace:

1. Na hlavní stránce IBM Cloud v levém horním rohu rozbalíme menu a klikneme na „Cloud Foundry“.
2. Zvolíme možnost „Public Applications“.
3. Zobrazí se nám možnosti, ze kterých si můžeme vybrat, jakou aplikaci chceme vytvořit. V našem postupu si ukážeme práci s Node.js, a proto zvolíme možnost „SDK for Node.js“.
4. Naší aplikaci dáme libovolné jméno a klikneme na „Create“.
Poznámka: Můžeme si všimnout, že zdarma verze Cloud Foundry je pouze na 30 dní, ale to je pro předmět B6B16INS (viz. kap. 2) dostačující.

5. Následně nás web přesměruje na stránku Getting started a my počkáme, až se u naší aplikace ukáže zelené kolečko a stav „Running“.
6. Na levé straně v panelu se přesunem do „Overview“, kde vidíme vše potřebné o naší aplikaci. V horní části si můžeme všimnout „Visit App URL“, kde najdeme námi vytvořenou aplikaci.
7. Nyní máme vytvořenou Node.js aplikaci v prostředí Cloud Foundry.

8.3.3 Vytvoření Continuous Delivery Toolchain

Continuous delivery toolchain je služba, která nám umožňuje při PUSHování kódu na GIT, vyvolat řetězec akcí, které v cloudu postaví a nasadí aplikaci. Tuto možnost můžeme také využít pro testování aplikace před nasazením.

1. V dolní části na pravé straně klikneme v sekci „Continuous Delivery“ na tlačítko „Enable“. Tato možnost nám vytvoří toolchain a repositář na githubu.
2. Pro jednoduchost jediné co uděláme, že po kliknutí na „Delivery Pipeline“ přidáme API Key, který nám buď prostředí vygeneruje nebo si zvolíme vlastní.
3. Nyní klikneme na „GIT“ pod nápisem Code. Měli bychom dostat upozornění, že stále nemáme nastavený SSH klíč. V tom případě pokračujeme podle tohoto návodu <https://help.github.com/en/articles/checking-for-existing-ssh-keys> případně <https://git.eu-gb.bluemix.net/help/ssh/README#locating-an-existing-ssh-key-pair>.
4. Nyní již máme vytvořený a připravený repositář na GitHubu a automaticky vygenerovaný řetězec událostí, který se zavolá při akci (PUSH) na GitHubu.

8.3.4 Přidání NoSQL databáze

NoSQL databáze jsou v dnešní době velmi používaným trendem, poskytují nám mechanismy pro ukládání a získávání dat. Většinou se používají pro big data a real-time webové aplikace. Na rozdíl od SQL databází nám poskytují jednodušší návrh, škálovatelnost a kontrolu nad databází.

1. V horní liště klikneme na záložku „Catalog“.
2. V levém panelu klikneme na „Databases“ a vybereme „Cloudant“.
3. Zvolíme jméno a nastavíme jakou autentikaci chceme. Pro naše účely bude stačit pouze IAM.

4. V přehledu služeb zvolíme tuto službu, budete ji moci nastavovat až bude mít status: Provisioned.
5. V panelu po levé straně klikneme na „Connections“ a „Create connection“.
6. Nyní vyplníme údaje a vyhledáme naši aplikaci. Klikneme na „Connect“.
7. Klikneme na „Connect & restage app“.

V této práci se již nebudeme zabývat propojením databáze s aplikací v cloudu či lokálně. Pokud tohoto budete chtít docílit pak můžete pokračovat podle tohoto video návodu <https://developer.ibm.com/tv/dw-mailbag-managing-cloud-foundry-credentials-application/>.

8.3.5 Příprava prostředí pro vývoj

Zde si ukážeme, jak si připravit prostředí pro vývoj aplikace na našem zařízení.

