

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Návrh, realizace a marketingová strategie webové aplikace
Diplomová práce

Autor: Denisa Andrlová
Studijní obor: Informační management

Vedoucí práce: Ing. Karel Mls, Ph.D.

Prohlášení:

Prohlašuji, že jsem diplomovou práci zpracovala samostatně a s použitím uvedené literatury.

V Hradci Králové dne 28.4.2017

Denisa Andrlová

Poděkování

Ráda bych touto cestou vyjádřila poděkování především mému vedoucímu diplomové práce, panu Ing. Karlu Mlsovi, Ph.D., za jeho cenné rady, ochotu a trpělivost.

Dále bych ráda poděkovala panu Ing. Filipu Yaghobovi a MUDr. Vojtěchu Novákovi, kteří mi poskytli praktické konzultace a doporučení z jejich odborné praxe.

V neposlední řadě děkuji také své rodině, která mi poskytla podporu při studiu a při psaní této diplomové práce.

Anotace

Diplomová práce na téma Návrh, realizace a marketingová strategie webové aplikace se v teoretické části věnuje životnímu cyklu vývoje software, základním modelům a metodikám vývoje aplikací. Dále se zaměřuje na návrh software v jazyku UML. V poslední části se dotýká marketingové strategie produktu, zaměřuje se na marketingový a komunikační mix včetně analýzy prostředí. V praktické části se diplomová práce zabývá návrhem webové aplikace pro řízení projektu malého rozsahu. Je sepsán Implementační projekt s návrhem základních UML diagramů. Součástí je také low-fidelity prototype aplikace, tedy základní grafický návrh. V rámci marketingové strategie se diplomová práce věnuje názvu, logu, marketingovému a komunikačnímu mixu navržené webové aplikace.

Klíčová slova:

webová aplikace, životní cyklus vývoje software, modely vývoje software, agilní metodiky, UML, marketingový mix, komunikační mix, analýza prostředí, grafický návrh, user experience.

Annotation

Title: Design, implementation and marketing strategy of the web application

The Diploma thesis deals with Design, implementation and marketing strategy of the web applications. The theoretical part include of the development software life cycle, basic models and methodology of application development and software design in UML. In the last part we can find the marketing strategy of the product Strategy is focus on marketing and communication mix, and include on environmental analysis. The practical part the Diploma thesis include the design of small-scale web application. An implementation project is write with basic UML diagrams. Thesis include a prototype with low fidelity and also basic graphic design. The marketing strategy has a lot of parts, they are the name, logo, marketing and communication mix of the proposed web application.

Keywords:

web application life cycle software development, software development models, agile methodologies, UML, marketing mix, communication mix, environmental analysis, graphic design, user experience.

Obsah

Úvod	1
Teoretická část	2
1. Životní cyklus vývoje software	2
1.1 Základní modely životního cyklu systému.....	3
1.1.1 Vodopádový model	3
1.1.2 Spirálový model.....	5
1.1.3 Metodika Rational Unified Process (RUP).....	6
1.2 Agilní metodiky.....	7
1.2.1 Stručný přehled nejznámějších agilních metodik.....	9
1.3 Metodiky vývoje webových aplikací.....	13
1.3.1 Metodika Jennifer Fleming.....	13
1.3.2 Metodika WebWAVE Development Process.....	15
1.3.3 WebWAVE Ongoing Development Process.....	15
1.4 UML	17
1.4.1 Stavební bloky jazyka UML.....	17
1.4.2 Diagramy v jazyce UML	18
1.4.3 USE Case Diagram	21
1.4.4 Class Diagram	26
1.5 Webdesign a User experience.....	32
1.5.1 Webdesign.....	32
1.5.2 User Experience.....	34
1.6 Marketingová strategie	35
1.6.1 Marketingový mix.....	35
1.6.2 Marketingový komunikační mix	39
1.6.3 Analýza prostředí	43
1.6.4 Cílený marketing.....	46
Praktická část	48
2. Implementační projekt webové aplikace	48
2.1 GC – Group Cooperation	48
2.2 Požadavky na aplikaci.....	49
2.2.1 Funkční požadavky	49
2.2.2 Non-Funkční požadavky	53
2.3 Návrh systému.....	54

2.3.1 Use Case Diagram.....	54
2.3.2 Class model.....	56
2.3.3 Datový model	58
2.4 Návrh uživatelského rozhraní – GUI.....	59
2.4.1 Přihlášení.....	59
2.4.2 Homepage aplikace	60
2.4.3 Nástěnka.....	62
2.4.4 Úkoly.....	64
2.4.5 Zprávy	67
2.4.6 Soubory	68
2.4.7 Nastavení.....	69
2.5 Práva a role.....	70
3. Marketingový mix.....	71
3.1. Produkt.....	71
3.2 Cena.....	73
3.3 Distribuce	73
3.4 Propagace.....	73
4. Marketingový komunikační mix	74
5. Analýza prostředí	75
5.1 Porterův model pěti konkurenčních sil	75
5.2 SWOT analýza.....	79
6. Shrnutí a závěr.....	80
Seznam obrázků	81
Seznam příloh.....	83
Seznam použité literatury	84

Úvod

Tématem Diplomové práce je Návrh, realizace a marketingová strategie webové aplikace. Cílem diplomové práce je navrhnout aplikaci pro podporu řízení méně rozsáhlých projektů, tedy např. pro malé neziskové organizace, startupy, ale také tým, který pracuje na vysokoškolském semestrálním projektu, aj. A právě u těchto projektů vznikla myšlenka takové aplikace. Ve většině skupinových projektů, kterých jsem byla na vysoké škole součástí, probíhalo řízení a komunikace zejména přes sociální sítě, což se dříve nebo později ukázalo jako zcela neefektivní z mnoha hledisek. Avšak studenti viděli tuto komunikaci za rychlou, přístupnou a zcela zdarma. Většina dostupných aplikací na projektové řízení je zpoplatněná nemalými částkami. Cílem této diplomové práce je navrhnout aplikaci pro podporu malých projektových týmů, která pro ně bude zcela zdarma a zároveň bude disponovat veškerými funkcemi, které zajistí efektivní řízení projektu.

Do problematiky práce uvádí nejdříve v teoretické rovině. Věnuje se životnímu cyklu vývoje softwaru, základním modelům a metodikám vývoje aplikací. Dále se zaměřuje na návrh software v jazyku UML. V poslední části se dotýká marketingové strategie produktu, zaměřuje se na marketingový a komunikační mix včetně analýzy prostředí.

V praktické rovině se věnuje návrhu zmíněné aplikace. Je sepsán Implementační projekt s návrhem základních UML diagramů. Součástí je také low-fidelity prototype aplikace, tedy základní grafický návrh. V rámci marketingové strategie se diplomová práce věnuje názvu, logu, marketingovému a komunikačnímu mixu s cílem definovat aktivity, které by měly aplikaci dostat do povědomí cílové skupiny.

Teoretická část

1. Životní cyklus vývoje software

Softwarové inženýrství

„Softwarové inženýrství je technická disciplína, která se zabývá všemi aspekty produkce softwaru od počátečních fází specifikace systému až po údržbu systému, který se již používá.“ [22, str. 20]

Každý software prochází několika vývojovými fázemi. Mezi základní fáze patří analýza, návrh, implementace, testování a zavádění, provoz a údržba. Přesný obsah, počet jednotlivých vývojových fází a jejich pořadí se může lišit podle metodiky vývoje software.

Analýza

Analýza neboli specifikace požadavků je základ celkového vývoje požadovaného software. Cílem je získání požadavků, cílů a funkcí systému. Výsledkem jsou odpovědi na následující otázky:

- účel systému,
- požadavky uživatelů,
- rozsah funkčnosti,
- zdroje nutné k řešení – technické a softwarové zajištění,
- analýza současného stavu,
- časový plán projektu,
- náklady na tvorbu systému,
- analýza rizik.

Na konci analýzy máme tedy představu o vyvíjeném produktu – pro koho tento software je a jakým způsobem se s ním bude pracovat.

Návrh

Definice požadavků a návrh systému až na úroveň, kdy je možné začít implementovat.

Detailně odpovídá na předešlé otázky kladené v první fázi a rozšiřuje je o další:

- Globální návrh IS – logický datový model, který udává funkce a data systému bez ohledu na technologické řešení.
- Detailní analýza – fyzický datový model, obsahuje funkční analýzu systému, datovou analýzu s ohledem na technologické řešení.
- SW a HW požadavky související s nasazením produktu.

Výsledkem analýzy a návrhu jsou dokumenty sloužící jako obsah smlouvy mezi zadavatelem softwaru a externí firmou vyvíjející daný IS (informační systém).

Implementace

Implementace je samotné programování, tedy tvoří se kód systému dle informací shromážděných v předešlých etapách. Výsledkem implementace musí být systém, který obsahuje všechny požadavky zákazníka. V rámci této fáze se vytváří uživatelská a provozní dokumentace včetně popisů pracovních procesů.

Testování

Cílem testování je nalezení chyb v softwaru a následné opravení. Základním dokumentem je Testovací plán, který obsahuje testovací případy, cíl jednotlivých testů, postup testování, podmínky, za jakých testování probíhalo, a harmonogram. Testovací případy definují vstupní data a očekávaná výstupní data. Na základě porovnání výstupních dat s očekávanými daty stanovíme výsledek testu. Po skončení testů se vše zapisuje do testovacího plánu včetně výsledků a následně dochází k nápravě chyb, tzv. bugů.

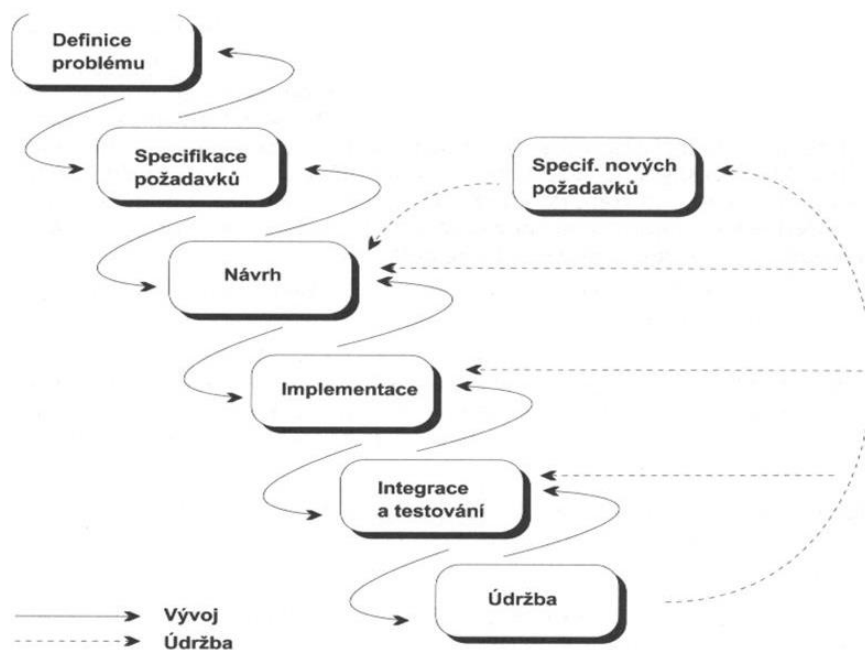
Zavádění a provoz

Zavádění je fáze životního cyklu, která obnáší samotnou instalaci systému a jeho zavedení do provozu včetně školení uživatelů. Během zkušebního provozu musí být poskytovatel připraven zajistit okamžitý servis při odstraňování chyb nalezených během provozu a dořešit dodatečné požadavky. Při ostrém provozu by měl být poskytovatel schopný zajistit údržbu a případný další rozvoj IS.

1.1 Základní modely životního cyklu systému

1.1.1 Vodopádový model

Vodopádový model patří k nejstarším modelům životního cyklu vývoje softwaru, který se používal už v 70. letech 20. století. Vodopádový model je charakterizován jako model, kde jednotlivé fáze životního cyklu vývoje IS se provádí postupně, navazují na sebe a vzájemně se neprotínají. K další fázi vývoje softwaru se postupuje vždy až po skončení dané fáze, není zde možné vyvíjet více fází IS najednou. Jedná se tedy o sekvenčnost jednotlivých fází [1]. Fáze vodopádového modelu můžete vidět na obrázku č.1.



Obr. 1 Vodopádový model [1]

Jsou zde pevně dány jednotlivé kroky a fáze. Nesetkáme se zde s iteracemi, nevytváří se demoverze ani se průběžně nekomunikuje se zákazníkem. Právě absence komunikace se zákazníkem a nedostatečná flexibilita modelu způsobila, že v dnešní době je tento přístup zejména pro velké projekty nedostatečný.

Mezi výhody a silné stránky vodopádového modelu životního cyklu patří zejména:

- první model, který určil posloupnost fází při vývoji softwaru,
- jednoduchost – v jednotlivých fázích je jasné, co se má dělat,
- vývoj postupuje systematicky [1].

Mezi nevýhody a slabé stránky vodopádového modelu životního cyklu patří zejména:

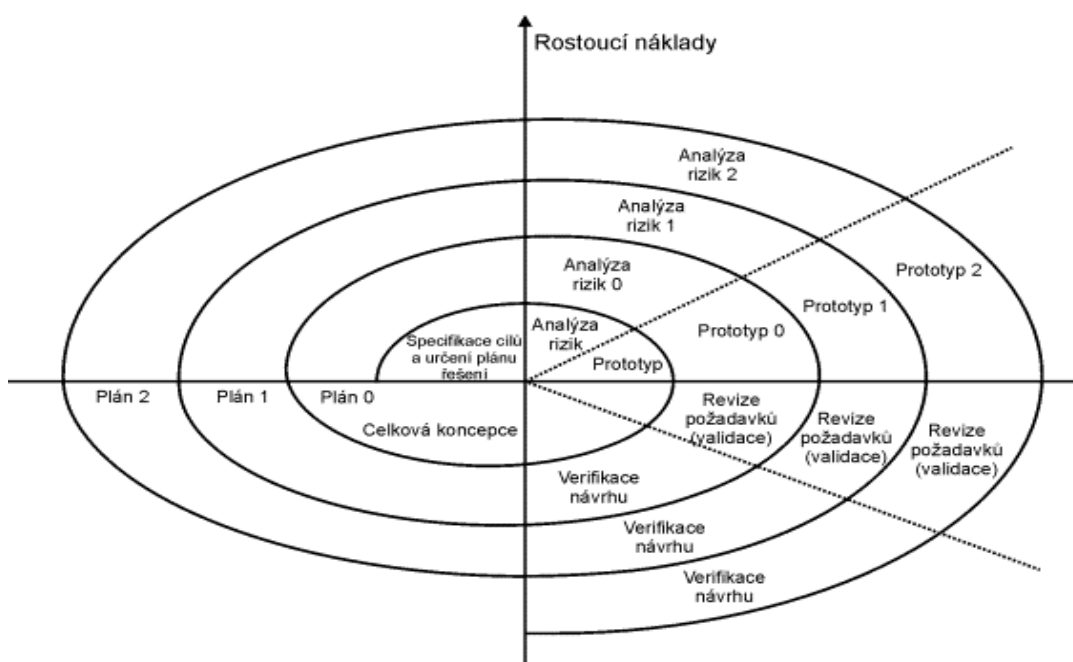
- příliš jednoduché na to, aby mohl být aplikován na rozsáhlé projekty,
- nepružnost – vše probíhá dle pevně stanovených fází,
- absence komunikace se zákazníkem:
 - mezi analýzou a předáním se mohou požadavky zákazníka měnit,
 - zákazník nemůže žádným způsobem ovlivňovat vývoj aplikace,
 - zákazník necítí žádnou sounáležitost,
 - pokud analytik špatně pochopí požadavky zákazníka, vyjde tato skutečnost najevo až při předání [1].

Vodopádový model je tedy vhodné použít pro vývoj systému, kde jsou přesně dány požadavky a způsob řešení, což jsou zejména dostatečně strukturované malé projekty. V takových případech může být tento postup poměrně rychlý a levný.

Naproti tomu je použití vodopádového modelu zcela nevhodné pro vývoj rozsáhlého systému, zejména proto, že se zákazníkem komunikujeme pouze v první fázi, kdy zjišťujeme jeho požadavky na systém, a v poslední, kdy mu předáváme hotový IS. Mohou tak tedy vzniknout velké nedostatky v tom, co zákazník chtěl a v tom, co opravdu potřebuje, což zjišťuje až ve fázi předání.

1.1.2 Spirálový model

Spirálový model byl navržen Barry Bohmem v roce 1985. Model se snaží eliminovat některé nedostatky vodopádového modelu a zavádí kromě jiného i další klíčové koncepty: iterativní přístup a důslednou analýzu rizik. Základním faktorem celého modelu jsou neustále se opakující jednotlivé fáze vývoje. V každém dalším kroku se zpřesňují požadavky a uvažují nová rizika [1].



Obr. 2 Spirálový model [1]

Grafické znázornění spirálového modelu vidíme na obrázku 2 – probíhá ve spirále od středu směrem ven. Spirála je rozdělena do 4 kvadrantů:

- stanovení cílů – stanovení cílů, alternativ a omezujících podmínek dalšího vývoje,
- analýza rizik – analýza směrem k následné iteraci,
- vývoj, testování a ověřování – revize požadavků, validace, verifikace návrh,
- plánování – forma plánování dalšího postupu.

Mezi výhody spirálového modelu patří zejména:

- znovupoužitelnost komponent,
- včasné vyloučení nevhodných řešení – díky rizikovým analýzám, které se provádí, se redukuje pravděpodobnost nesprávného postupu v každé iteraci,
- umožňuje konzultaci se zákazníkem. [1]

Mezi nevýhody spirálového modelu patří zejména:

- komplikovanost – mnoho byrokratických povinností, které snižují efektivitu a zpomalují vývoj,
- změny jsou možné až po dokončení cyklu,
- neumožňuje přesné naplánování termínů. [1]

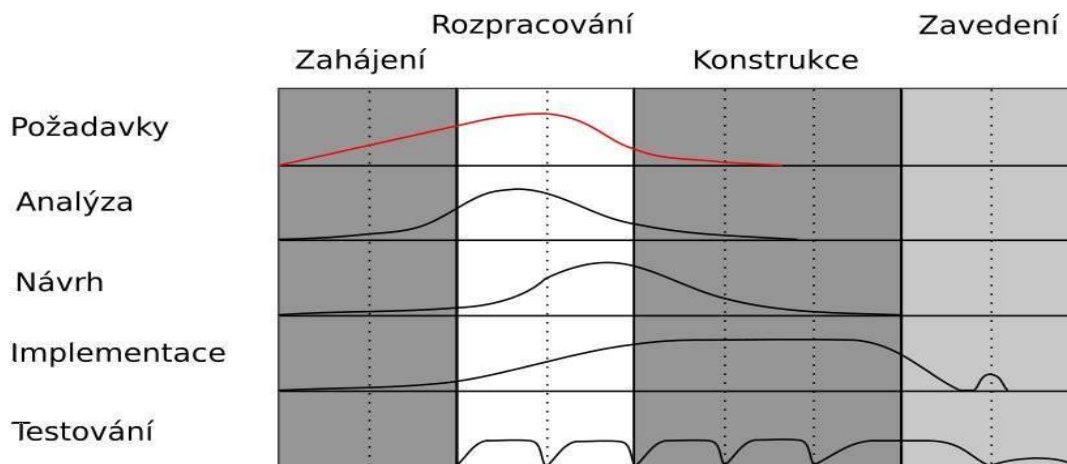
Spirálový model životního cyklu se díky své komplexnosti hodí pro rozsáhlejší projekty. Malé projekty může zahltit mnoha byrokratickými požadavky a může se stát neefektivním. Hodí se pro projekty, u kterých můžeme předpokládat časté změny a je vhodný, když zákazník souhlasí s častou iterací.

1.1.3 Metodika Rational Unified Process (RUP)

RUP na rozdíl od předchozích modelů není pouze konceptem, ale je reálně použitelný na řízení projektů. Metodika je vyvinuta a dodávána firmou Rational Software.

RUP pracuje se čtyřmi základními fázemi vývoje, které se vyskytují v několika iteracích. Pro postup do další iterace musí být splněna veškerá kritéria, určená před jejím začátkem.