1. Otevřeme příkazovou řádku, vložíme příkaz [bx login] a přihlásíme se do IBM Cloud.
2. Zadáme příkaz [bx target -o <username> -s dev]. (-o organizace, pokud jste si nepřidali vaše kolegy do IBM Cloud, -o bude váš login)
3. Příkazem [bx cf services] se ujistíme, zda námi vytvořené služby v cloudu jsou dostupné.
4. Nyní si naklonujeme náš GitHub repozitár příkazem [git clone <https/ssh link>].
5. V naklonované složce spusťte příkaz [npm install] a následně [npm start].
6. Nyní můžete aplikaci psát lokálně a v konzoli se vám ukáže port localhostu, na kterém aplikace běží.

8.3.6 Nastavení Delivery Pipeline

Pipeline je součástí Continuous Delivery Toolchain, o kterém jsme si říkali v přechodném kroku. Pipeline nám říká co se má vykonat, pokud nastane námi určená akce. My si nakonfigurujeme pipeline, která nám při PUSH požadavku na GIT vyvolá nový build aplikace a následné nasazení. Tyto kroky se nám vygenerovaly automaticky při vytvoření „Continuous Delivery Toolchain“ ovšem považuji za vhodné, se tímto nastavením alespoň proklikat, protože tato část je v rámci tohoto cloudu jedna z velkých výhod.

1. Otevřeme si naši Cloud Foundry aplikaci a ve spodní části klikneme na tlačítko „View toolchain“, který jsme vytvořili v návodu „Vytvoření Continuous Delivery Toolchain“.
2. Pod nápisem „Deliver“ klikneme na „Delivery Pipeline“.
3. Pokud zde vidíme stage „Build“ a „Deploy“, tak již máme vytvořenou pipeline. My si ji vytvoříme manuálně. Proto je nyní odstraníme.
4. Pokud zde nic nemáme, klikneme na „Add Stage“. Pojmenujeme tuto fázi „Build“.
5. V záložce „Jobs“ klikneme na „ADD JOB“ a přidáme „Build“, kde typ buildu zvolíme „npm“.
6. Uložíme nastavení a vytvoříme nový step s názvem „Deploy“.
7. V záložce „Jobs“ klikneme na „ADD JOB“ a přidáme „Deploy“. V dolní části si také můžeme všimnout příkazů, které se volají při fázi deploy.
8. Uložíme toto nastavení.
9. Pokud již máme na GITu nějaký kód, můžete spustit build a deploy manuálně. Případně PUSHnout nový fragment kódu.

Nyní již máme připravené vše pro vývoj naší aplikace. V případě zájmu o další zajímavé funkce bych vám doporučil krátké show od developerů z IBM na této adrese <https://developer.ibm.com/tv/>.

8.4 Shrnutí práce s prostředím

Na základě práce s tímto prostředím jsem sepsal výhody a nevýhody, které souvisí s prostředím IBM Cloud.

Výhody prostředí IBM Cloud

- Po krátké chvíli jsem se s prostředím naučil pracovat.
- Přehlednost prostředí.
- Poskytuje Starter Kits pro snadnou a rychlou tvorbu aplikací.
- Support mi v rámci pár minut zodpověděl mé dotazy.
- Jeden z největších cloudů, prostředí dobře zdokumentované jinými lidmi narozdíl od jiných cloudů.

- Velký počet služeb a možnosti jejich kombinací.
- Stačí minimální zkušenost s buildem a deployem aplikace.

■ Nevýhody prostředí IBM Cloud

- Prostedí nepodporuje frameworky jako je Spring.
- Při prvním přihlášení se vám bude zdát prostředí nepřehledné.
- Tím, že obsahuje mnoho služeb, ne všechny jsou dobře zdokumentované.
- V rámci složitějších věcí musíte kontaktovat tým developerů na Slack kanálu.
- Většina návodů je pro IBM Bluemix předchůdce IBM Cloud.
- Neposkytuje tutoriály, jak vytvořit aplikaci step-by-step, ale pouze dokumentaci k určitým částem.
- Oficiální dokumentace je v angličtině a velmi odborná.

■ Celkové zhodnocení

Původním cílem této kapitoly bylo nasazení aplikace pro rezervaci místností do prostředí IBM Cloud. Jak jsem již zmínil v nevýhodách a na začátku této kapitoly, toto prostředí nespokytuje podporu frameworku Spring. I když ve většině dokumentacích to Spring nazývají, ale je tím myšleno Spring Boot.

Díky tomuto faktu jsme se s vedoucím práce dohodli, že vytvoříme step-by-step manuál, ve kterém jsme si popsali kroky od vytvoření aplikace, po její nasazení a správu prostředí, ve kterém je nasazena.

Práci s prostředím ve finále hodnotím kladně, když nastal nějaký problém ochotný support mi pomohl, nebo jsem našel řešení po delším hledání. Tím, že je IBM Cloud jeden z lídrů na trhu, tak lidé vytvářejí různé návody jak s prostředím pracovat.

V rámci předmětu B6B16INS bych potom každému studentovi doporučil aby v prostředí začal aplikaci vytvářet a ušetřil si tím pak čas strávený nad modifikací existující aplikace.



Kapitola 9

Závěr

V této práci jsem si za hlavní cíl položil rozšíření existujících výukových materiálů předmětu B6B16INS.

V rámci této práce jsem provedl analýzu předmětu B6B16INS, kde jsem zjistil problémy a požadavky tohoto předmětu. V návaznosti na tyto požadavky jsem provedl rešerši existujících prototypovacích nástrojů a analýzu prostředí IBM Cloud. Následně jsem začal s návrhem aplikace, kde jsem se poznatky z analýzy snažil maximálně uplatnit, aby výsledná aplikace mohla demonstrovat tyto poznatky. Prošel jsem od sběru požadavků, přes návrh, implementaci až k nasazení. Bohužel v rámci této práce se mi nepovedlo tuto aplikaci nasadit do prostředí IBM Cloud, a proto jsem se s vedoucím práce dohodl na vytvoření tutoriálů pro práci s tímto prostředím. Provedením těchto kroků jsem tak splnil stanovený cíl a rozšířil jsem výukové materiály předmětu B6B16INS o prototypování, prototypovací nástroje a práci s prostředím IBM Cloud.

Pokud by někdo chtěl na tuto práci navázat a dále tak rozšířit tyto materiály. Doporučil bych se zabývat prostředím IBM Cloud, kde by mohl vytvořit nějakou větší aplikaci tvořenou více kontejnery postavenou v Cloud Foundry využívající třeba Servlet Funkce.

Z této práce jsem si odnesl mnoho nových zkušeností, jako je například zkušenost s novými prototypovacími nástroji, zkušenost s cloudovým prostředím a reálná možnost podrobení aplikace testování uživateli. Věřím, že tato práce bude maximálně přínosná a to nejen předmětu a studentům B6B16INS.