„Rational Unified Process patří do skupiny tzv. přístupů řízených případy užití (use-case-driven approach). Vývoj prostřednictvím RUP probíhá v iteracích a každá iterace se člení na čtyři fáze: Zahájení, Rozpracování, Konstrukce, Zavedení [6].



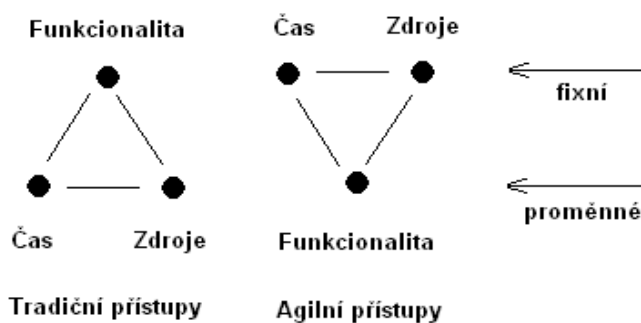
Obr. 3 Rational Unified Process [6]

Mezi hlavní výhody RUP patří jeho obecnost a komplexnost, která umožňuje aplikovat RUP do široké škály projektů. Používání iterací zabezpečuje včasné odhalení rizik, snazší úpravu změn a umožňuje znovupoužitelnost a dobrou kontrolu nad projektem. Mezi další silné stránky metody RUP patří provázanost s notací UML.

Metodika RUP je uplatnitelná především ve velkých firmách, které ji mohou využít pro rozsáhlé a náročné projekty. RUP je komerčním produktem, proto je pro většinu malých firem značně nedostupný.

1.2 Agilní metodiky

Rozdíly mezi tradičními přístupy a agilními metodikami vývoje software můžeme vidět na obr. 4. Mezi základní rozdíly patří pohled na dané proměnné.



Obr. 4 Agilní metodiky vs. tradiční přístupy [1]

Tradiční přístupy považují funkcionalitu za fixní, je jasně popsána v požadavcích a hlavním cílem je její naplnění. Proměnné jsou čas a zdroje, protože je při tradičních přístupech těžké odhadovat, kolik to bude stát a jak dlouho to bude trvat. Agilní metodiky

považují za fixní naopak čas a zdroje za proměnnou funkcionalitu. Na začátku je jasně dána nejvyšší možná cena a nejdelší možný čas a mění se funkcionalita [1]. Aplikování agilních přístupů je nejvhodnější za předpokladu, že zadání je nejasné a mohou nastávat změny.

Základem agilního přístupu je manifest z roku 2001 – Agile Manifesto, který vznikl na konferenci v Utahu, kde se sešli přední světový vývojáři a experti na softwarové inženýrství.

Manifest vychází ze dvou základních tezí:

1. Přijmout a umožnit změnu je mnohem efektivnější, než pokoušet se jí zabránit.
2. Je třeba být připraven reagovat na nepředvídatelné události, neboť ty bezpochyby nastanou: tento bod lze formulovat také tak, že jedinou jistotou je změna [1, str. 120].

„Objevujeme lepší způsoby vývoje software tím, že jej tvoříme a pomáháme při jeho tvorbě ostatním. Při této práci jsme dospěli k těmto hodnotám:

- **jednotlivci a interakce** před procesy a nástroji,
- **fungující software** před vyčerpávající dokumentací,
- **spolupráce se zákazníkem** před vyjednáváním o smlouvě,
- **reagování na změny** před dodržováním plánu.

Jakkoliv jsou body napravo hodnotné, bodů nalevo si ceníme více.“ [2, Manifest agilního vývoje]

Z předešlých tezí a hodnot, které Manifest upřednostňoval, lze vyvodit dvanáct základních hodnot agilních metodik [3]:

- včasná a kontinuální dodávka hodnotného softwaru,
- změny jsou žádané,
- časté dodávky fungujícího software, zdůrazňuje kratší iterace,
- spolupráce se zákazníkem po celou dobu vývoje,
- motivace lidí,
- vzájemná konverzace,
- za úspěch se považuje fungující produkt,
- udržitelný rozvoj,
- důraz na kvalitu návrhu,

- jednoduché postupy,
- důvěra, komunikace a kreativita,
- snaha zefektivnit práci.

1.2.1 Stručný přehled nejznámějších agilních metodik

Počet metodik, které vycházejí z Manifestu agilního vývoje, v současné době stále roste. Nyní jsou uvedeny ty nejznámější z nich spolu s jejich základními charakteristikami.

Extrémní programování – Extreme Programming, XP

Extrémní programování patří mezi nejznámější zástupce agilních metodik. Tato metodika je dobře aplikovatelná pro malé a střední týmy, které vyvíjejí software podle zadání, které se často mění nebo je nejasné. Extrémní programování vychází z přesvědčení, že jediným exaktním, jednoznačným, změřitelným, ověřitelným a nezpochybnitelným zdrojem informací je zdrojový kód. XP používá běžně známé postupy a principy, které dotahuje do extrémů [1].

Extrémní programování klade důraz na týmovou práci. Programátoři stále komunikují se svými zákazníky a kolegy. XP pracuje na základě čtyř hodnot – komunikace, jednoduchost, zpětná vazba, odvaha a respekt. Při vývoji dle zásad XP existují čtyři základní činnosti [5]:

- **Testování** – probíhá průběžně, pracuje se s automatizovanými testy, které se na jednotlivé moduly připravují ještě před samotným vývojem daného modulu.
- **Psaní zdrojového kódu** – zdrojový text je cílem.
- **Poslouchání** – dochází ke stálé komunikaci mezi zákazníky i kolegy.
- **Navrhování (design)** – vytváření struktury, tak aby potřeba změny v jedné části systému nevyvolala potřebu změny v ostatních částech.

SCRUM Development Process

Metodika SCRUM patří mezi nejčastěji aplikované agilní metodiky pro vývoj software. Vývoj probíhá v tzv. iteracích, můžeme se také setkat s pojmem sprint. Iterace (sprint) může trvat cca 2–4 týdny, kde výsledkem každé iterace je funkční prototyp. Základní charakteristikou SCRUMU jsou tzv. SCRUM meetings, každodenní schůzky vývojového týmu, kde dochází ke shrnutí dosavadního vývoje a následné určení dalšího postupu.

Porady vedou k mnohem pružnějšímu a flexibilnějšímu vývoji software, a zároveň k větší soudržnosti vývojového týmu.

SCRUM představuje čtyři vývojové fáze:

- **Plánování** – definice rozsahu aktuální verze, určení harmonogramu a stanovení tzv. backlog – funkce, vlastnosti, úkoly, které je potřeba realizovat.
- **Architektura a design** – změna stávající podoby na základě nových požadavků, rizik, analýz atd.
- **Vývoj** – probíhá v iteracích, výsledkem každé iterace je předání funkčního prototypu zákazníkovi.
- **Uzavření** – pokud se vývojový tým se zákazníkem shodne, že aktuální verze je plně dostačující, nepokračuje se dalším vývojem (sprintem), ale produkt se přesune do finální fáze tak, aby mohl být předán zákazníkovi včetně dokumentace atd.

Lean Development

Lean Development určuje deset klíčových pravidel, jejichž dodržování slibuje efektivnější a kvalitnější vývojový proces [1, str. 160]

1. Odstranit vše, co je zbytečné.
2. Minimalizovat zásoby (meziprodukty).
3. Maximalizovat tok (zkrátit vývojový čas).
4. Vývoj je tažen poptávkou (rozhodnutí na poslední chvíli).
5. Pracovníci mají pravomoci rozhodovat (rozhodování i na nejnižších úrovních).
6. Hlavním cílem je uspokojovat požadavky zákazníků (a to nejen teď, ale i v budoucnu).
7. Zavést zpětnou vazbu (a nebát se učiněných rozhodnutí).
8. Odstranit lokální optimalizace (neustálé optimalizace stávajícího řešení nedávají smysl).
9. Vybudovat partnerství s dodavateli (využívat subdodávek a předpřipravených komponent).
10. Vybudovat kulturu pro možnost neustálého zlepšování.

V rámci metodiky Lean Development se setkáváme i se sedmi principy vzešlými na základě výše uvedených deseti pravidel [1]:

- **Eliminace plýtvání** – určení zdrojů plýtvání a snaha o jejich eliminaci, identifikace důležitých zdrojů pro vývoj.
- **Rozvinout učení** – ověřování fungování experimentem.
- **Rozhodovat co nejpozději** – udržení otevřené mysli a rozhodování se v pravou chvíli.
- **Rychlé a časté dodávky** – tvoření meziproduktů, nechat rozhodovat zákazníky (možnost specifikovat požadavky).
- **Pravomocní pracovníci** – jasně daná hierarchie pravomocí, rozdělení vedení – vede k motivaci lidí.
- **Integrita** – integrita daného softwaru, vývojového procesu, důraz na testování.
- **Vidět celek** – odstranění lokálních optimalizací.

Feature-Driven Development, FDD

Feature-Driven Development je agilní metodika vývoje software, která se zaměřuje zejména na vlastnosti výsledného produktu.

Podobně jako většina agilních metodik je FDD zaměřena na iterativní vývoj a krátké iterace. Každá iterace trvá 1–3 týdny a na konci je k dispozici vždy betaverze, která se dodává zákazníkovi. Zákazník je tak dostatečně informován o průběhu a vlastnostech daného produktu a může se podílet na vývoji a přizpůsobovat ho požadovaným směrem. Nemůže se tedy stát, že zákazník nebude s výsledkem daného produktu spokojený, protože původně chtěl něco jiného.

Jako vlastnost definuje FDD velice malou část funkčnosti důležitou a užitečnou z pohledu zákazníka. Charakteristiky vlastností jsou zejména [1]:

- **měřitelnost** (implementovaná vlastnost musí být stejná jako vlastnost požadovaná zákazníkem),
- **srozumitelnost** (musíme vědět, co má být výsledkem dané vlastnosti),
- **realizovatelnost** (musíme mít jistotu, že daná vlastnost je realizovatelná a doba potřebná pro její realizaci není dlouhá).

Jednou z nejdůležitějších činností při vývoji dle metodiky FDD je modelování a dobrý objektový model se považuje za základní východisko při vývoji. Dle metodiky FDD je důležitější model než samotný kód. Nejmenší základní prvek vývoje je vlastnost, což

můžeme považovat za hlavní odlišnost od ostatních agilních metod, kde je to většinou iterace.

Metoda FDD popisuje 5 vývojových fází, které znázorňuje následující obrázek č. 5.



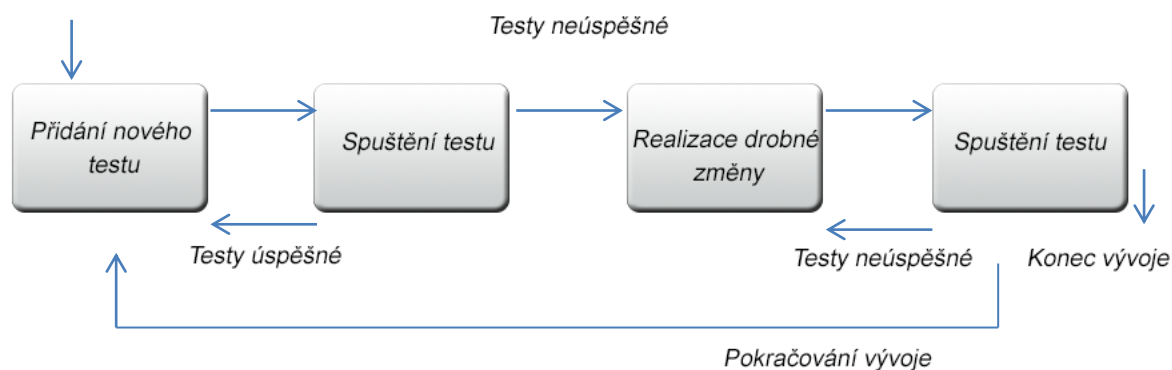
Obr. 5 Vývojové fáze FDD [1]

Metodika FDD je vhodná spíše pro menší projekty, kde je možné zpracovat soubor požadovaných vlastností.

Test-Driven Development, TDD

Metoda Test-Driven Development se nezabývá primárně specifikací, dokumentací, plány, ale jako pomyslný vrchol pyramidy poslušnosti kroků při vývoji software staví testování. Základním pravidlem TDD je napsání testu ještě před napsáním kódu. Řídí se pravidlem: „Implementujeme přesně takové množství programového kódu, kolik dokáže projít testem.“ [1, str. 198].

Jako první krok v rámci TDD je tedy psaní testovacího kódu. TDD si klade za cíl napsat napoprvé test, který selže. Teprve potom můžeme implementovat požadovanou testovanou funkci. Teprve když funkce projde daným testem, je vývoj dané funkcionality u konce.



Obr. 6 Vývojové fáze TDD [1]

TDD při vývoji software postupuje od testování přes implementaci k návrhu na rozdíl od většiny metodik, které postupují přesně naopak.

1.3 Metodiky vývoje webových aplikací

Při vývoji webových aplikací lze uplatnit klasické metodiky vývoje IS uvedené v předchozích kapitolách. Vývoj webových aplikací má však svá specifika a odlišnosti, a proto vzniklo několik metodik za účelem podpory vývoje webových aplikací, kterým se budeme věnovat v následující kapitole.

Mezi hlavní odlišnosti můžeme zahrnout:

- **Složení vývojového týmu** – objevují se i netradiční profese jako grafik, copywriter, specialista online marketingu aj.
- **Vývoj a modernizace** – u webových aplikací je předpokladem, že se budou dále vyvíjet, upravovat a zlepšovat.
- **Rozdíly v testování** – testování layoutu, grafického návrhu, barev, rozlišení aj. (existují oblasti, které se musí testovat manuálně, uživatelsky).
- **Multiplatformní řešení** – pro webovou aplikaci je základem multiplatformní řešení (jedna aplikace musí podporovat několik zařízení, prohlížečů aj.).
- **Uživatelské rozhraní** – kreativní návrh uživatelského prostředí, konzistence ovládacích prvků, důraz na formu textů.

Pro webové aplikace není na prvním místě počet uživatelů, ale primárně se soustřeďuje na počet tzv. věrných uživatelů, kteří se na web vrací a pravidelně je využívají.

1.3.1 Metodika Jennifer Fleming

Metodika Jennifer Fleming je založená na iterativním přístupu stejně jako agilní metodiky. Iterativní přístup je výsledkem dnešního moderního přístupu k vývoji aplikací. Zákazník chce být součástí vývoje a požaduje funkční meziprodukty, toho ale docílíme jen iterativním přístupem k práci.

Metodika Jennifer Fleming z těchto potřeb vychází a skládá se ze šesti základních fází [1]:

1. **Shromažďování informací** – „Cílem první fáze je získání dostatečného množství informací potřebných pro úspěšné dokončení projektu.“ [1, str. 248]. V této fázi se snažíme pochopit cíle a poslání projektu. Důležitá je v této fázi komunikace se zákazníkem jakožto zadavatelem projektu, analýza konkurenčních projektů, zjišťování novinek v dané problematice atd. Výsledkem dané fáze je získání přehledu o požadované webové aplikaci a předběžný plán, ze kterého plyne základní harmonogram práce, rozpočet, kapacita lidí a další nezbytné informace.
2. **Stanovení strategie** – „Hlavním cílem druhé fáze je určit, jak nejlépe realizovat cíle webu a jak naplnit očekávání uživatelů.“ [1, str. 249].
3. **Vytvoření prototypu** – důležitou fází při vývoji webových aplikací je vytvoření prototypu, protože zákazník často dokáže vyjádřit své požadavky, až když vidí reálný produkt a má prostor pro srovnání („I will know it when I see it.“). Pomocí prototypu lze získat zpětnou vazbu od zákazníka, která je důležitou součástí pro vytvoření finální architektury webové aplikace.
4. **Implementace** – na základě předchozích fází projektu je odsouhlasená strategie projektu, požadavky aplikace, grafický design aplikace a může se přejít k implementaci. Součástí implementace je editace obsahu webové prezentace, dokončení uživatelského rozhraní a samotné programování projektu.
5. **Uvedení do chodu** – vytvořením požadovaného projektu v předešlé části vývoj webové aplikace nekončí, následuje testování a aktivní marketing. Před veřejným spuštěním se aplikace musí důkladně otestovat – mimo základní funkce aplikace i funkčnost odkazů, konzistenci, přesnost, atd. Vzhledem k vývoji webové aplikace je důležité zabývat se i internetovým marketingem – reklamní bannery, SEO analýza, PPC reklamy, ...
6. **Údržba, správa a rozšiřování** – neustálá kontrola funkcionality, aktualizace obsahu, změny designu.

1.3.2 Metodika WebWAVE Development Process

Metodika WebWAVE Development obsahuje mnoho doporučení pro vývoj webových aplikací. Metodika se nezabývá pouze funkcionalitou, ale zaměřuje se i na vzhled, výkonnost, marketing atd.

Metodika obsahuje čtyři základní etapy:

1. **Fáze získání informací** – sběr informací pro vytvoření představy o tom co, jak, komu a proč má webová aplikace poskytovat.
2. **Fáze návrhu** – jedná se o vizuální a technický návrh webové aplikace.
3. **Fáze implementace** – implementace veškerých požadavků. Na konci implementace by měla být aplikace připravena ke spuštění.
4. **Fáze zprovoznění** – dochází ke spuštění webové aplikace a zároveň se zjišťuje zpětná vazba, pracuje se na marketingové propagaci, apod.

Každá etapa obsahuje určité množství kroků a každý krok je detailně realizován pomocí definovaných úkolů. Metodika WebWave Development Process definuje třináct základních pravidel, které mají zaručit požadovaný kvalitní výsledek. Mezi ně patří například: „Začni, až když víš, jak to má na konci vypadat.“, „Soustřed' se na maličkosti.“, „Soustřed' se na výběr technologie.“, „Základ úspěchu jsou pravidla a postupy.“ a další.

1.3.3 WebWAVE Ongoing Development Process

Metodika WebWAVE Ongoing Development (WODP) přímo navazuje na předchozí metodiku WebWAVE Development a zabývá se dalšími vývojovými fázemi vývoje webové aplikace, zejména na správu, modernizaci a údržbu webu.

Metodika definuje tři fáze:

1. **Vyhodnocování výkonnosti aplikace** – v rámci této fáze dochází k měření a vyhodnocování, které slouží jako podklady pro změny v dané webové aplikaci. Součástí je měření týkající se obchodního záměru (odpovídá na otázku „Proč to dělám?“, sleduje chování uživatelů na webu, požadavky na rozšíření, zpětnou vazbu od uživatelů, ...) a měření technických parametrů (doba odezvy, bezpečnost, dostupnost, ...). Výsledkem je vytvoření závěrečné zprávy obsahující

změřené hodnoty a výsledky. Zpráva je základem pro rozhodnutí provádějící se ve fázi údržby a modernizace webové aplikace.

2. **Modernizace webu** – zabývá se vylepšením, modernizací, rozšířením dané webové aplikace. Obsahuje dvě fáze: vylepšení obchodního výkonu (seřazení požadavků na zlepšení podle priorit, zahrnutí požadavků uživatelů, zabývání se standardy, uskutečnění požadovaných změn, ...) a vylepšení technického výkonu (zlepšení doby odezvy, výkonu, bezpečnosti, ...).
3. **Údržba webu** – obsahuje dvě fáze: kroky související se správou obsahu (získávání a udržování relevantních dat, kontrola odkazů, kontrola validity webu, správa databáze, ...) a kroky související s technickou správou (zálohování, optimalizace, správa uživatelských účtů, ...).

Metodika WODP se tedy nezaměřuje na vývoj webové aplikace, ale soustřeďuje se zejména na činnosti související s provozem a rozvojem aplikace.

1.4 UML

„UML (Unified Modeling Language, unifikovaný modelovací jazyk) je univerzální jazyk pro vizuální modelování systému.“ [6, str. 28]. UML slouží jako jazyk pro specifikaci, vizualizaci, konstrukci a dokumentaci při analýze a návrhu software.