Příloha A

Literatura

- [1] MCCLENDON, C. M., L. REGOT, G. AKERS. *The Analysis and Prototyping of Effective Graphical User Interfaces* [online]. [cit. 2018-11-20]. Dostupné z: <https://web.archive.org/web/20000503141817/http://www.ums1.edu/~s980548/gproj1/intro.html>.
- [2] *Cloud Computing pro každého: Jasně a stručně* [online]. [cit. 2019-05-01]. Dostupné z: <http://www.businessit.cz/cz/cloud-computing-vysvetleni-jasne-a-strucne.php>.
- [3] Wikipedia. *Cloud computing* [online]. [cit. 2019-05-01]. Dostupné z: https://en.wikipedia.org/wiki/Cloud_computing.
- [4] KENETT, R. a Emanuel R. BAKER. *Software process quality: management and control*. New York: Marcel Dekker c1999. Computer aided engineering (New York, N.Y.), 6. ISBN 978-0-8247-1733-9.
- [5] SAMUELS M. *Looking forward to the future of cloud computing* [online]. [cit. 2019-05-01]. Dostupné z: <https://www.raconteur.net/technology/future-cloud-computing>.
- [6] STACY, D. A. *Rapid Software Prototyping* [online]. [cit. 2018-11-20]. Dostupné z: <https://archive.vn/20010222114908/http://hebb.cis.uoguelph.ca/~dave/343/Lectures/prototype.html>.
- [7] OVERMEYER, S. P. *Revolutionary Vs. Evolutionary Rapid Prototyping: Balancing Software Productivity and Hci Design Concerns* [online]. [cit. 2018-11-20]. Dostupné z: https://www.academia.edu/328275/Revolutionary_Vs._Evolutionary_Rapid_Prototyping_Balancing_Software_Productivity_and_Hci_Design_Concerns.
- [8] ROUSE M. *IBM Cloud (formerly IBM Bluemix and IBM SoftLayer)* [online]. [cit. 2019-05-01]. Dostupné z: <https://searchcloudcomputing.techtarget.com/definition/IBM-Bluemix>.

- [9] PATEL P. *What exactly is Node.js?* [online]. [cit. 2019-05-19]. Dostupné z: <https://medium.freecodecamp.org/what-exactly-is-node-js-ae36e97449f5>.
- [10] IBM Cloud. *Getting started with IBM Cloud: What is the IBM Cloud platform?* [online]. [cit. 2019-05-01]. Dostupné z: <https://cloud.ibm.com/docs/overview?topic=overview-what-is-platform#what-is-platform>.
- [11] WIGGINS A. *The Twelve Factors* [online]. [cit. 2019-05-18]. Dostupné z: <https://12factor.net/>.
- [12] TYSON M. *THE JAVA PERSISTENCE SERIES: What is JPA? Introduction to the Java Persistence API* [online]. [cit. 2019-05-19]. Dostupné z: <https://www.javaworld.com/article/3379043/what-is-jpa-introduction-to-the-java-persistence-api.html>.
- [13] IBM SUPPORT CENTRE. *What is Cloud Foundry?* [online]. [cit. 2019-05-01]. Dostupné z: <https://cloud.ibm.com/unifiedsupport/supportcenter>.
- [14] BRIDGWATER A. *A guide to cloud data* [online]. [cit. 2019-05-01]. Dostupné z: <https://www.raconteur.net/technology/guide-cloud-data>.
- [15] FOWLER, M. *Patterns of enterprise application architecture*. . Boston: Assison-Wesley, c2003. ISBN 0-321-12742-0.
- [16] NIELSEN, J. *Usability Engineering*. 1. vyd. Boston: Academic Press, 1993. ISBN 0-12-518405-0.
- [17] WARCHOLINSKI, M. *What Is the Difference Between Wireframe, Mockup and Prototype?* [online]. [cit. 2018-11-20]. Dostupné z: <https://brainhub.eu/blog/difference-between-wireframe-mockup-prototype>.
- [18] VINCENT, L. *Wireframe vs Mockup vs Prototype & Selection of Prototyping Tools* [online]. [cit. 2018-11-20]. Dostupné z: <https://www.mockplus.com/blog/post/wireframe-mockup-prototype-selection-of-prototyping-tools>.
- [19] HEATON, S. *5 of Best Prototyping Tools to Test Out Your Web and Mobile Design* [online]. [cit. 2018-12-05]. Dostupné z: <https://www.shopify.com/partners/blog/62745923-5-of-the-best-prototyping-tools-to-test-out-your-web-and-mobile-designs>.
- [20] CAHILL C. *12 top prototyping tools* [online]. [cit. 2018-12-05]. Dostupné z: <https://blog.prototypr.io/10-best-prototyping-tools-for-ui-ux-designers-in-2018-6591eale2e71>.