1.4.1 Stavební bloky jazyka UML

Jazyk UML je reprezentován třemi základními stavebními bloky [6, 7]:

Předměty (Things) – prvky modelu

Předměty se v jazyce UML dělí na:

- Strukturní – představují podstatná jména v modelu (třídy, rozhraní, případy užití, ...).
- Chování – představují slovesa v modelu (chování v čase, stav, ...).
- Seskupení – rozdělení do balíčků podle významových prvků v modelu.
- Poznámky – vysvětlující část modelu (komponenty sloužící pro popis modelu).

Relace (Relationships)

Relace znázorňují vztahy mezi dvěma předměty, viz obrázek č.7.

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

Obr. 7 Relace [6]

Diagramy (Diagrams)

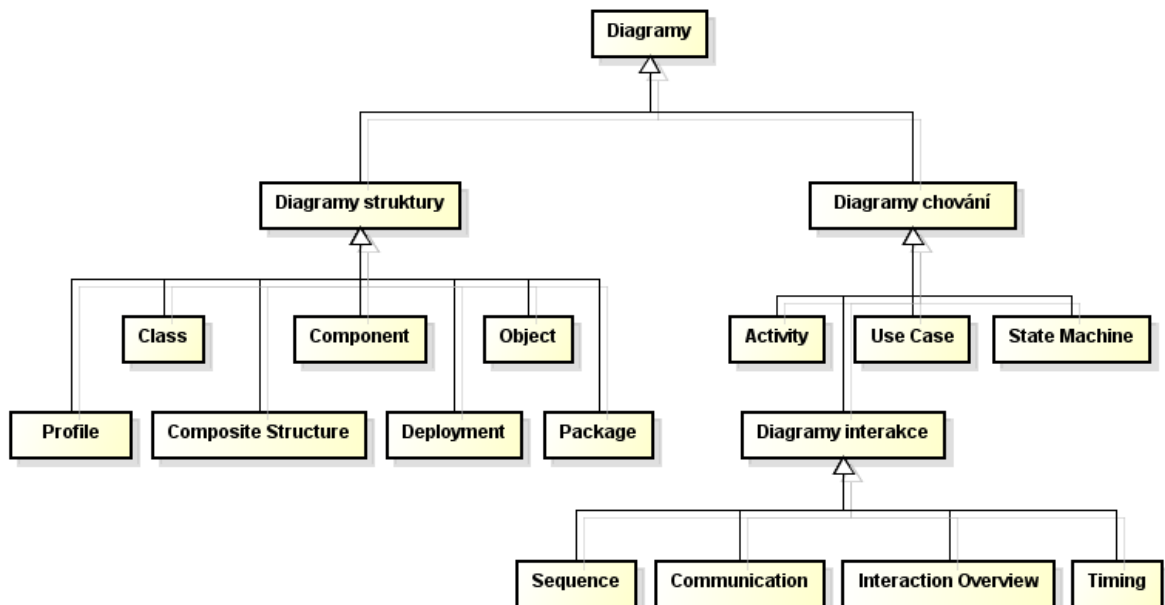
Vizualizace toho, co systém bude dělat a jak to bude dělat – pohledy na model. Diagramy se dělí na dvě základní skupiny [8]:

- **Diagramy struktury (Structure Diagrams)** – popisují strukturu systému.
- **Diagramy chování (Behaviour Diagram)** – popisují chování systému.

V diagramech chování ještě nalezneme samostatnou skupinu:

- **Diagramy interakce (Interaction Diagrams)** – popisují interakci mezi jednotlivými částmi systému.

UML využívá čtrnácti diagramů, které si v následující kapitole popíšeme.



Obr. 8 UML diagramy [8]

1.4.2 Diagramy v jazyce UML

Use Case Diagram

Use Case Diagram, v překladu také Diagram případů užití, popisuje funkcionalitu systému, tedy to, co uživatel od systému vyžaduje. Z diagramu je zřejmé, co má systém umět, ale neznázorňuje, jakým způsobem toho docílí. Use Case Diagram se skládá z případů užití, aktérů a vztahů mezi nimi. Zachycuje komunikaci mezi aktérem a systémem, znázorňuje vztahy mezi případy užití, a to ve vizuální i textové podobě. Je tak srozumitelný jak

pro vývojáře, tak i pro zákazníka. Tvorba Use Case Diagramu spadá do modelování na nejvyšší úrovni abstrakce [6, 8, 9].

Class Diagram

Class Diagram, v překladu také Diagram tříd, popisuje jednotlivé třídy a vztahy mezi nimi. Vychází z požadované funkcionality systému popsané v Use Case Diagramu. Class Diagram se skládá ze tříd, jejich atributů, metod a v neposlední řadě z asociací, které definují vztahy na další objekty. Diagram má dvě fáze: identifikace analytických tříd – definuje logické celky a vztahy bez ohledu na implementační prostředí – a identifikaci návrhových tříd – rozvádí analytické třídy do implementační podoby [6, 8, 9].

Sequence Diagram

Sequence Diagram se překládá jako Sekvenční diagram. „Sekvenční diagram vyjadřuje detailně posloupnost zpráv mezi objekty (instancemi) v čase.“ [10, str. 35]. Sekvenční diagramy mohou vycházet z diagramu případů užití, znázorňují tak časově uspořádanou posloupnost posílání zpráv mezi objekty v rámci jednoho případu užití. Jednotlivé instance jsou znázorněny v horní části diagramu a od každého vede svislá čára znázorňující běh času. Vodorovné čáry potom reprezentují posílané zprávy mezi objekty a jejich následnou interakci, inicializovanou uživatelem systému.

Communication Diagram

Communication Diagram, v překladu Diagram komunikace, je velice podobný Sekvenčnímu diagramu. Znázorňuje instance tříd a jejich vzájemné vztahy a tok zpráv [7]. Na rozdíl od sekvenčního diagramu nebere v úvahu čas, ale klade důraz na strukturální aspekty interakce a umožňuje libovolné rozložení instancí, tříd a pro znázornění pořadí využívá číslování [6].

Activity Diagram

Activity Diagram, v překladu Diagram aktivit nebo činností, modeluje proces jako aktivity a přechody mezi nimi. Lze je připojit k různým modelovaným prvkům, např. k případům užití, třídám, komponentám aj. S diagramem aktivit se můžeme setkat také u modelování obchodních procesů a pracovních postupů. Diagram aktivit se nejčastěji používá ve fázi analýzy [6].

State Diagram

State Diagram, v překladu Stavový diagram, podobně jako Diagram aktivit zachycuje dynamické chování systému. Prvky a účel diagramu jsou ale odlišné. Zatímco Activity Diagramy se používají nejčastěji k modelování obchodních procesů za účasti několika objektů, Stavový diagram se zaměřuje spíše na historii životního cyklu jednoho reaktivního objektu jako konečného stavového automatu. „Stavový diagram obsahuje předně jeden stavový automat pro jeden reaktivní objekt.“ [6, str. 428]. Stavový diagram se skládá z konečného počtu stavů a dále z událostí a přechodů. Stavový diagram lze použít k modelování dynamického chování reaktivních objektů, zejména u tříd, případů užití aj. [6].

Object Diagram

Object Diagram, v překladu Objektový diagram, používá stejnou syntaxi jako diagramy tříd, ale znázorňují objekty a jejich vztahy v určitém čase [7]. Diagram se využívá v počátečních fázích projektu pro lepší pochopení struktury informace [10].

Component Diagram

Component Diagram, v překladu Diagram komponent, se používá na úrovni technologického návrhu [10]. Využívá větší míry abstrakce než diagramy tříd. Komponenta je modulární část systému, která zapouzdřuje svůj obsah [7]. „Diagram komponent může zobrazovat komponenty, závislosti mezi nimi a způsob, jímž jsou klasifikátory přiřazovány ke komponentám.“ [6, str. 392].

Deployment Diagram

Deployment Diagram se překládá jako Diagram nasazení. „Diagram nasazení mapuje architekturu softwaru vytvořenou ve fázi návrhu na fyzickou architekturu systému, na němž bude software spouštěn.“ [6, str. 469]. S diagramem nasazení se můžeme setkat ve dvou vyhotoveních – diagram nasazení a diagram konkrétního nasazení. Diagram nasazení zobrazuje komponenty, uzly a znázorňuje vztahy mezi nimi. Uzly představují typ hardwaru a komponenty typ softwaru. Diagram konkrétního nasazení uzly a komponenty detailně konkretizuje – obsahuje instance uzlů a instance komponent [6].

Package Diagram

Package Diagram se překládá jako Diagram balíčků. „Balíček je mechanismem jazyka UML pro sdružování předmětů.“ [6, str. 232]. Znárodnuje skupiny elementů a závislosti mezi nimi. Diagram balíčků si můžeme představit jako základní organizační strukturu modelu [6, 10].

Timing Diagram

Timing Diagram, v překladu Diagram časování, znázorňuje změny stavů objektů souběžně v čase. Trvání jednoho stavu objektu je znázorněné vodorovnou čarou, její změna pak vyjadřuje změnu stavu objektu [10].

Composite Structure Diagram

Composite Structure Diagram se překládá jako Diagram vnitřní struktury nebo Diagram kompozitních struktur. Jedná se o doplňkový diagram, který je kombinací Object Diagramu a Class Diagramu [10].

Interaction Overview Diagram

Interaction Overview Diagram se překládá jako Diagram přehledu interakcí. Tento diagram se používá za předpokladu, že chceme vyjádřit určitou následnost mezi následnými diagramy – Sequence Diagram, Activity Diagram, Timing Diagram, Communication Diagram.

(Na tuto sekvenci zpráv reprezentovanou Sekvenčním diagramem navazuje tento průběh aktivit reprezentovaný Diagramem aktivit) [10].

1.4.3 USE Case Diagram

Use Case Diagram už jsme si stručně představili v předešlé kapitole a nyní si detailněji probereme jeho sestavení. Mezi první kroky spadající do tvorby softwaru je specifikace požadavků. Požadavky se dají rozdělit do dvou skupin – funkční a nefunkční.

„Funkční požadavek popisuje požadovanou funkci systému, požadavek nefunkční formuluje omezení kladená na systém nebo proces vývoje.“ [6, str. 80]. Příkladem funkčního požadavku může být jakákoli funkcionalita systému – možnost přihlásit se, objednat si, zarezervovat si, zaplatit aj. Naopak příkladem nefunkčních požadavků můžeme zařadit – výkon, dostupnost, zabezpečení softwaru aj.

Diagram případů užití se věnuje identifikaci funkčních požadavků na systém. Skládá se z několika kroků:

- nalezení hranic systému,
- vyhledání aktérů,
- nalezení případů užití,
- specifikace případů užití,
- určení alternativních scénářů,
- opakování předchozích kroků, dokud nedojde k jejich ustálení [6].

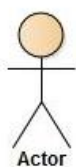
Use Case Diagram se skládá ze čtyř komponent – hranic systému, aktérů, případů užití a relací.

1.4.3.1 Hranice systému (System Boundary)

„Hranice systému odděluje systém od zbytku světa.“ [6, str. 92]. Stanovení hranic bývá často první věcí, co musíme při tvorbě softwaru udělat. Stanovuje se, co je součástí systému, resp. co je uvnitř systému a co je naopak za hranicemi systému. Stanovení hranic velice úzce souvisí s funkčními požadavky na systém. Hranice systému jsou graficky v UML2 znázorněny jako rámeček (ohraničení), kde vně stojí aktéři, tedy ti co systém používají, a uvnitř rámečku jsou znázorněny případy užití, které vyjadřují přínos systému pro aktéry [6].

1.4.3.2 Aktéři systému

Aktér (uživatel) systému je ten, kdo systém bezprostředně používá, potřebuje nebo s ním komunikuje. Aktéři jsou v UML2 externí entity stojící mimo hranice systému, které jsou znázorněny jako figurky s jejich názvem. „Aktér specifikuje roli, kterou určitá externí entita přijímá v okamžiku, kdy začíná daný systém bezprostředně používat.“ [6, str. 93].



Obr. 9 Aktér (Vlastní)

1.4.3.3 Případy užití

„Případ užití popisuje funkce, které systém poskytuje k užítku jednoho nebo více aktérů.“ [6, str. 95]. Při definování případů užití si musíme pokládat otázky typu: „Jak budou jednotliví aktéři systému používat?“ a „Co systém bude aktérům poskytovat?“. Součástí případů užití jsou posloupnosti činností, které nastávají při interakci systému s aktérem. Jednotlivé případy užití jsou vždy iniciovány aktérem. Případy užití jsou v UML2 graficky znázorněny jako elipsy, jejich název je uprostřed [6].



Obr. 10 Use Case (Vlastní)

Specifikace případů užití obsahuje následující informace [6]:

Název – Pro název případu užití se používají slovesa či slovesní fráze vždy zapsané tzv. VelbloudímZápisem – tedy bez mezer a každé slovo začíná velkým písmem, např. *ZaplatitObjednávku*.

Identifikátor – Názvy případů užití jsou v rámci modelu jedinečné; vzhledem k tomu, že je můžeme v průběhu času měnit, je dobré označit je také jedinečným číselným identifikátorem, tzv. ID případu.

Popis – Popisem systému je zpravidla odstavec textu popisující hlavní cíle případu užití.

Aktéři – Aktéři zapojení do realizace případu užití. Můžeme definovat dva typy aktérů – hlavní a vedlejší. Hlavní aktér spouští případ užití, kdežto vedlejší aktér se zapojuje do případu užití až po jeho spuštění.

Vstupní podmínky – Omezení týkající se stavu systému, která musí být splněna před jeho spuštěním.

Hlavní scénář – Popis jednotlivých kroků případů užití. Případ užití má jeden hlavní scénář a několik dalších vedlejších, neboli alternativních scénářů, které charakterizují chybové situace, které mohou při realizaci případu užití nastat.

K tzv. rozvětvení hlavního scénáře se používají klíčová slova Když (nová větev hlavního scénáře), Pro (opakování), Jakmile (modelování posloupností).

Např. 1. „Když zákazník zadá ... provede se krok 1.1.“; 2. „Když zákazník zadá ... provede se krok 2.2.“ apod.

Výstupní podmínky – Omezení týkající se stavu systému na konci případu užití.

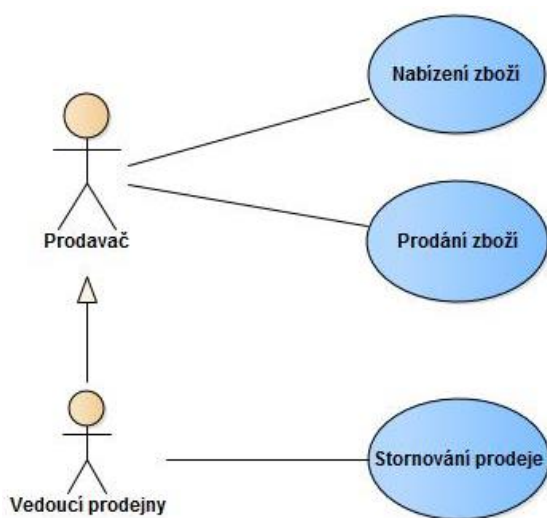
Alternativní scénáře – Alternativy hlavního scénáře znázorňující chyby, rozvětvení nebo přerušení hlavního scénáře.

1.4.3.4. Relace

Vztahy vznikající mezi aktéry, případně mezi případy užití se nazývají relace, které mezi nimi znázorňují plynoucí tok informací. V Use Case Diagramu neboli v Diagramu případů užití mohou nastat tyto typy vztahů: zobecnění aktéra, zobecnění případu užití, include a extend.

Zobecnění aktéra

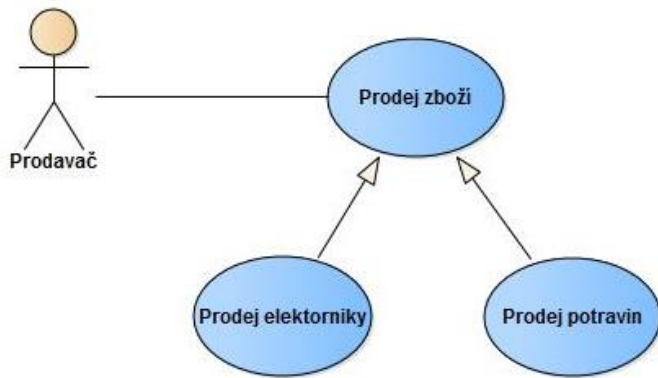
„Zobecnění aktéra umožňuje vyčlenit chování společné dvěma nebo více aktérům do nového rodičovského (nadřazeného) aktéra.“ [6, str. 116]. Zobecnění aktéra zachycuje stejné chování. Potomek dědí od předka role a relace s případy užití. Potomci jsou tak více specializovaní než jejich předci. Potomka lze dosadit všude, kde by mohl být i jeho předek.



Obr. 11 Zobecnění aktéra (Vlastní)

Zobecnění případů užití

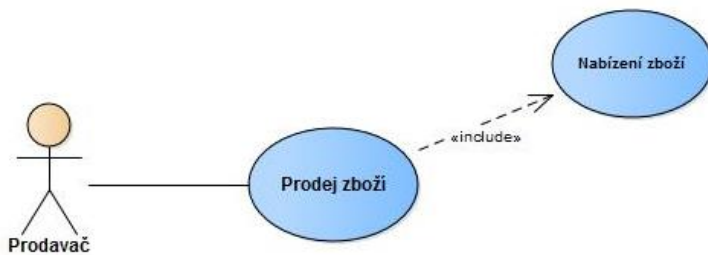
„Zobecnění případů užití umožňuje vyčlenit chování společné dvěma nebo více případům užití do rodičovského (nadřazeného) případu užití.“ [6, str. 118]. Podobně jako u generalizace aktérů se používá jen za předpokladu zpřehlednění diagramu.



Obr. 12 Zobecnění případů užití (Vlastní)

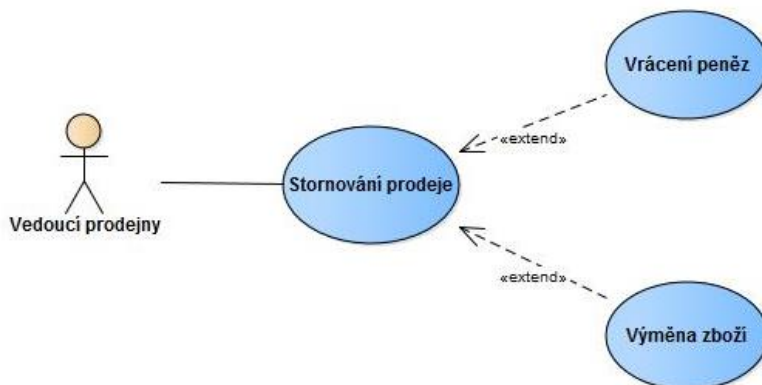
Relace << include >>

„Relace << include >> vyčleňuje kroky společné několika případům užití do samostatného případu užití, který je do příslušných případů užití následně zahrnut.“ [6, str. 121].



Obr. 13 Relace Include (Vlastní)

Relace << extend >> „Relace << extend >> poskytuje způsob, jímž lze existující případ užití rozšířit o nové chování.“ [6, str. 123]. Extend nabízí alternativní volitelný průběh akcí. Důležité je, že rozšíření nemusí být realizováno, pokud nedojde ke splnění dané podmínky.



Obr. 14 Relace Extend (Vlastní)

1.4.4 Class Diagram

Class Diagram v překladu též Diagram tříd je základní strukturální diagram UML. Popisuje základní stavební prvky systému a jejich vzájemné vztahy. Diagram tříd vychází z definované funkcionality v Use Case Diagramu. Diagram tříd se vytváří ve dvou úrovních [9]:

Model analytických tříd

Model analytických tříd definuje základní logické celky a jejich vztahy. Jednotlivé analytické třídy se skládají z názvu, hlavních atributů (vlastností) a hlavních metod (chování).



Obr. 15 Analytická třída (Vlastní)

„Postup identifikace analytických tříd:

1. Analýza scénářů typových úloh, pojmů problémové domény a dalších zdrojů informací, tj. hledání činností, které musí provádět systém, aby realizoval vybrané kroky scénářů typových úloh.
2. Navržení metod zodpovídajících za identifikované činnosti.
3. Rozmístění metod do analytických rozhraní dle hlediska soudržnosti (metody pracují s jednou skupinou dat, se stejnou mírou podrobnosti, minimalizací vztahů a závislostí). Navržení analytických tříd realizujících analytická rozhraní – třídy musí být obrazem entit reality s jasným významem a zodpovědností.
4. Doplnění atributů a dalších metod do tříd – identifikace na základě reálných vlastností entit (primární a vypočtené atributy, podpůrné metody).
5. Doplnění vztahů a závislostí mezi třídami.“ [9, Analytický model tříd].