- [21] FOSTER N. *The 20 best wireframe tools* [online]. [cit. 2018-12-05]. Dostupné z: <https://www.creativebloq.com/wireframes/top-wireframing-tools-11121302>.
- [22] BOREK B. *Úvod do architektury MVC* [online]. [cit 2019-05-19]. Dostupné z: <https://www.zdrojak.cz/clanky/uvod-do-architektury-mvc/>.
- [23] ROSSI B. *Cloud computing's next phase in business transformation* [online]. [cit. 2019-05-01]. Dostupné z: <https://www.raconteur.net/technology/cloud-computings-next-phase-business-transformation>.
- [24] JAKARTA EE. *Jakarta EE Working Group: About* [online]. [cit. 2019-05-18]. Dostupné z: <https://jakarta.ee/about/>.
- [25] MATULÍK P. *SPRING FRAMEWORK I – ÚVODNÍ POJMY* [online]. [cit. 2019-05-18]. Dostupné z: <https://vsadnajakvu.cz/2005-04/odborne/spring-framework/spring-framework-i-uvodni-pojmy/>.
- [26] ARLOW J., Neustat, I. *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.
- [27] FOWLER, M. *Destilované UML*. Praha: Grada, 2009. Knihovna programátora (Grada). ISBN 978-80-247-2062-3.

Příloha B

Aplikace: Rezervační systém

V této příloze jsou popsány programy, které pro vývoj aplikace potřebujeme, a postup, kterým si nastavíme aplikační prostředí. Součástí této přílohy jsou také snímky z aplikace.

B.1 Prerekvizity práce s aplikací

Pro vývoj a spuštění této aplikace musí být v systému nainstalovány tyto programy:

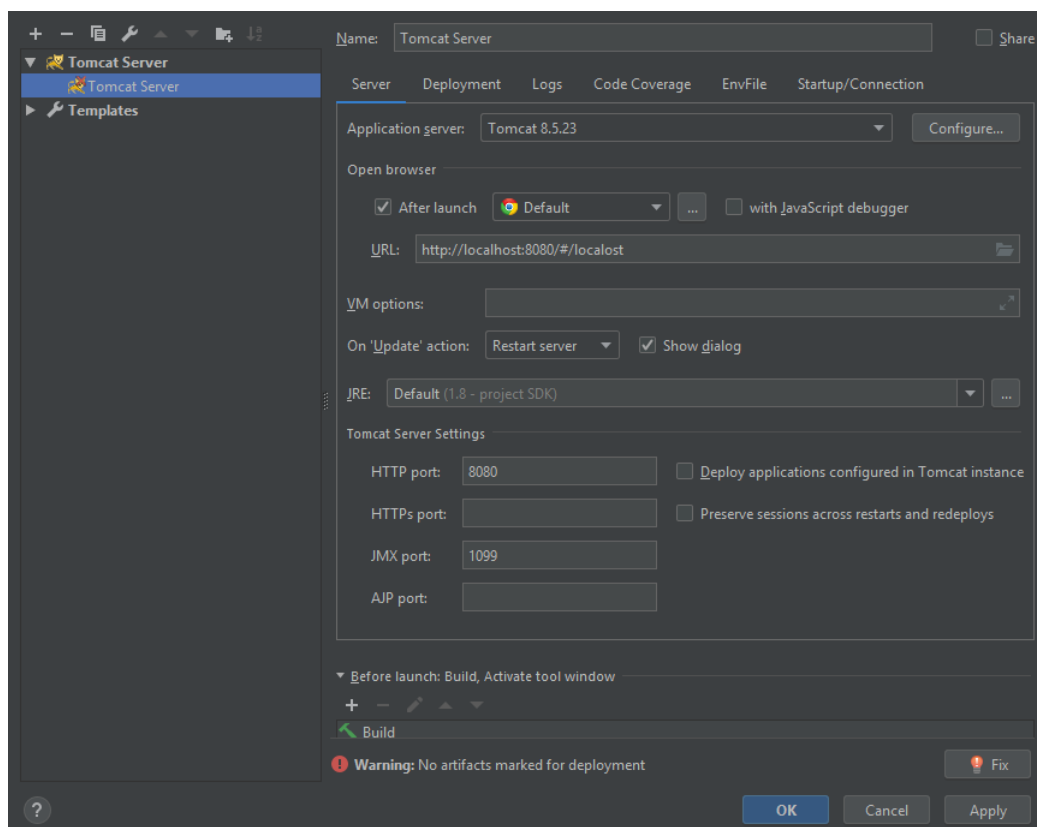
- JDK 8
- Git
- NodeJS v6 nebo novější
- Maven
- Apache Tomcat (případně jiný aplikační server)

B.2 Instalace a příprava prostředí

Zdrojový kód aplikace najdete na CD příloze nebo na adrese <https://gitlab.fel.cvut.cz/vostrja6/room-reservation-system>. Poté pokračujte podle tohoto návodu:

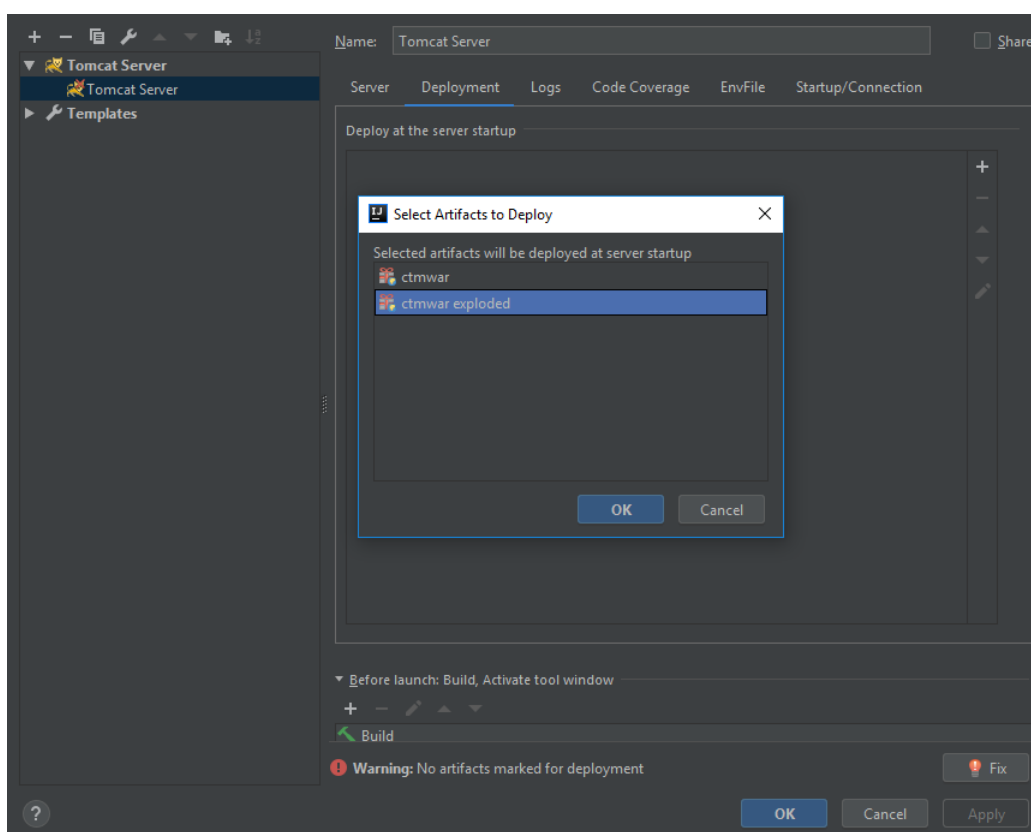
1. Naklonujte repozitář do libovolné složky [`git clone git@gitlab.fel.cvut.cz:vostrja6/room-reservation-system.git`].