Mezi vlastnosti dobré analytické třídy patří [9,6]:

- název odráží její účel,
- definována na běžně používaný pojem – zobecnění prvku problémové domény,
- jasně definovaná množina odpovědností 3–5,
- vlastnosti třídy dávají přehled o jejím účelu,
- minimum vazeb na okolní třídy.

Mezi nejčastější chyby při tvorbě analytických tříd patří [9,6]:

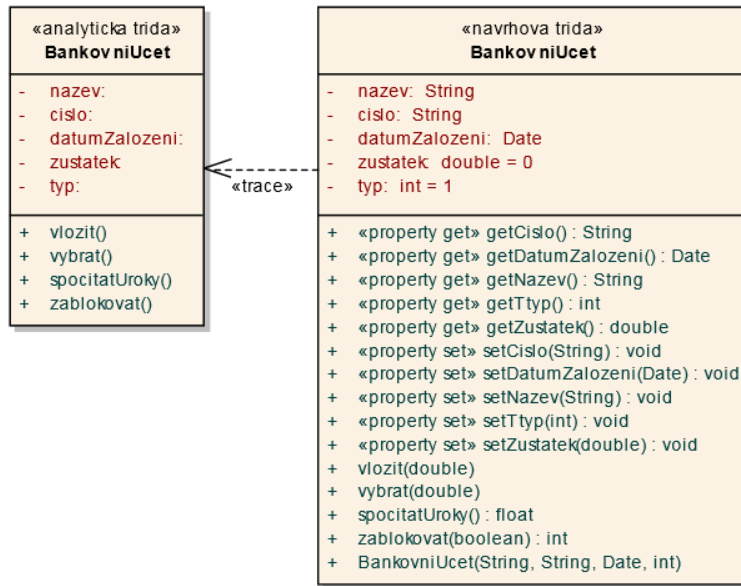
- špatné názvy tříd – zkratky, technické názvy, aktivity, nevystihují účel aj.,
- špatné názvy metod – přesně nevystihují podstatu,
- jedná se o tzv. funktoid, což je třída s jedinou operací,
- široce rozvětvený strom dědičnosti,
- mnoho vazeb na další třídy.

Model návrhových tříd

Model návrhových tříd vzniká postupným zpřesňováním analytických tříd (zpřesňování stávajících a doplňování nových) až do té fáze, kdy je lze implementovat.

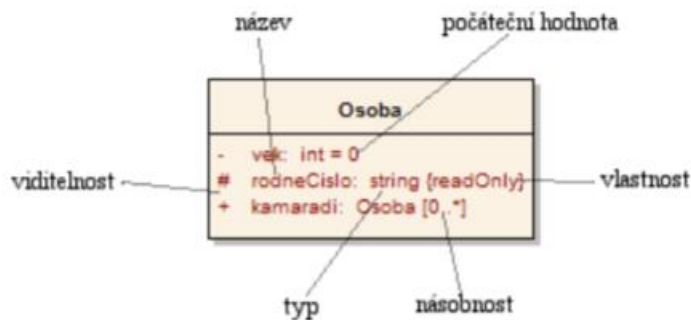
„Postup zpřesňování analytického modelu do návrhového:

- Doplnění datových typů atributů.
- Určení návratových hodnot metod.
- Doplnění parametrů metod.
- Doplnění přístupových metod.
- Doplnění výchozích hodnot atributů event. parametrů.
- Doplnění konstruktorů a destruktorů.
- Určení viditelnosti atributů a metod.
- Zavedení pomocných tříd u atributů s vnitřní strukturou (např. Adresa).
- Využití tříd v knihovnách implementačního prostředí (např. Date, Time, ...).
- Zpřesnění relací.
- Zavedení vztahu dědičnosti.“ [9, Návrhový model tříd].



Obr. 16 Zpřesňování analytického modelu do návrhového [9]

Pro správnou syntax atributů se používá [6]:



Obr. 17 Atributy [9]

viditelnost **název**: typ [násobnost] = PočátečníHodnota {vlastnost}

- **Viditelnost**

„Omezení viditelnosti je mechanismus řídící přístup k členům třídy.“ [9, str. 155].

V jazyku UML2 se definují čtyři typy viditelnosti uvedené na obrázku 18.

+	Veřejný (public)	Jakýkoli prvek s přístupem k třídě může používat jakékoli její vlastnosti a funkce deklarované jako veřejné.
-	Soukromý (private)	K soukromým vlastnostem a funkcím mají přístup pouze operace uvnitř dané třídy.
#	Chráněný (protected)	K chráněným vlastnostem a funkcím mají přístup pouze operace uvnitř dané třídy a uvnitř jejich potomků.
~	Balíček (package)	K vlastnostem a funkcím třídy deklarovaným pomocí klíčového slova package mohou přistupovat libovolné prvky z téhož balíčku jako příslušná třída, ale i prvky z vnořených dílčích balíčků.

Obr. 18 Typy viditelnosti [6]

- **Název**

Názvy atributů se píše tzv. VelbloudímZápisem, většinou se jedná o podstatná jména či jmenné fráze.

- **Typ**

Jazyk UML definuje čtyři primitivní datové typy, které určují typ a rozsah atributů (viz obrázek č. 19).

Integer	Celé číslo
UnlimitedNatural	Celé číslo větší nebo rovno nule. Nekonečno je vyjádřeno hvězdičkou.
Boolean	Může mít hodnotu true nebo false.
String	Posloupnost znaků. Řetězcové doslovné hodnoty jsou v uvozovkách.

Obr. 19 Primitivní datové typy UML [6]

- **Násobnost**

Násobnost neboli multiplicita značí počet objektů v daném vztahu. Udává se ve tvaru Minimum ... Maximum.

- **Počáteční hodnota**

Definuje počáteční hodnotu atributu v době vytvoření objektu.

1.4.4.1 Relace v Class Diagramu

Podobně jako v Use Case Diagramu i zde můžeme definovat vztahy, které mohou mezi třídami nastat [6,12]:

Závislost

Závislost vyjadřuje, že změna provedená v hlavní třídě se projeví i v závislé třídě. Značí se přerušovanou šipkou ve směru vztahu.



Obr. 20 Závislost (Vlastní)

Asociace

Asociace je základním vztahem mezi třídami. Takto spojené třídy mohou existovat nezávisle na sobě. Značí se plnou čarou, může být uvedena šipka značící směr, roli a vlastnosti.



Obr. 21 Asociace (Vlastní)

Agregace a kompozice

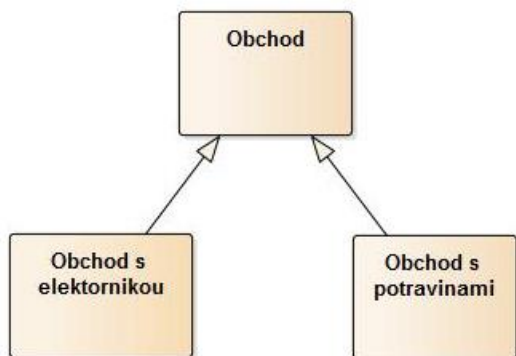
Agregace i kompozice reprezentují vztah „celek – část“, který si můžeme představit jako „skládá se“. Agregace značí volnou vazbu mezi objekty, což znamená, že objekt reprezentující část může existovat i bez celku sám o sobě. Naproti tomu kompozice značí silnější vztah – část nemůže existovat bez celku.



Obr. 22 Agregace a Kompozice (Vlastní)

Generalizace

Generalizace neboli zobecnění (dědičnost) mezi třídami se chová podobně jako dědičnost v Use Case Diagramu. Specializované třídy mají všechny vlastnosti a chování svého předka a navíc nějaké další. Generalizace se značí plnou čarou s uzavřenou šipkou na straně předka.



Obr. 23 Generalizace (Vlastní)

1.5 Webdesign a User experience

1.5.1 Webdesign

Cílem webdesignu je přilákat na web návštěvníky, upoutat je, přimět je k akci, docílit toho, aby se na web vraceli, a v neposlední řadě získání zisku. Webdesign se neustále vyvíjí a mění, co platilo před pár lety, je dnes velkou minulostí.

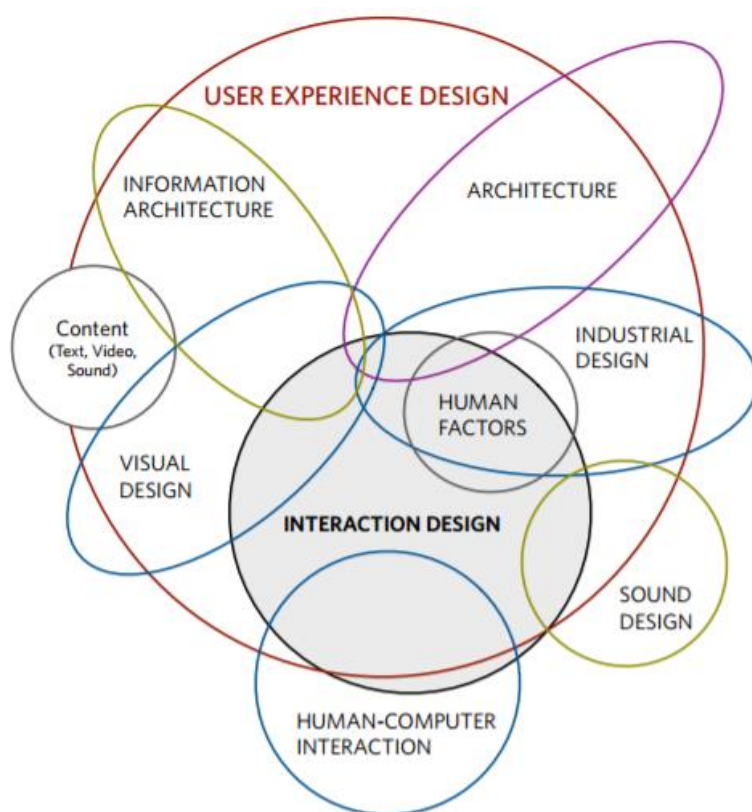
Mylnou představou je vidět webdesign jen jako grafickou podobu webových stránek, popř. aplikace. U webdesignu je potřeba soustředit se na průnik tří základních oblastí [23, 24]:

1. Obsahová analýza

Jednou z nejdůležitějších věcí webu je jeho obsah. Text musí být pro zákazníky srozumitelný, dostatečně dlouhý, strukturovaný a musí zákazníka zaujmout.

2. Interakční design

Web musí být dostatečně srozumitelný, intuitivní a pro uživatele přehledný.



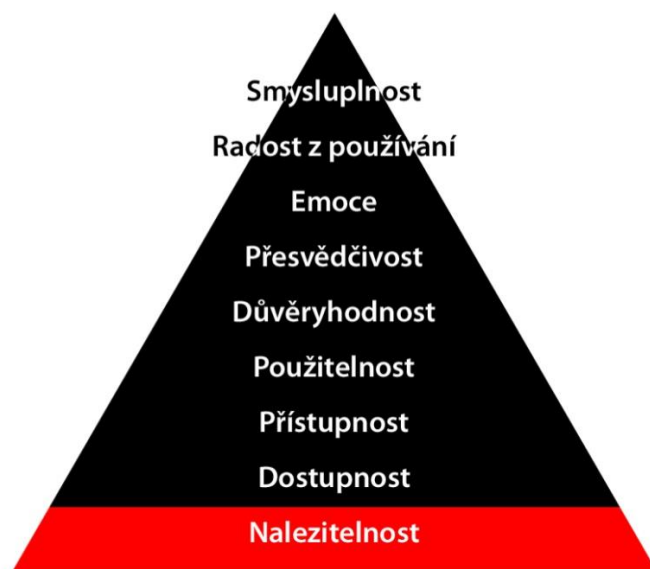
Obr. 24 User experience [25]

3. Vizuální komunikace

Určuje grafickou podobu webu.

Pro dosažení cílů kvalitního webdesignu slouží tzv. Maslowova pyramida webdesignu [24]:

- **Nalezitelnost** – uživatelé musí web nalézt.
- **Dostupnost** – web musí být uživatelům dostupný, rychlý (kvalitní webhosting, kvalitní programátorský kód, SEO, ...).
- **Přístupnost** – web musí být přístupný komukoli, kdekoli s připojením k internetu, důležité je věnovat pozornost přístupnosti webu na různých zařízeních.
- **Použitelnost** – web musí být intuitivní, „Nenuťme uživatele přemýšlet.“
- **Důvěryhodnost** – web musí budit důvěru (jasný provozovatel, úplné kontaktní fakturační údaje, podmínky prodeje, ...).
- **Přesvědčivost** – využívání metod psychologie k přesvědčení zákazníka k požadovanému cíli (uskutečnění prodeje, návratnost, ...).
- **Emoce** – emocionální působení na uživatele.
- **Radost z používání** – uživatelé musí s webem pracovat rádi (herní principy, jednoduchost a jiné motivační prostředky, ...).
- **Smysluplnost** – dlouhodobý smysl pro uživatele.



Obr. 25 Maslowova pyramida webdesignu [24]

Mezi nejčastější chyby webdesignu můžeme zařadit například: nestrukturovaný text (absence odstavců), což je pro uživatele zcela nečitelné, chybně zvolená velikost textu, chybně zvolená barva textu v kontrastu s pozadím webu, umístování rušivých reklam, porušení designové konzistence, ... [27].

1.5.2 User Experience

User Experience, zkráceně UX, můžeme chápat jako soubor pravidel a způsobů, jak udělat web použitelný, užitečný a pro uživatele designově atraktivní. Tvorba kvalitního User experience prochází opakovaně několika fázemi:

- **Výzkum** (Research) – pro koho tvoříme web, jakou mají představu, ... (tvorba Person).
- **Požadavky** (Requirements) – proč web tvoříme, jaký je účel webu, čeho chceme dosáhnout, ...
- **Koncept** (Concept) – představa.
- **Design** – návrh webu (drátěné modely, funkční prototypy, ...).
- **Testování** (Validation) – testování webu (uživatelské testování, testování profesionály, ...).

Jacob Nielsen uvádí 10 heuristik (pravidel), které musí splňovat web, který má tzv. „dobré User Experience“ [26]:

1. **Viditelnost** – uživatel musí vždy vědět kde je a co se odehrává.
2. **Shoda mezi systémem a reálným světem** – web musí komunikovat jazykem uživatele (fráze a pojmy jsou pro uživatele srozumitelné).
3. **Uživatelské ovládání a svoboda** – uživatel musí mít vždy možnost vrátit se zpět.
4. **Soudržnost a standardy** – jednotnost webu, uživatel musí vědět, co se stane při stisknutí tlačítka, ...
5. **Předcházení chybám** – lepší než chybové hlášky je takový návrh webu, který nedovolí, aby uživatel chybu provedl (validátor hesla, upozornění na povinná pole, ...).
6. **Rozpoznávání** – nenutit uživatele k zapamatování mnoha věcí, ale zobrazovat jim informace, pro ně v danou chvíli užitečné (nápověda, ...).
7. **Flexibilita a efektivita** – možnost přizpůsobení se danému uživateli (personifikace).

8. **Estetický a minimalistický design** – „Méně je více.“ Mnoho informací často vede k přehlédnutí důležitých aspektů webu.
9. **Pomoc pro uživatele** – chybová hlášení musí přesně identifikovat problém, aby uživatel věděl, jak může chybu opravit.
10. **Nápovědy a dokumentace.**

1.6 Marketingová strategie

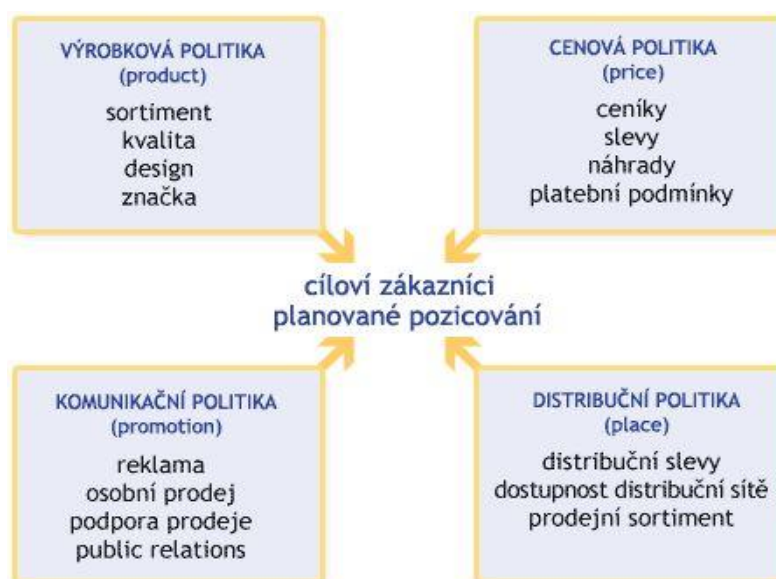
Marketingová strategie se věnuje analýze produktu a následné segmentaci, targetingu a positioningu. Definiuje skupinu zákazníků, analyzuje konkurenci a napomáhá dosáhnout stanovených cílů.

1.6.1 Marketingový mix

„Marketingový mix je souborem taktických marketingových nástrojů – výrobní, cenové, distribuční a komunikační politiky, které firmě umožňují upravit nabídku podle přání zákazníků na cílovém trhu.“ [13, s. 105].

„Marketingový mix je tedy tvořen souhrnem všech nástrojů, které vyjadřují vztah podniku k jeho vnějšímu okolí, tzn. k zákazníkům, dodavatelům, médiím, distribučním a dopravním organizacím, zprostředkovatelům apod.“ [14, s. 75]. O marketingovém mixu můžeme hovořit jako o tzv. „4P“ a patří sem:

- produkt (**P**roduct),
- cena (**P**rice),
- místo, distribuce (**P**lace); propagace (**P**romotion).



Obr. 25 4P Marketingového mixu [16]

Můžeme se také setkat s marketingovým mixem označovaným jako „4C“. Jde o marketingový mix vnímaný z pohledu zákazníka, nikoli kupujícího. Marketingový mix pak bude vypadat takto [15]:

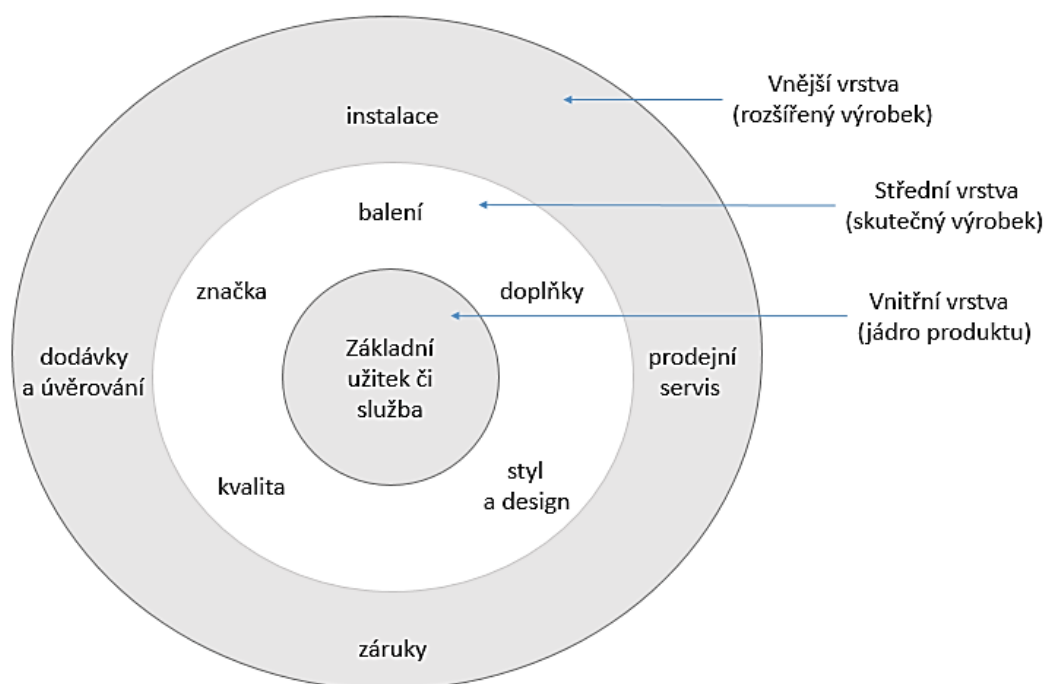
- z produktu se stane zákaznická hodnota (Customer Value),
- z ceny zákaznickova vydání (Cost to the Customer),
- místo se přemění na zákaznické pohodlí (Convenience),
- z propagace se stane komunikace se zákazníkem (Communication).

Produkt

„Cokoli, co lze nabídnout na trhu k upoutání pozornosti, ke koupi, k použití nebo ke spotřebě, co může uspokojit touhy, přání nebo potřeby; patří sem fyzické předměty, služby, osoby, místa, organizace a myšlenky.“ [16, s. 615].

Definují se tzv. tři úrovně produktu [16]:

- **Základní produkt** – představuje účel koupě produktu, základní užitek nebo služba.
- **Vlastní produkt** – charakteristiky výrobku zastupující základní přínos (kvalita, značka, funkce, obal, design).
- **Rozšířený produkt** – služby a výhody produktu.



Obr. 26 Produkt [14]

Cena

Cena jako prvek marketingového mixu jediná generuje zisk, ostatní prvky tvoří náklady.

„Peněžní částka účtovaná za výrobek nebo službu; případně souhrn všech hodnot, které zákazníci vymění za užitek z vlastnictví nebo užívání výrobku nebo služby“ [16, s. 749].

Stanovení ceny patří k nejdůležitějším a nejnáročnějším otázkám, které si při podnikání musíme klást. Při stanovení ceny sledujeme nejen vlastní náklady, ale také změny trhu, životní cyklus produktu, cílovou skupinu zákazníků, konkurenci, soulad celé marketingové komunikace, celkovou firemní strategii, aj. [14].

Faktory ovlivňující stanovení ceny lze rozdělit [14]:

Vnitřní faktory – cíle firmy, marketingový mix, diferenciacce výrobku, náklady aj.

Vnější faktory – poptávka, konkurence, distribuční síť, ekonomické podmínky aj.

Ostatní faktory – psychologické vlivy aj.

Tvorbu cenové strategie si lze zjednodušeně definovat do šesti etap, viz obrázek č. [27]:

1. definice cílových zákazníků,
2. definice nákupního chování určených zákazníků,
3. definice konkurence,
4. odhad prodaného množství při různých cenách,
5. odhad výše nákladů při různých objemech prodeje,
6. definice rozpětí, v kterém se cena výsledného produktu může nacházet.



Obr. 27 Tvorba cenové strategie [14]

Distribuce

„Veškeré činnosti společnosti, které činí produkt nebo službu dostupné zákazníkům.“ [16, s. 71].

Součástí distribuce je [17]:

- **Fyzické přemístování** – přeprava produktu od výrobce ke konečnému spotřebiteli.
- **Změna vlastnictví** – během distribuce přechází vlastnictví výrobku mezi několika články distribučního procesu (od výrobce k zákazníkovi nebo od výrobce přes zprostředkovatele/dodavatele až k zákazníkovi).
- **Doplňkové činnosti** – další činnosti, které s přepravou bezprostředně nesouvisí, ale zajišťují správné podmínky (pojištění, rychlost dodání, ...).

Hlavní rozdělení distribučních cest definuje přímé distribuční cesty (firma prodává své výrobky přímo konečným zákazníkům, bez jakýchkoli prostředníků) a nepřímé distribuční cesty (zahrnují jednoho nebo více prostředníků).

Z hlediska množství zainteresovaných mezičlánků rozeznáváme tři typy distribuce [14]:

- **Intenzivní** – běžné zboží sloužící k uspokojování běžných potřeb zákazníků je k dostání téměř všude (potravin).
- **Selektivní** – množství prodejních míst je omezené, především díky specifické povaze výrobků (elektronika).
- **Exkluzivní** – velmi malý počet prodejních míst, velmi malý počet distributorů, kteří mají výhradní právo prodeje (luxusní zboží).

Propagace

„Propagace znamená šíření informací o něčem s cílem uvedení předmětu propagace (výrobek, služba, akce, skutečnost atd.) ve známost.“ [18, <https://managementmania.com/cs/propagace>].

Můžeme se setkat se dvěma způsoby propagace: nepřímá propagace (směřuje všeobecně na cílovou skupinu – letáčky, inzerce, webové stránky, média, ...) a přímá propagace

(směřuje na konkrétního zákazníka – dopis, email, telefonní hovor) - většinou navazuje na nepřímou propagaci [18].

Forma propagace závisí na cílovém trhu produktu. Je potřeba si odpovědět na otázky, představující tzv. „5M“:

- **Mission** – Jaké jsou cíle propagace?
- **Money** – Kolik můžeme do propagace investovat?
- **Message** – Jaká zpráva by měla být zákazníkům odeslána?
- **Media** – Jaké komunikační kanály by měly být použity?
- **Measurement** – Jak správně hodnotit výsledky? [19]

Propagací se budeme více zabývat v následující kapitole „Marketingový komunikační mix“.

1.6.2 Marketingový komunikační mix

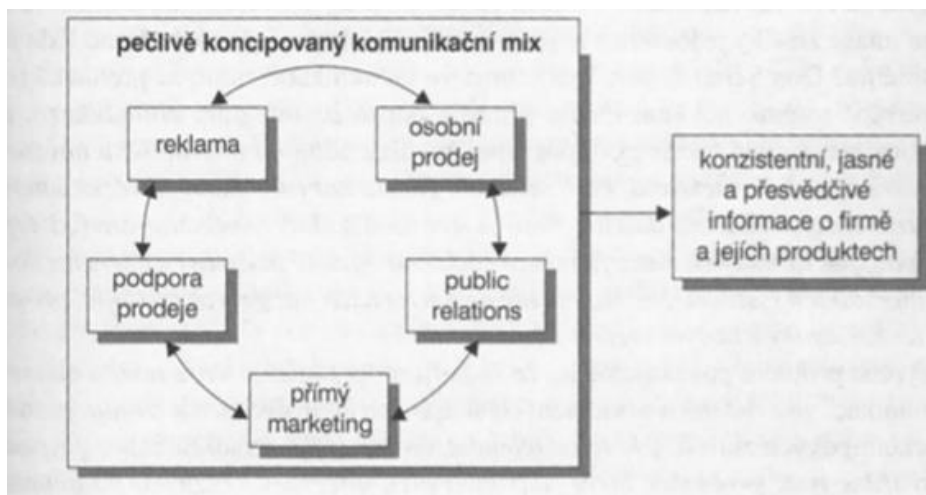
Marketingový komunikační mix se skládá z pěti hlavních nástrojů, prostřednictvím kterých se snaží naplnit marketingové cíle společnosti:

- reklama,
- podpora prodeje,
- public relations,
- osobní prodej,
- přímý marketing.

Marketingová komunikace prošla za dobu své existence velkým vývojem. Zatímco v minulosti se bral zřetel spíše na masový nediferencovaný marketing, dnes dochází k segmentaci a soustředíme se na cílové skupiny zákazníků a jejich specifické potřeby.

Důležitým aspektem je jednotnost marketingové komunikace – sdělení a obsah komunikovaný různými formami musí být v souladu, aby nedocházelo ke zmatení zákazníka.

„Za jednotnou marketingovou komunikaci (IMC) je považována koncepce, jejímž cílem je sladit a koordinovat veškeré firemní komunikační aktivity. Sdělované informace o firmě a její nabídce musí být jasné, konzistentní a přesvědčivé.“ [14, s. 213].



Obr. 28 Jednotná marketingová komunikace [16]

Reklama

„Pojem reklama je definován jako neosobní placená forma komunikace firmy se zákazníkem prostřednictvím různých médií. Je to cílená činnost, která potenciálnímu zákazníkovi přináší informace o existenci produktu, o jeho vlastnostech, přednostech, kvalitě apod., a pak se pomocí racionálních či emocionálních argumentů snaží vzbudit jeho zájem o daný výrobek.“ [14, s. 163].

Reklamu lze charakterizovat podle její úlohy v životním cyklu produktu nebo podle objektu reklamy. Podle úlohy v životním cyklu produktu rozdělujeme reklamu do třech základních skupin:

- **Reklama zaváděcí** – používá se v první fázi životního cyklu výrobku. Slouží k seznámení spotřebitele s daným produktem. Klade se důraz na první dojem – spotřebitel se musí dozvědět podstatné informace o vlastnostech, přednostech, ceně aj.
- **Reklama přesvědčovací** – používá se ve druhé fázi životního cyklu výrobku, tedy ve fázi růstu. Využívá se k upevnění a zlepšení pozice na trhu, cílem je zákazníky přesvědčit, aby produkt preferovali a nakupovali. Často se v této fázi můžeme

setkat s tzv. srovnávací reklamou, která vyzdvihuje silné stránky produktu na úkor konkurenčních.

- **Reklama připomínací** – používá se zejména ve třetí fázi životního cyklu produktu, tedy ve fázi zralosti. Hlavním cílem připomínací reklamy je udržovat produkt stále v povědomí spotřebitelů a ujistit zákazníky, že jejich nákupní rozhodnutí bylo správné.

Podle objektu reklamy rozdělujeme reklamu na dvě skupiny:

- **Výrobová reklama** – především zdůrazňuje přednosti a výhody produktů. Výrobce se snaží produkt co nejvíce odlišit od konkurenčních výrobků zejména v inovaci tvaru, provedení, lepší konstrukci, použití kvalitnějších materiálů aj.
- **Podniková reklama** – používá se zejména u produktů, které díky jejich charakteru nelze nijak výrazně odlišit. Firmy se snaží vzbudit u spotřebitelů důvěru v podnik, zdůrazňují kladné stránky jejich podnikatelské činnosti a budují tak image firmy.

Podpora prodeje

„Podporu prodeje lze charakterizovat jako soubor činností působících jako přímý podnět, který buď nabízí dodatečnou hodnotu kupujícím, anebo v rámci distribuční cesty působí jako podpůrný stimul na prostředníky a prodejce.“ [14, s. 169].

Podpora prodeje přitahuje pozornost spotřebitelů a vyvolává silné stimuly k nákupu prostřednictvím výhod, které jim nabízí. Na rozdíl od reklamy má podpora prodeje krátkodobý charakter – zákazníci kupují produkt jen z hlediska výhod, které jim podpora prodeje nabízí, poté se vrátí k výrobkům konkurenčním.

Do podpory prodeje můžeme zařadit tyto nástroje: kupóny, vzorky výrobku, rabaty, prémie, cenově výhodné balení, spotřebitelské soutěže, předvádění výrobků, ochutnávky, dárky, ...

Firmy se uchylují k podpoře prodeje zejména kvůli potenciálním efektům, které může přinést, např. zvýšení tržního podílu, ziskovost, opakovaný nákup, zvýšení prodeje kategorie produktů, zvýšení prodeje značky, zdůraznění image značky, ...

Časté uplatňování podpory prodeje může ale přinést řadu rizik – příliš časté slevy mohou vyvolat image nízké kvality, zákazníci vyčkávají a produkty kupují pouze za zvýhodněných podmínek, aj

Public relations

Public relations, zkráceně PR, se věnuje vytváření a udržování dobrých vztahů veřejnosti k firmám, k jejich výrobkům a aktivitám. Firmy se snaží vzbudit u zákazníka dojem dobrého jména firmy, sponzorují různé akce, věnují se charitativní činnosti, dbají na kvalitní životní prostředí, ...

PR je soubor z několika nástrojů označovaných jako „PENCILS“ :

- **P**ublications – publikace = firemní časopisy, výroční zprávy, brožury pro potřeby zákazníků, ...
 - **E**vents – veřejné akce = sponzorování sportovních, uměleckých, či kulturních akcí.
 - **N**ews – novinky = materiály pro novináře, podklady pro tiskové konference, novinky přinášející pozitivní zprávy o firmě, zaměstnancích, ...
 - **C**ommunity involvement activities – angažovanost pro komunitu = investice do místních společenství – školství, kultura, charita, ...
 - **I**ntity media – nosiče a projevy vlastní identity = hlavičkové papíry, vizitky, firemní kultura oblékání, ...
 - **L**obbing aktivity – lobbistické aktivity = lobování za cíle společnosti, prosazení příznivých legislativních opatření, ...
 - **S**ocial responsibility – aktivní sociální odpovědnost = aktivity v sociální oblasti, ...
- [14, str. 175].

Osobní prodej

Při osobním prodeji dochází k osobní interakci mezi prodejcem a spotřebitelem. Prodejce prezentuje, zdůrazňuje jeho hlavní výhody a snaží se získat zákaznickovy preference.

„... cílem osobního prodeje je nalézt zájemce, informovat je, předvést jim funkcionalitu výrobku, přesvědčit je, aby produkt koupili, a poskytovat jim prodejní služby.“ [20, str. 465].

Výhodou osobního prodeje je okamžitá zpětná vazba, prodej umožňuje komunikaci s konkrétním zákazníkem a prodejce tak může v prodejním rozhovoru cílit přesně na jeho potřeby. Podstatnou stránkou osobního prodeje není jen samotný prodej, ale komplexní představení produktu zákazníkovi, informování o vhodném používání a praktické seznámení s výrobkem či službou.

Přímý marketing

Přímý marketing (Direct Marketing), si klade za cíl kontaktovat zákazníky s cílem vyvolat okamžitou reakci. Vychází z přesné segmentace trhu a snaží se udržovat individuální vztahy se zákazníkem. Pro udržení vztahu se zákazníky je zapotřebí budovat rozsáhlou databázi zákazníků. Mezi nejčastější formy přímého marketingu patří e-mail, telemarketing, on-line marketing, ...

1.6.3 Analýza prostředí

Marketingové prostředí je definované jako: „Činitelé a síly vně marketingu, které ovlivňují schopnost marketing managementu vyvinout a udržovat úspěšné vztahy s cílovými zákazníky.“ [16, str. 129].

Marketingové prostředí se dělí na dvě oblasti:

Marketingové mikroprostředí

Faktory v rámci marketingového mikroprostředí může firma snáze či hůře ovlivnit. Můžeme mezi ně zařadit zejména [16]:

- **Společnost** – všechny podnikové aktivity napříč různými odděleními musí korespondovat s cíli marketingové koncepce.
- **Dodavatelé** – poskytují zdroje pro výrobu zboží a služeb. Špatná dostupnost, nedostatek či zpoždění dodávek mohou firmu dlouhodobě poškodit.
- **Marketingoví zprostředkovatelé** – distribuční společnosti, které pomáhají propagovat a prodávat produkt cílovému spotřebiteli.
- **Zákazníci** – firma musí důkladně sledovat trhy svých zákazníků a přizpůsobit svůj přístup k nim.
- **Konkurenti** – každá firma si musí uvědomit svou pozici na trhu ve vztahu ke konkurenci – firmy se snaží získat strategickou výhodu.
- **Veřejnost** – mínění veřejnosti, finanční instituce, média, vládní instituce, místní komunita a občané.

Marketingové makroprostředí

Marketingové makroprostředí nemůže firma jakkoli ovlivnit ani kontrolovat. Makroprostředí tvoří šest faktorů, které firmu ovlivňují zvenčí [16]:

- **Demografické prostředí** – růst populace, porodnost, úmrtnost, stárnutí obyvatelstva, migrace.
- **Ekonomické prostředí** – míra inflace, kurzy měny, nezaměstnanost.
- **Přírodní prostředí** – nedostatek surovin, znečištění vody, devastace životního prostředí, rostoucí náklady na energii.
- **Technologické prostředí** – zrychlující se tempo inovací, nové technologie, materiály, postupy.
- **Politické prostředí** – legislativa, zákony, vyhlášky.
- **Kulturní prostředí** – náboženské rozdíly, jazykové rozdíly.

Mezi nejčastěji používané analýzy marketingového prostředí patří:

SWOT analýza

SWOT analýza je nástrojem pro zhodnocení vnitřních a vnějších vlivů působících na organizaci nebo konkrétní produkt.

Cílem je identifikace silných a slabých stránek společnosti/produktu jako vlivů působících uvnitř společnosti a určení hrozeb a příležitostí jako vlivů ovlivňující firmu zvenčí.



Obr. 29 Analýza marketingového prostředí [21]

PESTEL analýza

PESTEL analýza se používá pro určení vlivů okolního prostředí organizace, které dělí do několika skupin:

- P** – politické – působení politických vlivů;
- E** – ekonomické – působení a vliv místní, národní a světové ekonomiky;
- S** – sociální - sociální a kulturní (lokální, národní, regionální, světové);

- T** – technologické – vliv nových a vyspělých technologií;
- E** – ekologické (environmentální) – otázky životního prostředí;
- L** – legislativní – vlivy národní, evropské a mezinárodní legislativy.

Porterův model pěti konkurenčních sil

Porterův model pěti konkurenčních sil označovaný také jako analýza 5F se používá k analýze daného odvětví a rizik, resp. ke zkoumání konkurenčního prostředí.

- **Stávající konkurenti** – mohou konkurenti ovlivnit cenu, existuje dominantní konkurent.
- **Potenciální konkurenti** – za jakých podmínek mohou vstoupit na trh noví konkurenti, existují bariéry vstupu.
- **Dodavatelé** – mohou dodavatelé výrazně ovlivňovat ceny nebo dodávané množství, jak silná je vyjednávací pozice dodavatelů, jedná se o monopolní dodavatele nebo jich existuje pro dané odvětví více.
- **Kupující** – jak silná je pozice kupujících, mohou ovlivnit cenu či nabízené objemy daného produktu.
- **Substituty** – existují produkty, které by mohly snadno nahradit ty naše.

1.6.4 Cílený marketing

„Snaha společnosti obsloužit jednu či více skupin zákazníků sdílejících stejné potřeby nebo charakteristiky.“ [16, str. 457]

Kotler ve své knize uvádí šest kroků k dosažení cíleného marketingu prostřednictvím tržní segmentace, tržního targetingu a tržního positioningu.



Obr. 30 Cílený marketing [16]

Tržní segmentace

„Rozdělení trhu na různé skupiny kupujících s rozdílnými potřebami, charakteristikami či chováním, které mohou vyžadovat odlišné produkty a marketingové mixy.“ [16, str. 457]

Je třeba si uvědomit, že není možné uspokojit potřeby všech zákazníků, vzhledem k jejich odlišnostem. Je tedy potřeba trh segmentovat, neboli rozdělit na skupiny. Neexistuje jediný nebo vhodný způsob segmentace trhu, vždy je potřeba analýza trhu a následný výběr tržního segmentu. Analýzu trhu můžeme provádět z hlediska mnoha proměnných, mezi základní hlediska patří:

- **geografické hledisko** (obec, region, země, ...) – určuje zacílení, působení firmy;
- **demografické hledisko** (věk, pohlaví, příjem, povolání, vzdělání, ...);
- **psychografické hledisko** (společenská třída, životní styl, osobnost, ...);
- **behaviorální hledisko** (uživatelský status, frekvence užívání, loajalita, postoj k produktu, ...).

Pro efektivní segmentaci je nutné, aby daný segment byl:

- **měřitelný** – určuje velikost segmentu a jeho kupní sílu;
- **dostupný** – segment musí být dosažitelný a obsluhovatelný;
- **významný** – segment musí být dostatečně veliký, aby byl rentabilní;
- **praktický** – lze praktikovat realizovatelný marketingový program;

Tržní targeting

„Proces hodnocení atraktivity a výběru tržních segmentů, kam vstoupit.“ [16, str. 457]

Po tržní segmentaci následuje tržní positioning, neboli tržní zacílení, tedy hodnocení vybraných segmentů a rozhodnutí, na které se marketéři zaměří. Hodnotí se atraktivita daného segmentu, neboli potenciál, který daný segment představuje na základě velikosti, konkurence, růstu aj.

Tržní positioning

„Proces, který má zařídít, aby produkt zaujal v myslích cílových zákazníků jasnou, vyhraněnou a žádoucí pozici vzhledem ke konkurenčním produktům.“ [16, str. 458] Jde tedy o umístění produktu do mysli potenciálního zákazníka prostřednictvím vhodné marketingové komunikace.