2. V naklonované složce jděte do složky webapp [cd src/main/webapp]
3. Spustě instalaci příkazem [npm run build].
4. Následně zadejte příkaz [npm start].
5. Nyní máme připravený Node.js prostředí a musíme vytvořit server.
6. Pokud používáte vývojové prostředí IntelliJ klikněte na „Edit Configurations“ a přidejte server „Add Tomcat Server Local“.



Obrázek B.1: Aplikace: Tomcat Server

7. Nastavte server (viz. obr. B.1).



Obrázek B.2: Aplikace: Tomcat Deployment

8. V záložce „Deployment“ zvolte artefakt (viz. obr.B.2) z Gitu, případně vytvořte nový.

B.3 Snímky aplikace

Na obrázcích (obr. B3-B9) můžeme vidět snímky z aplikace pro rezervaci místností.

RRS - Login

Username

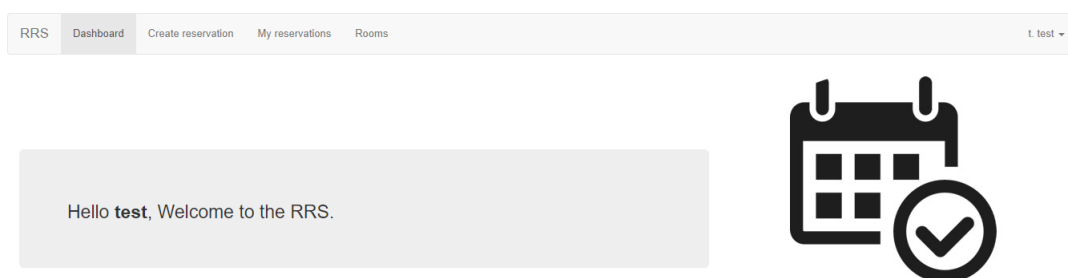
Password

Obrázek B.3: Aplikace: Přihlášení

RRS - Registration

First name *	<input type="text"/>	Last name *	<input type="text"/>
Email *	<input type="text"/>	Phone number *	<input type="text"/>
ID number *	<input type="text"/>		
Adress			
Street *	<input type="text"/>	House number *	<input type="text"/>
City *	<input type="text"/>	ZIP code *	<input type="text"/>
Password			
Password *	<input type="text"/>	Password confirm *	<input type="text"/>

Obrázek B.4: Aplikace: Registrace



Obrázek B.5: Aplikace: Hlavní stránka

RRS Dashboard Create reservation My reservations Rooms t. test ▾

Reservation					Actions
Date	Time from	Time to	Status	Room	
05-06-19	15:30	16:00	Created		Delete
01-06-19	16:00	16:30	Created		Delete
09-06-19	12:00	12:30	Created		Delete
Previous					Next
Page		1	of 1	5 rows ▾	

RSS, 2019 - Jan Vostrý

Obrázek B.6: Aplikace: Moje rezervace

RRS Dashboard Create reservation My reservations Rooms t. test ▾

Room			Info
Building	Block	Number	
T2	C3	305	Show
T2	A4	203	Show
Previous			Next
Page		1	of 1
		10 rows ▾	

RSS, 2019 - Jan Vostrý

Obrázek B.7: Aplikace: Místnosti

B. Aplikace: Rezervační systém

Room reservations												
Date	9:00	9:30	10:00	10:30	11:00	11:30	12:00	12:30	13:00	13:30	14:00	14:30
05-06-19	-	X	-	-	-	-	-	-	-	-	-	-
01-06-19	-	-	-	-	-	-	-	-	-	-	-	-

RSS, 2019 - Jan Vostrý

Obrázek B.8: Aplikace: Místnost

Reservation form

Time *
10:00

Date *
05-06-19

Room *
T2 C3-305

Find reservation Create reservation

Room is not reserved for this date and time. You can create reservation

RSS, 2019 - Jan Vostrý

Obrázek B.9: Aplikace: Rezervace