Praktická část

V praktické části se práce věnuje návrhu webové aplikace sloužící pro podporu kooperace členů projektového týmu a marketingové strategii.

2. Implementační projekt webové aplikace

Předmětem diplomové práce je Implementační projekt a návrh webové aplikace s názvem Group Cooperation (produkt je popsán v kapitole 3.1) Implementační projekt (IP) obsahuje detailnější popis webové aplikace. V běžné praxi slouží IP jako základní dokument, který se sestavuje na základě požadavků zákazníka. IP je součástí smlouvy při vývoji softwaru na míru a definuje veškeré funkční a nefunkční požadavky. Nemůže se tedy stát, že zákazník bude vyžadovat další a další rozšíření aplikace a zároveň se nemůže stát, že zákazník neobdrží, co si objednal. Součástí IP jsou také wireframy (drátěné modely), tedy jednoduché grafické návrhy aplikace a základní UML diagramy s datovým modelem. IP je pak předkládáno do fáze implementace.

2.1 GC – Group Cooperation

Webová aplikace Group Cooperation ve zkratce GC je nástrojem pro řízení projektů malého rozsahu, který zajišťuje veškerou kooperaci mezi členy projektového týmu. Aplikace GC zajišťuje činnosti probíhající při řízení projektu – plánování, řízení, správa dokumentů a komunikace. Po přihlášení se uživateli zobrazí seznam projektů, ve kterých je členem.

Aplikace má několik modulů umožňující veškerou správu a komunikaci na jednom místě:

Nástěnka nabízí přehled o průběhu daného projektu prostřednictvím přehledných grafů průběhu projektu, kalendář s vyznačenými termíny ukončení jednotlivých úkolů a osobní podúkoly, které si uživatelé přidávají na své úkoly pro lepší orientaci. Součástí Nástěnky je také Harmonogram. Znázorňuje posloupnosti projektu v čase. Na první pohled je vidět datum předpokládaného zahájení a dokončení úkolů (tasků), hotové tasky a jejich řešitel. Po najetí myši na konkrétní údaj se zobrazí okno s detailnějším popisem. Jednotlivé úkoly lze rozkliknout a editovat. Na nástěnce se také zobrazují nejaktuálnější úkoly a podúkoly přihlášeného uživatele.

Záložka Úkoly je rozdělena na dvě části. Myšlenková mapa je nástroj umožňující vytvářet logickou strukturu projekt, pomocí rozdělování na části a podřízení úkoly.

Druhá část nabízí přehlednou tabulku všech úkolů se základními charakteristikami. Jednotlivé úkoly lze editovat, přidávat, mazat, popřípadě filtrovat. Veškeré úkoly a podúkoly vytvořené v Myšlenkové mapě se generují do Úkolů a naopak. Obsah je tedy stejný, ale forma je jiná. Uživatel si může zvolit, jaká metoda tvorby struktury projektu je pro něj výhodnější

Záložka **Soubory** umožňuje správu souborů potřebných k projektu. U každého souboru je vidět název, k jakému úkolu patří, Autor souboru a popis.

Zprávy umožňují rychlou komunikaci mezi členy projektového týmu. Možné komunikovat s celou skupinou najednou, popřípadě s jednotlivými členy. Lze vytvářet nová konverzační vlákna na téma jednotlivých úkolů, aj.

Nastavení umožňuje editaci základních charakteristik daného projektu a přidávání jednotlivých členů projektového týmu. Dále je zde možnost označení projektu za hotový.

2.2 Požadavky na aplikaci

2.2.1 Funkční požadavky

- **Úvodní strana s přihlašovacím formulářem**
 - Uživatel pro přihlášení zadá email a heslo.
 - Odkaz na registraci nového uživatele – Odkaz Registrovat = uživatelské jméno, heslo, email.
 - Odkaz zapomenuté heslo – na zadaný email bude posláno nové vygenerované heslo.
- **Homepage aplikace GC**
 - Přehledná tabulka projektů – Název, Datum zahájení, Datum dokončení, % plnění.
 - Možnost přidat, editovat, odstranit projekt a filtrovat.
 - Možnost řazení dle záhlaví tabulky.
 - Možnost prokliku na detail projektu (Nástěnka) nebo do nastavení projektu.
- **Nástěnka konkrétního projektu**
 - Přehled průběhu projektu.

- Graf celkového průběhu projektu – koláčový graf vyjadřující procentuální rozložení stavů všech úkolů v projektu – Zadaný, Rozpracovaný, Vyřešený. Po kliknutí na graf se uživatel dostane na záložku úkoly, která bude zobrazovat všechny úkoly v projektu.
- Graf splněných úkolů (přehled práce) dle jednotlivých členů týmu – 100% představuje všechny zadané úkoly danému řešiteli. V grafu je vyjádřeno procentuální rozložení stavů úkolů jednotlivých řešitelů – Zadaný, Rozpracovaný, Vyřešený. Po kliknutí na graf se uživatel dostane na záložku Úkoly, která bude zobrazovat úkoly zvoleného uživatele.
- Harmonogram – zobrazuje harmonogram projektu jako Gantův diagram – přehled naplánovaných posloupností jednotlivých úkolů v čase – lze zobrazit den, týden, měsíc a kompletní průběh projektu.
- Kalendář – Ukazuje aktuální měsíc, vyznačené termíny všech úkolů v projektu.
- V harmonogramu i v kalendáři se po najetí myši na jednotlivé úkoly zobrazí detail – název, řešitel, termín a stav. Možnost prokliku na detail úkolu a barevně odlišené splněné úkoly, úkoly přihlášeného uživatele
- Zobrazení nejaktuálnější úkolů s podúkoly přihlášeného uživatele.

- **Úkoly**

- Dvě záložky - Úkoly a Myšlenková mapa.
- Úkol vytvořený v záložce úkoly se zobrazují také do záložky myšlenková mapa a naopak - totožný obsah, jiné zobrazení.

- **Úkoly**

- Výchozí záložka.
- Přehled úkolů na projektu v tabulce – zobrazuje název úkolu, řešitele, datum zahájení, termín ukončení a stav (Zadaný, Rozpracovaný, Vyřešený).
- Možnost filtrovat „Mé úkoly“, „Všechny úkoly“, „Úkoly po termínu“ a „Úkoly konkrétního řešitele“.
- Možnost přidání, smazání úkolu, filtrování.
- Proklikem se otevře detail úkolu k editaci.
- Možnost řazení dle záhlaví tabulky.

- **Myšlenková mapa**

- Struktura projektu v myšlenkové mapě.
- Možnost přidání, smazání úkolu.
- Zobrazení jednotlivých úkolů jako zaoblené obdélníky.
- Po najetí myši na jednotlivé obdélníky se zobrazí okno s detailem – název, řešitel, datum zahájení a ukončení.
- Proklikem se otevře detail úkolu k editaci.

- **Detail úkolu**

- Okno detailu úkolu je rozděleno na části – základní údaje, soubory, podúkoly, poznámky a historie.
- Základní údaje – možnost zadat název úkolu, možnost zvolit řešitele úkolu, možnost zařadit úkol do existující části projektu, možnost zadat datum zahájení a termín ukončení projektu, možnost nastavit stav úkolu.
 - Základní údaje jsou povinné položky k vyplnění, bez nich nelze úkol uložit.
- Soubory – tabulka obsahující název souboru, autora popis a datum vložení.
 - Možnost přidat soubor, smazat soubor, editovat soubor.
- Podúkoly – tabulka obsahující název podúkolů a termín splnění.
 - Možnost přidat a smazat podúkol.
 - Checkbox pro označení hotového podúkolů.
- Poznámky – možnost přidat jakoukoli poznámku k úkolu – u poznámky se zobrazuje autor a datum vložení.
- Historie – zaznamenává všechny změny týkající se daného úkolu – editaci, změnu stavu, přidání uživatele, souboru, podúkolů.

- **Soubory**

- Přehled přiložených souborů v tabulce – zobrazuje název souboru, úkol, ke kterému patří, autora a popis.
- Možnost přidat, smazat, editovat, stáhnout a filtrovat.
- Pro přidání se zobrazí pop-up okno – ve sloupci úkol rolovací nabídka se zadanými úkoly v projektu.
- Možnost řazení dle záhlaví tabulky.
- Proklikem se otevře detail úkolu k editaci.

- **Zprávy**
 - Možnost poslání zprávy celé projektové skupině, popřípadě jednotlivým členům.
 - Možnost vyhledávání v konkrétní komunikaci.
 - Zobrazení člena skupiny – jméno a fotka uživatele.
 - Identifikace stavu členů – online, offline.
 - Možnost vytvoření konverzačních témat s vybranými uživateli.
 - Možnost smazání konverzačního vlákna.

- **Nastavení**
 - Editace údajů o projektu – název, datum zahájení projektu, datum ukončení projektu, popis.
 - Přehledná tabulka účastníků projektu – Jméno, Email.
 - Možnost přidat, smazat uživatele.
 - Možnost řazení dle záhlaví tabulky členů.
 - Možnost označit projekt za hotový – pouze, když jsou splněny veškeré úkoly.

- **Notifikace**
 - Zobrazení upozornění na změny v projektu.
 - Přidání do nového projektu – možnost přijmout a odmítnout.
 - Upozornění na datum zahájení úkolu, kde je přihlášený uživatel uveden jako řešitel.
 - Upozornění na blížící se termín úkolu, kde je přihlášený uživatel uveden jako řešitel.
 - Termíny osobních podúkolů, které si uživatel připojí na svůj úkol.
 - Upozornění na úkoly po termínu.
 - Přidání nového úkolu, kde je uživatel vedený jako řešitel.
 - Veškeré změny provedené na úkolu, kde je přihlášený uživatel veden jako řešitel a které provedl jiný člen projektového týmu.
 - Veškeré notifikace se budou posílat i na emailovou adresu člena projektového týmu.

2.2.2 Non-Funkční požadavky

- Systém bude webová aplikace.
- Podpora mobilních zařízení.
- Při návrhu formulářů bude využito standardních UX doporučení.
- Podpora nejrozšířenějších standardních prohlížečů – Google Chrome (verze 57), Internet Explorer (verze 11), Microsoft Edge (verze 39), Mozilla Firefox (verze 52).
- Pro vývoj bude použita Základní infrastruktura Microsoft Windows Server 2016.
- MSSQL 2016.
- Bude zaručena perzistence dat v relační databázi (RDBMS).
- Pro vývoj bude využito ASP.NET a MVC, Angular, SignalR, bootstrap.
- Bude využito IIS.
- Pro aplikační vrstvu bude použit standardní web based kontejner.
- Není potřeba dedikovaný aplikační server.
- Aplikace bude využívat dostupné VPS služby pro provoz aplikačního a databázového serveru.
- Použití HTTPS.
- Využití CSP (Content Security Policy) a ochrany proti XSS (Cross Site Scripting).
- Pro ukládání hesel použít hashovací algoritmus bcrypt s podporou salt.

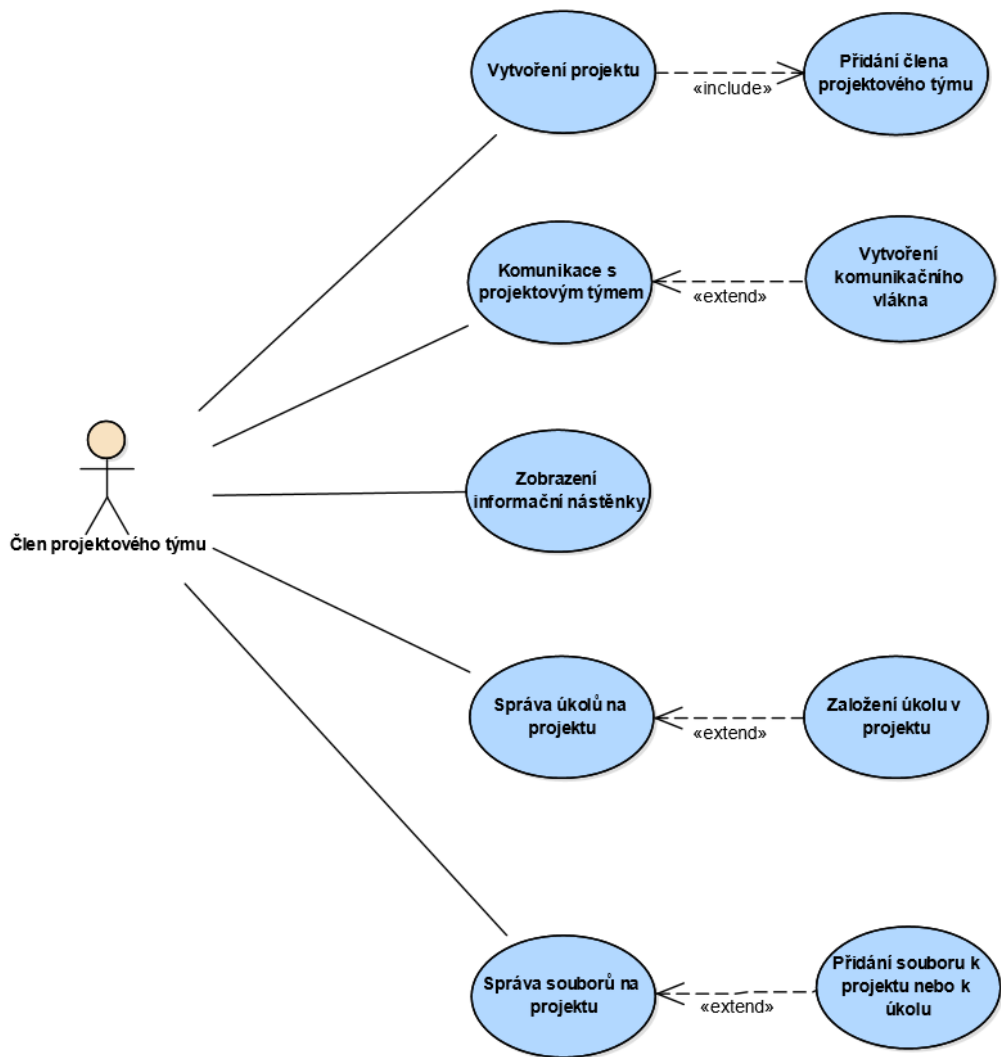
2.3 Návrh systému

Pro návrh systému byl využit CASE nástroj Enterprise Architect, software pro podporu analýzy a návrhu systémů v notaci UML. Pro návrh webové aplikace byl použit Use Case diagram, Class diagram dále pak databázový model a model požadavků.

2.3.1 Use Case Diagram

Use Case diagram je abstraktní popis aplikace bez implementačních podrobností. Hlavní předností diagramu je, že mu rozumí jak vývojáři, tak konzultanti a zákazníci bez znalostí problematiky tvorby IS a programování. Využívá se tedy ke komunikaci se zákazníkem a k definici veškerých požadavků zákazníka. Spolu se seznamem požadavků tvoří nezbytné dokumenty analýzy IS.

Vzhledem k modelování Use Case diagramu pro aplikaci Group Cooperation ve verzi Free byl identifikován pouze jeden Aktér – Člen projektového týmu. V této verzi jsou si všichni členové projektového týmu rovni, všichni mají stejná práva. Aplikace dle zadání musí umožňovat založení projektu. Na konkrétním projektu musí aplikace umožňovat zobrazení informační nástěnky, vytvoření struktury projektu, správu souborů na projektu a komunikaci se členy projektového týmu. Use Case diagram je tvořen na základě identifikovaných požadavků uživatele na systém. Požadavky rozdělujeme na funkční a nefunkční. Požadavky na aplikaci jsou součástí IP v kapitole 2.2. Hierarchii požadavků naleznete v příloze 1.

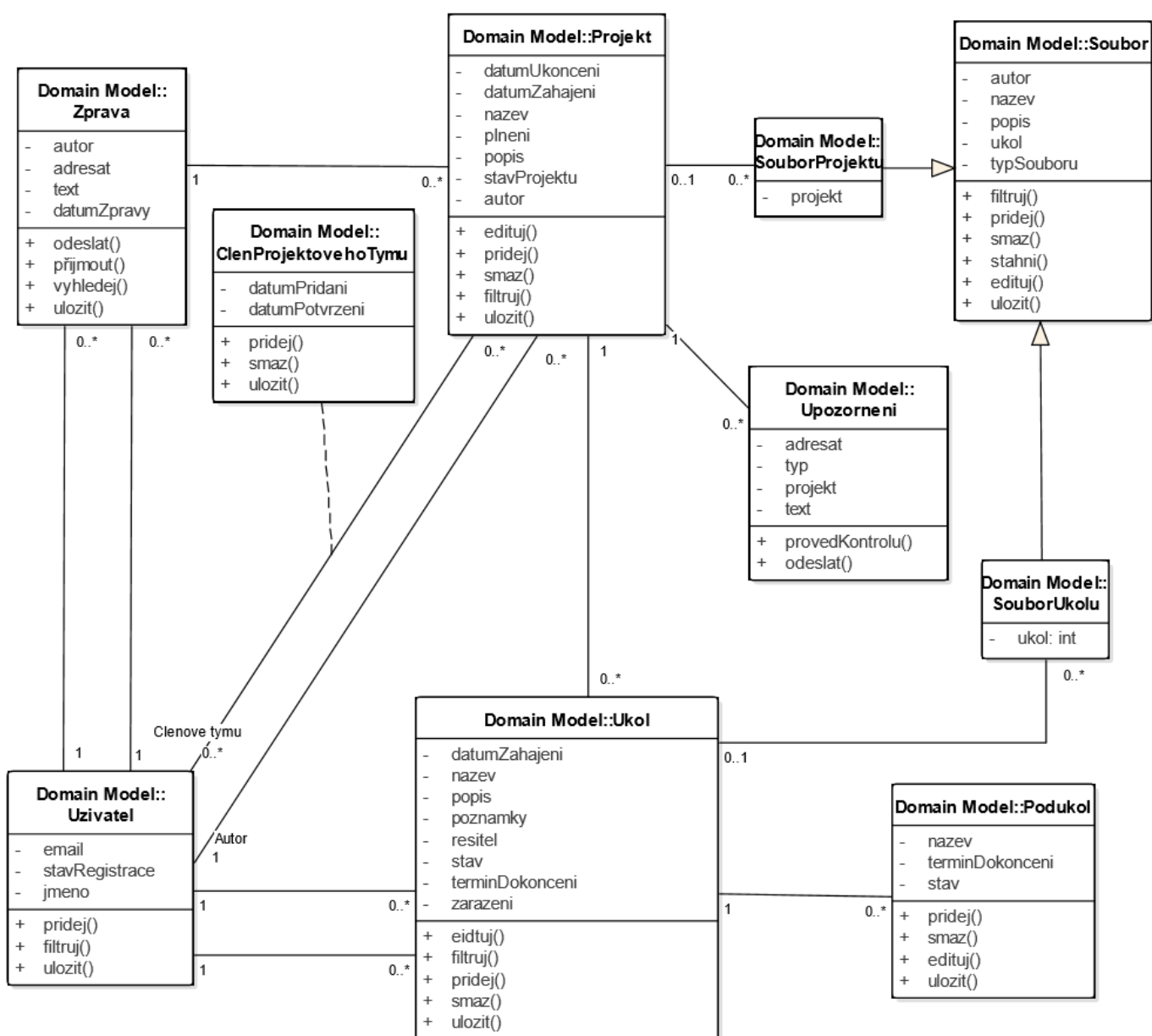


Obr. 32 Use Case model aplik ace (Vlastní)

2.3.2 Class model

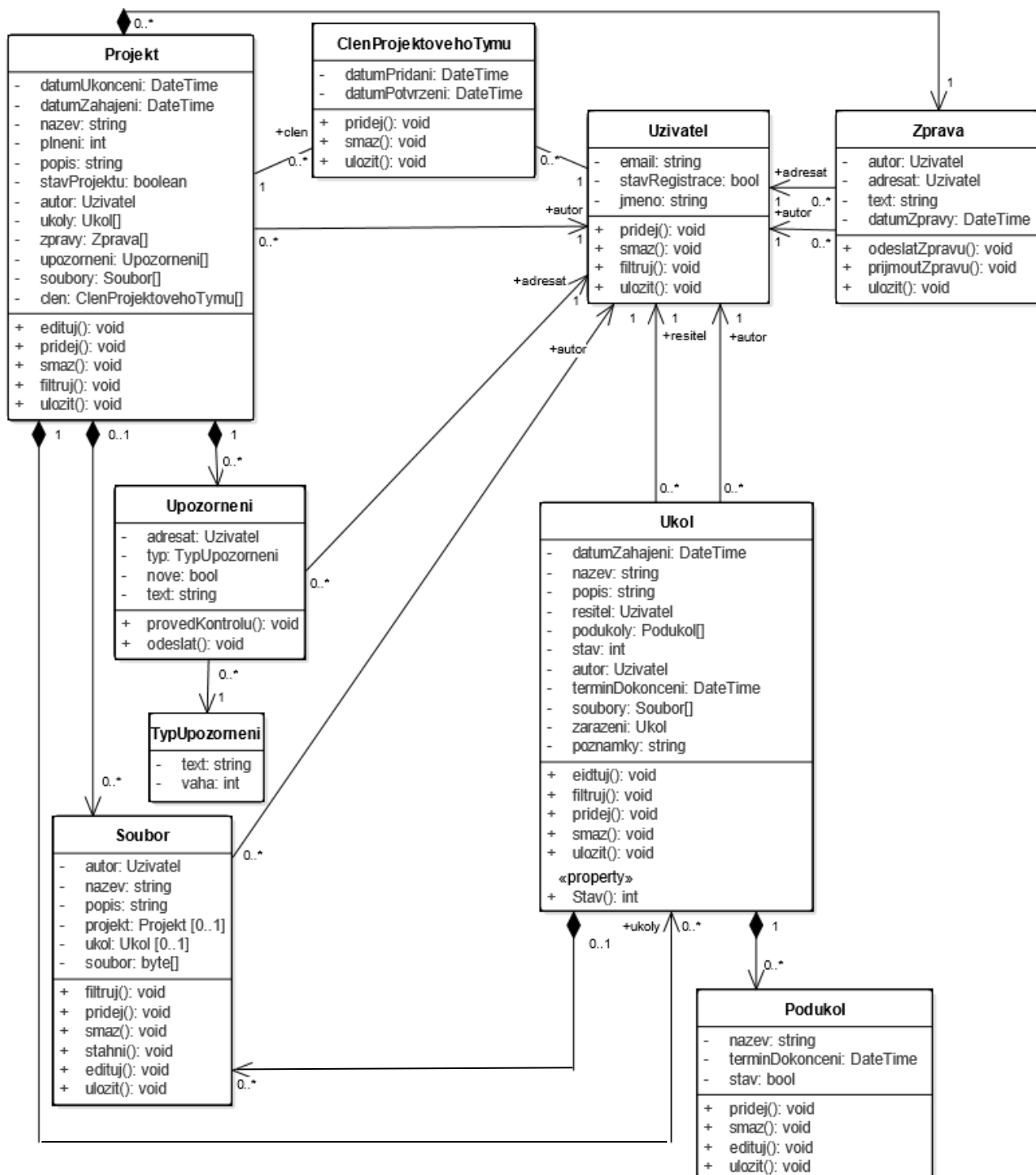
Class Model lze chápat jako grafickou obdobu zdrojového kódu. V první fázi se tvoří analytický model, který definuje třídy, jejich atributy, operace a vztahy. Třídy vznikají z analýzy případů užití – identifikování podstatných jmen a sloves ze scénářů typových úloh.

V modelu byly definovány třídy Zpráva, Projekt, Soubor, Úkol, Podúkol, Uživatel, Soubor u projektu, Soubor u úkolu, Upozornění. Dále byly identifikovány atributy, operace a základní vztahy mezi třídami, viz. Analytický model tříd.



Obr. 33 Analytický model tříd aplikace (Vlastní)

Z fáze analýzy se dostáváme do fáze návrhu a analytický model tříd se rozpracuje na model návrhový. Návrhový model tříd bere v úvahu to, jak bude model implementován. Pro přechod k návrhovému modelu tříd se specifikují atributy, jejich typy a viditelnost, doplní se úplná sada metod a dochází k upřesňování analytických relací, viz Návrhový model tříd.



Obr. 34 Návrhový model tříd (Vlastní)

Byly definovány entity a jejich atributy, následně doplněné o vazby a kardinalitu (omezení v počtu instancí druhé entity). Pro implementaci by byla vygenerována struktura databáze z daného modelu relační databáze, kde by každá entita odpovídala jedné relační tabulce s příslušnými atributy včetně primárních a cizích klíčů.

2.4 Návrh uživatelského rozhraní – GUI

Návrhy uživatelského rozhraní slouží jako základní podklad pro grafické oddělení. Zároveň slouží ke komunikaci se zákazníkem, a k vizualizaci jeho požadavků. Není nutné, resp. je spíše nežádoucí, aby návrh uživatelského rozhraní byl dokonalou grafickou reprezentací budoucí aplikace. Zákazník má v takových případech tendenci soustředit se na grafickou podobu a zapomíná na požadovanou funkcionalitu. Wireframy musí zobrazovat všechny funkční požadavky zákazníka.

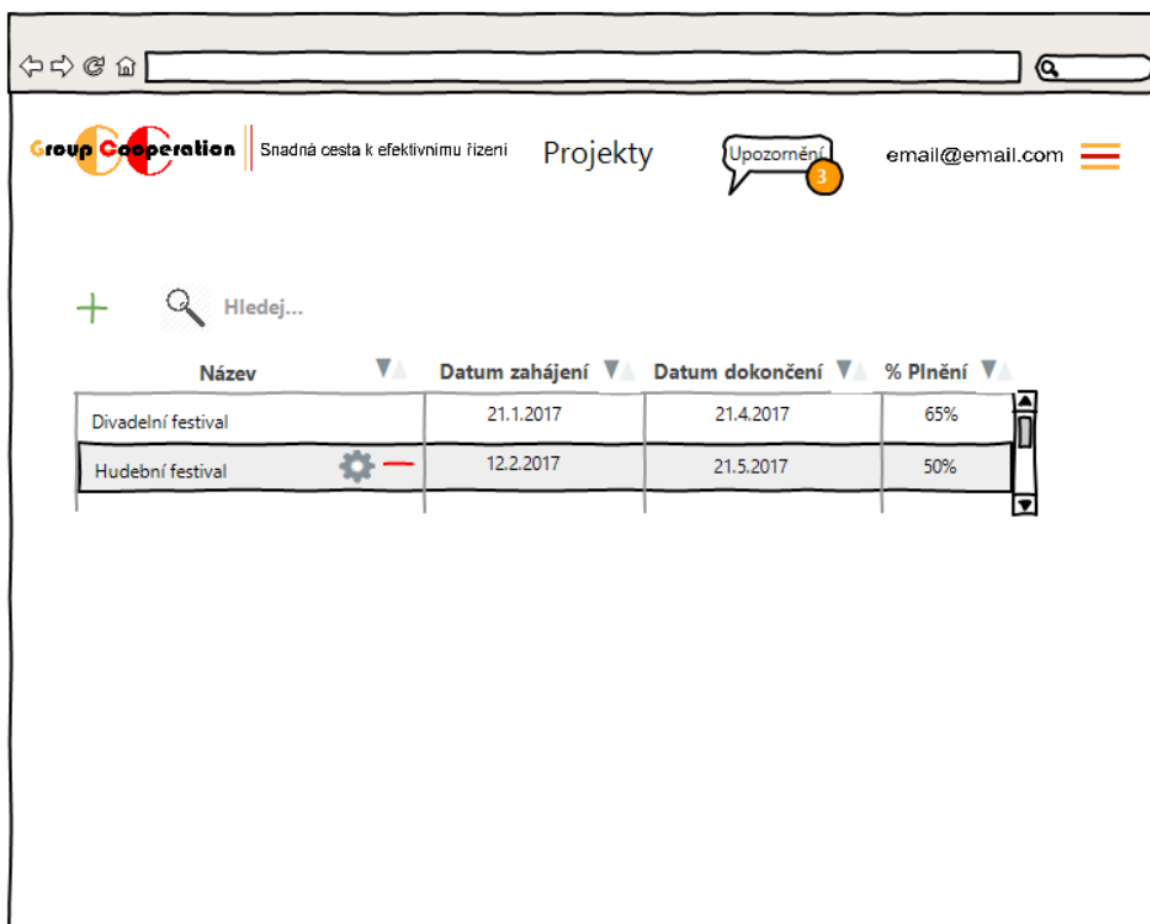
2.4.1 Přihlášení

The image shows a wireframe of a login page. At the top, there is a browser window header with navigation icons and a search bar. Below the header, the logo for "Group Cooperation" is displayed, followed by the tagline "Snadná cesta k efektivnímu řízení". The main content area features a central login form with two input fields labeled "E-mail:" and "Heslo:". Below these fields is an "OK" button. At the bottom of the form, there are two links: "Registrovat" and "Zapomenuté heslo".

Obr. 36 Wireframe - Přihlášení (Vlastní)

Pro vstup do aplikace je nutné přihlášení. Na úvodní straně je uživateli umožněno přihlášení pomocí zadání emailové adresy a heslo. Pokud uživatel není registrovaný, klikne na tlačítko Registrovat, kterému zobrazí registrační formulář obsahující jméno, a e-mailovou adresu a heslo. Zároveň může uživatel na stránce zažádat o heslo nové, přes tlačítko Zapomenuté heslo.

2.4.2 Homepage aplikace



Obr. 37 Wireframe - Homepage aplikace (Vlastní)

Po přihlášení se uživatel dostane na svou úvodní stranu (Homepage), která mu nabídne přehled projektů, ve kterých je členem projektového týmu. Je zde možnost založení nového projektu přes ikonu „plus“. Tabulka zobrazuje název, datum zahájení, datum dokončení a procento plnění projektu. Je zde možnost řazení dle jednotlivých sloupců. Po označení řádku lze kliknout na ikonu Nastavení, která umožňuje proklik do detailu projektu, popřípadě kliknout na ikonu „mínus“ pro odstranění projektu.

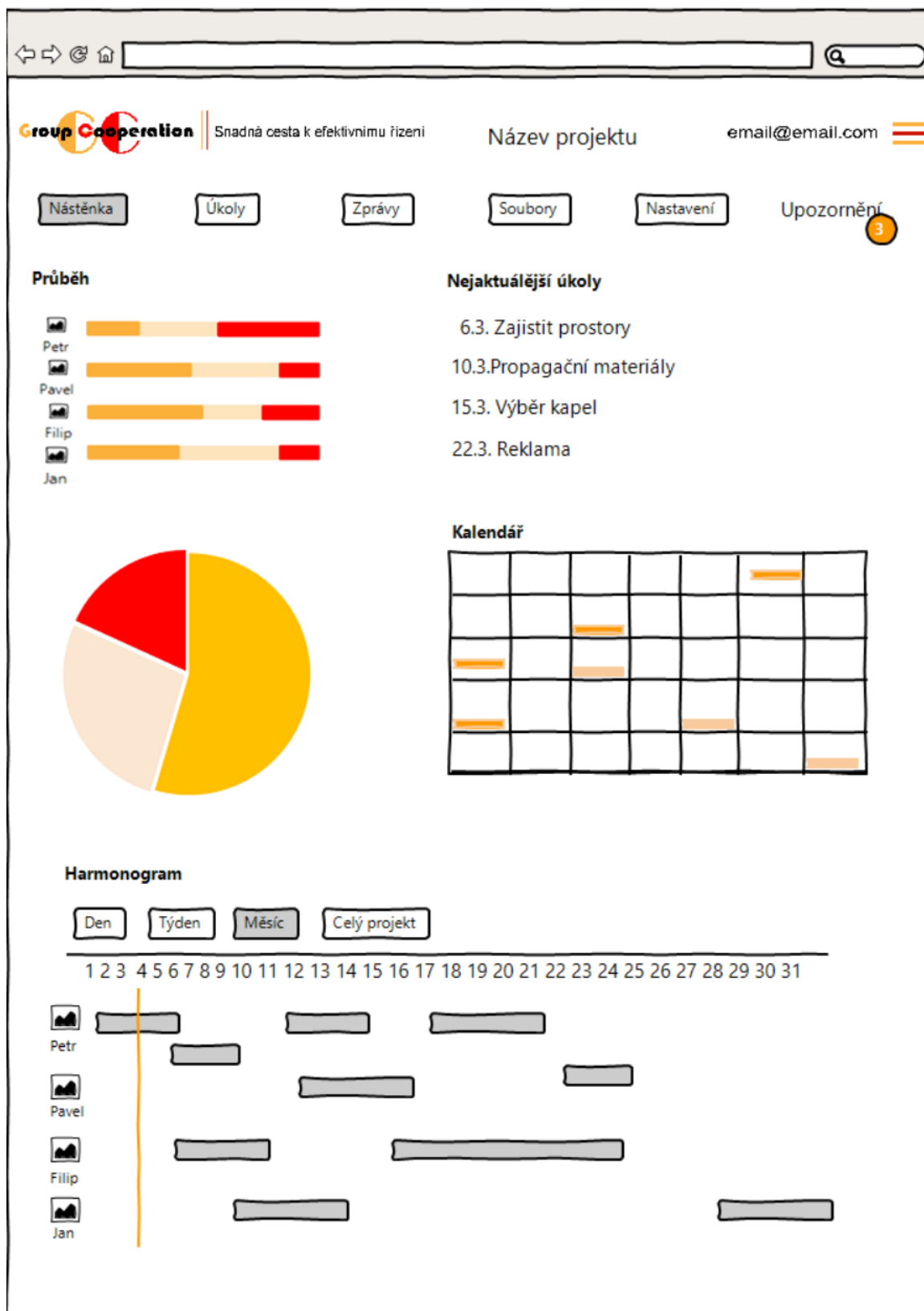
Tlačítko Upozornění zobrazuje notifikace (připomenutí) pro přihlášeného uživatele:

- Přidání do nového projektu – možnost přijmout a odmítnout.

- Blížící se termíny zahájení nebo ukončení úkolů.
- Termíny osobních podúkolů, které si uživatel připojí na svůj úkol.
- Přidání nového úkolu, kde je uživatel vedený jako řešitel.

Tlačítko Upozornění je přístupné i na detailu projektu.

2.4.3 Nástěnka



Obr. 38 Wireframe - Nástěnka (Vlastní)

Po kliknutí na název projektu na domovské stránce se uživatel dostane na nástěnku daného projektu. Na nástěnce je koláčový graf celkového průběhu projektu, resp. plnění úkolu. Zobrazuje tři výseče – procentuální zobrazení stavů úkolů – Zadaný (červená barva), Rozpracovaný (světlá barva) a Vyřešený (oranžová barva). Po kliknutí na graf se uživatel dostane na záložku úkoly, která bude zobrazovat úkoly uživatele, na kterého kliknul.

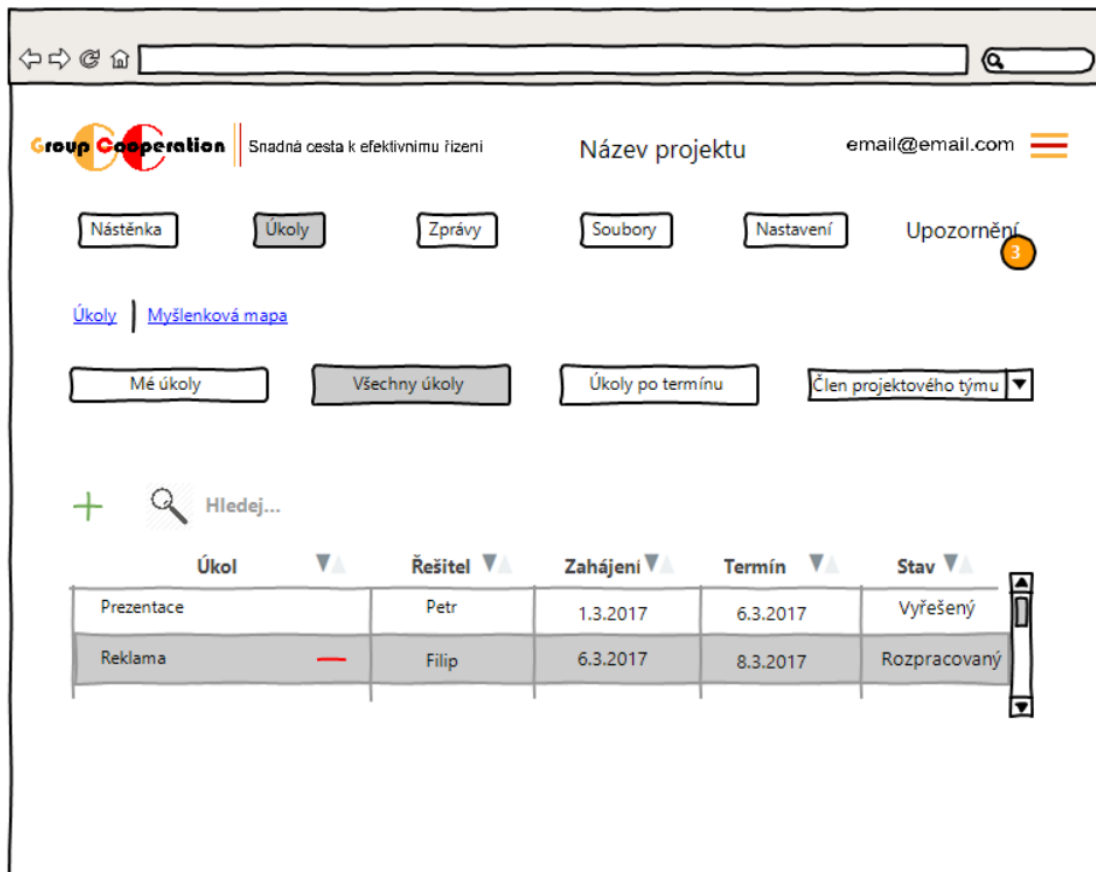
Následující graf zobrazuje totéž, ale ve vztahu k jednotlivým členům projektového týmu a jím přiděleným úkolům. 100% vždy představuje všechny zadané úkoly. Po kliknutí na graf se uživatel dostane na záložku úkoly, která bude zobrazovat všechny úkoly v projektu.

Na nástěnce je umístěn také kalendář. Jsou v něm vyznačeny termíny dokončení jednotlivých úkolů. Barevně jsou úkoly přihlášeného uživatele a ostatní. Po najetí myši na úkol se zobrazí malý detail - název, řešitel, datum zahájení a ukončení a stav. Po kliknutí na úkol se zobrazí detail úkolu.

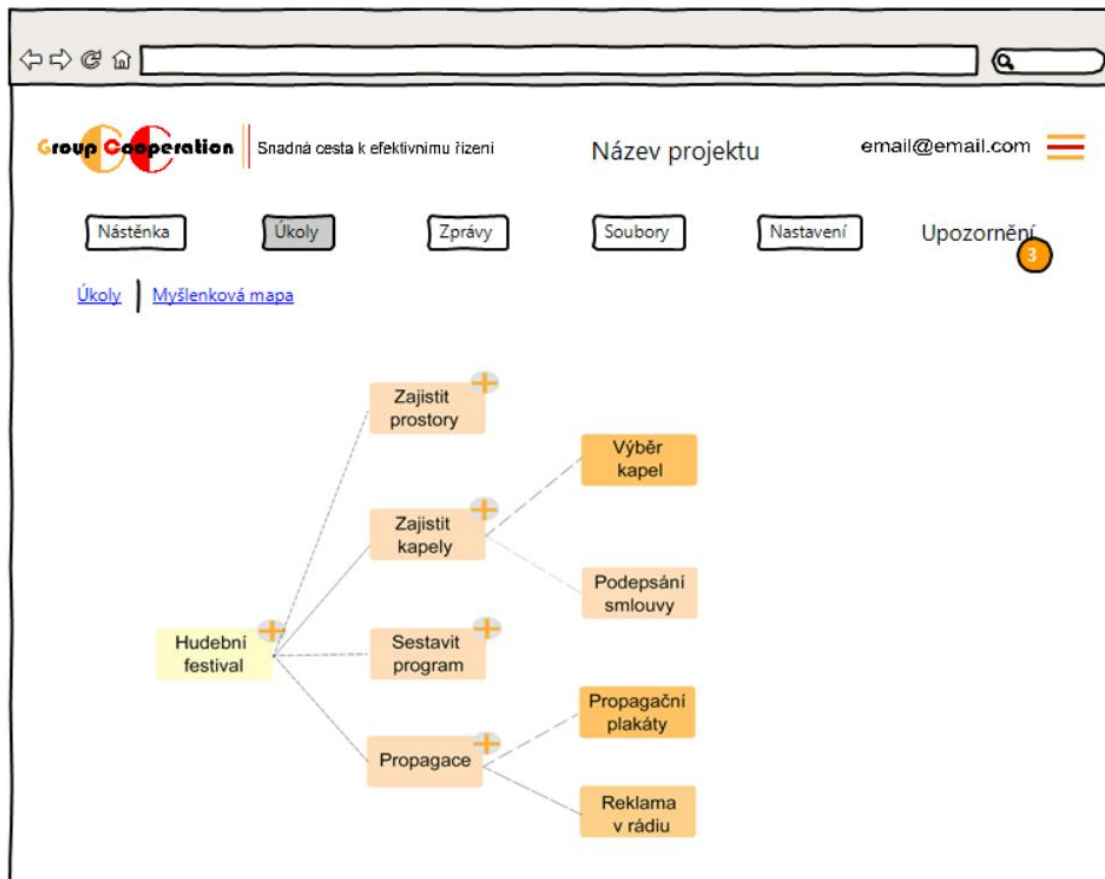
Vpravo nahoře se generují nejaktuálnější úkoly s podúkoly přihlášeného uživatele. Po kliknutí na úkol se zobrazí detail úkolu.

V poslední části nástěnky je umístěn harmonogram projektu – Gantův diagram. Stejně jako v kalendáři jsou barevně odlišeny úkoly přihlášeného uživatele a ostatních, zvýrazněné jsou též úkoly po termínu. Po najetí myši na úkol se zobrazí malý detail - název, řešitel, datum zahájení a ukončení a stav. Po kliknutí na úkol se zobrazí detail úkolu. Harmonogram lze zobrazit pomocí odpovídajících tlačítek pro daný den, týden, měsíc či pro celou dobu trvání projektu.

2.4.4 Úkoly



Obr. 39 Wireframe - Úkoly (Vlastní)



Obr. 40 Wireframe - Myšlenková mapa (Vlastní)

Stránka Úkoly je rozdělena na dvě záložky – Myšlenková mapa a Úkoly. První záložka zobrazuje strukturu projektu, tj. zadané úkoly jako myšlenkovou mapu.

Záložka Úkoly nabízí přehled úkolů v tabulce. Tlačítka „Všechny úkoly“, „Mé úkoly“, „Úkoly po termínu“ a „Úkoly konkrétního řešitele“ umožňují filtrovat v zadaných úkolech dle určené charakteristiky. Po kliknutí na „Úkoly konkrétního řešitele“ se zobrazí výběr z členů projektového týmu.

Tabulka eviduje název úkolu, řešitele, datum zahájení, termín ukončení a stav. Jednotlivé řádky tabulky lze řadit dle jednotlivých sloupců přes ikonu šipek v záhlaví. Přes ikonu „plus“ lze přidávat další úkol. Po najetí na řádek s konkrétním úkolem se zobrazí ikona „mínus“, která umožní úkol smazat. Přes kolonku hledej, je možné vyfiltrovat konkrétní úkol. Kliknutím na úkol se uživatel dostane na detail úkolu.

Druhá záložka zobrazuje úkoly jako přehlednou tabulku. Myšlenková mapa zobrazuje jednotlivé úkoly jako zaoblené obdélníky. Projekt je možno roztrždit na části a ty pak přes tlačítko plus rozložit na jednotlivé úkoly. Po najetí myši na jednotlivé obdélníky se zobrazí okno s detailem – název, řešitel, datum zahájení a termín ukončení zároveň je mu umožněno úkol smazat – zobrazí se ikona „mínus“. Proklikem se uživatel dostane do detailu úkolu.

2.4.4.1 Detail úkolu

Název úkolu

Úkol

Zařazení

Řešitel

Popis

Doba trvání úkolu od 1/3/2017 do 6/3/2017

Stav

Soubory +

Soubor	Autor	Popis
info.pdf	Pavel	Informace k prezentaci
prezentace.pptx	Pavel	Finální verze

Podúkoly + -

- Přečíst soubor info.pdf 1/3/2017
- Vybrat šablonu 1/3/2017
- Zařídít 5/3/2017

Poznámky | Historie

Uložit Storno

Obr. 41 Wireframe - Detail úkolu (Vlastní)

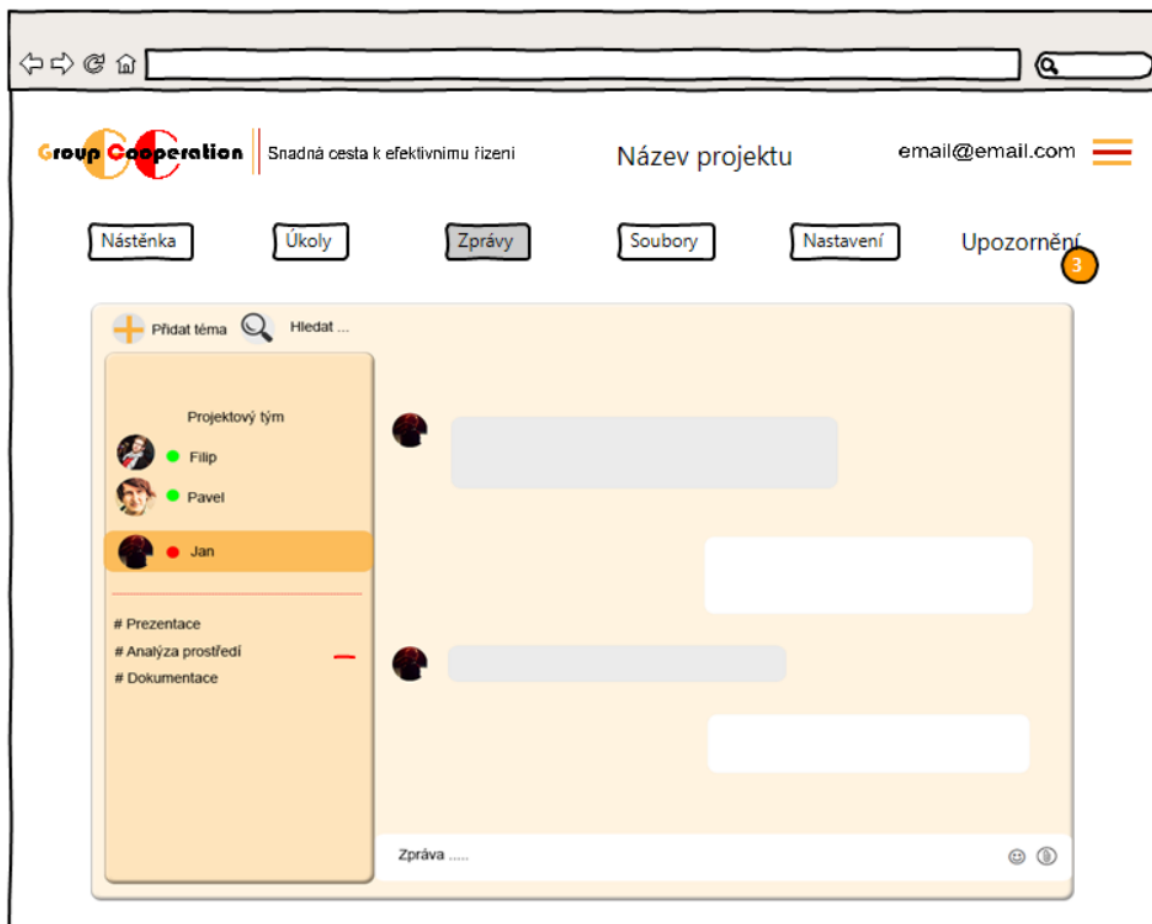
Po kliknutí na ikonu „plus“ nebo kliknutí na název úkolu v záložce Úkoly se uživatel dostane na detail úkolu. Lze vyplnit název úkolu a popis. Z combo boxu lze vybrat část, do které úkol patří, tzv. Zařazení. Dále se z combo boxu vybere řešitel daného úkolu a nastaví datum zahájení a termín dokončení. Tyto obecné charakteristiky jsou povinná pole, bez kterých nelze úkol uložit. Řešitel může z combo boxu vybrat stav úkolu, který je ve výchozí hodnotě ve stavu „Zadaný“. Řešitel si může k danému úkolu přes ikonu „plus“ (v daném bloku) přidat své osobní podúkoly s termíny dokončení. Dané podúkoly lze označit zaškrtnutím check boxu jako hotové. Při najetí myši na podúkol je možné jej odstranit přes ikonu „mínus“. Dále je zde možnost přidat soubor přes ikonu „plus“ (v daném bloku). Soubory jsou zobrazeny v tabulce – eviduje se název souboru, autor a popis. Soubor lze stáhnout kliknutím na název souboru. Dále je možnost soubory řadit dle názvu a autora přes ikonu „šipek“ v záhlaví tabulky příslušného sloupce.

Vlevo dole je okno, které je rozdělené na dvě záložky – Poznámky a Historie. Do poznámek lze vložit jakýkoli další komentář, popis, atd. Historie se generuje

automaticky – zaznamenává všechny změny týkající se daného úkolu – editaci, změnu stavu, přidání uživatele, souboru, aj.

Změny na detailu úkolu lze uložit přes tlačítko „Uložit“, popřípadě zrušit a zavřít okno přes tlačítko „Storno“.

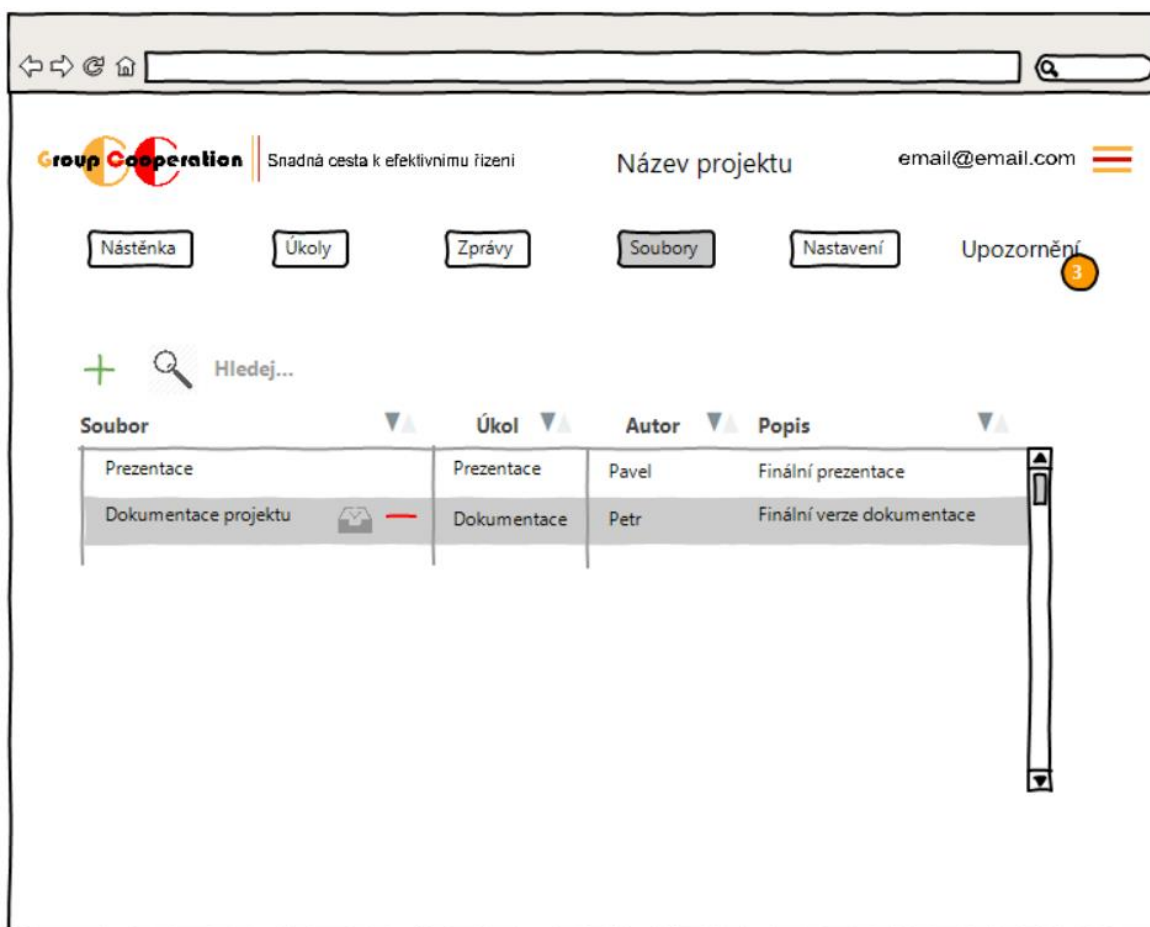
2.4.5 Zprávy



Obr. 42 Wireframe - Zprávy (Vlastní)

Záložka zprávy umožňuje rychlou komunikaci mezi členy projektového týmu. Na levé straně si uživatel zvolí, s kým chce komunikovat. Buď s celým projektovým týmem najednou, nebo si vybere konkrétního uživatele, popřípadě konverzační téma. Po vybrání se zobrazí v pravé části detail komunikace a pole pro zadání nové zprávy. Zprávu odešle uživatel pomocí klávesy Enter. Přes ikonu „plus“ lze přidávat nové konverzační vlákno a vytvořit tak tematickou konverzaci. Uživateli se zobrazí okno pro přidání – název konverzace a určení, kteří členové týmu mohou danou komunikaci využívat. Pomocí tlačítka Hledat lze vyhledávat v konkrétní komunikaci.

2.4.6 Soubory



Obr. 43 Wireframe - Soubory (Vlastní)

Záložka soubory umožňuje správu dokumentů. Tabulka eviduje název souboru, autora a popis souboru. Soubory lze přidávat pomocí ikony plus – zobrazí se okno pro přidání souboru. Soubor lze připojit k jednotlivým úkolům v projektu. Po najetí myši na konkrétní řádek souboru se zobrazí ikona pro stáhnutí a smazání souboru. Soubor lze odstranit přes ikonu „mínus“, uživatel bude upozorněn, že soubor se odstraní i z detailu úkolu. Jednotlivé řádky tabulky lze řadit dle jednotlivých sloupců přes ikonu „šipek“. Připojené soubory se zobrazí v detailu úkolu a naopak připojené soubory v detailu úkolu se generují do této záložky.

2.4.7 Nastavení

Group Cooperation | Snadná cesta k efektivnímu řízení

Název projektu email@email.com

Nástěnka Úkoly Zprávy Soubory Nastavení Upozornění 3

Název projektu

Doba trvání projektu od 1/1/2017 do 1/3/2017

Popis

Uložit

Členové projektového týmu +

Jméno	E-mail
Filip	Denisa@gmail.com
Pavel	Pavel@gmail.com
Petr	Petr@gmail.com
	Jan@gmail.com

Označit projekt za hotový

Obr. 44 Wireframe - Nastavení (Vlastní)

Záložka nastavení slouží pro editaci základních údajů projektu a správu členů projektového týmu. Obsah záložky Nastavení odpovídá formuláři pro přidání nového projektu z Úvodní stránky přihlášeného uživatele. Uživatel vyplní název, dobu trvání a popis projektu. Pomocí tlačítka Plus lze přidat nového člena projektového týmu – pro přidání je potřeba pouze email uživatele. Na základě zadaného emailu a potvrzení, danému uživateli přijde email s informací o přidání do projektu. Pokud se uživatel přihlásí a potvrdí přidání do projektu v tabulce se vyplní jméno člena projektového týmu. Uživatele lze řadit dle jména, emailu přes ikonu „šipek“. Po najetí myši na konkrétního člena je uživateli umožněno daného uživatele smazat. Dále lze zaškrtnutím check boxu označit projekt za hotový – lze pouze v případě, že jsou všechny úkoly ve stavu Vyřešeno.

2.5 Práva a role

Ve verzi Free existuje pouze role Člen projektového týmu – všichni uživatelé mají stejná práva.

Ve verzi Standard a Profi lze jednotlivým členům týmu přiřadit role, které určují jejich práva. Ve verzi Standard a Profi (pokud není v implementačním projektu sjednáno jinak) existují tři role:

Uživatel U1

Člen projektového týmu, který nemá právo editovat projekt. Vidí veškeré základní údaje o projektu včetně harmonogramu. Vidí úkoly, které jsou mu přiděleny a ty může rozkliknout a editovat – přidávat soubory, popis a označit jako vyřešený.

Nástěnka – má právo vidět pouze Harmonogram, Kalendář, kde jsou označeny jen jeho úkoly a pouze celkový graf průběhu projektu.

Myšlenková mapa a Úkoly – Úkoly, u kterých je označen jako řešitel má právo číst a editovat, ostatní může pouze číst. Nemá právo přidávat ani mazat jednotlivé úkoly.

Harmonogram – má právo vidět celý harmonogram. Podobně jako u Myšlenkové mapy a úkolů má právo číst a editovat pouze vlastní tasky, ostatní může pouze číst.

Soubory – Má právo přidávat a mazat soubory pouze ke svým úkolům nebo celkově na celý projekt. Vidí pouze soubory, které vložil nebo jsou označeny jako veřejné.

Zprávy - Může používat veškeré funkcionality.

Nastavení – Má právo pouze číst.

Uživatel U2

Má všechna práva jako Uživatel U1. Navíc má právo přidávat další úkoly v záložce Úkoly.

Uživatel U3

Má stejná práva jako uživatel U2. Navíc má právo číst a editovat veškerou agendu projektu. Má právo přidávat a mazat další členy vývojového týmu.

3. Marketingový mix

3.1. Produkt

V rámci diplomové práce je navržena webová aplikace Group Cooperation ve zkratce GC. Vzhledem k novému produktu bylo nutné vymyslet název a vyhotovit logo, které bude aplikaci představovat.

Název Group Cooperation vznikl spojením dvou slov „Skupina“ a „Spolupráce“ což vystihuje cíl webové aplikace. Ve spojení s názvem produktu se pojí heslo: „Snadná cesta k efektivnímu řízení“. Pro logo byly použity dvě barvy, které budou také použity pro celou webovou prezentaci, a to barva červená a oranžová. Logo je zobrazeno na obrázku č.45.



Obr. 45 Logo Group Cooperation (Vlastní)

Webová aplikace Group Cooperation ve zkratce GC je nástrojem pro řízení projektů malého rozsahu, který zajišťuje veškerou kooperaci mezi členy projektového týmu. Aplikace GC zajišťuje činnosti probíhající při řízení projektu – plánování, řízení, správa dokumentů.

Aplikace je nabízena ve třech různých variantách:

- Free – vhodná zejména pro malé projekty, např. školní skupinové projekty
- Standard – vhodná pro malé podniky, start up, aj.
- Profi - vhodná pro střední projekty

GC free	GC Standard	GC Profi
Zdarma	990 Kč/měsíc	2990 Kč/měsíc
Objem dat: 500 MB	Objem dat: 5 GB	Objem dat: 200 GB
Maximální počet uživatelů na jeden projekt: 4	Maximální počet uživatelů na jeden projekt: 10	Maximální počet uživatelů na jeden projekt: 20
Neomezené množství projektů	Neomezené množství projektů	Neomezené množství projektů
Automatická aktualizace aplikace zdarma	Automatická aktualizace aplikace zdarma	Automatická aktualizace aplikace zdarma
Online podpora	Online podpora	Online podpora
Moduly:	Moduly:	Moduly:
Nástěnka	Nástěnka	Nástěnka
Myšlenková mapa	Myšlenková mapa	Myšlenková mapa
Úkoly	Úkoly	Úkoly
Soubory	Soubory	Soubory
Zprávy	Zprávy	Zprávy
	Role a práva	Role a práva
		Finanční řízení projektu

Obr. 46 Verze aplikace Group Cooperation (Vlastní)

Moduly:

Nástěnka nabízí přehled o vývoji daného projektu prostřednictvím přehledných grafů průběhu projektu, kalendář s vyznačenými termíny ukončení jednotlivých úkolů a harmonogram, který znázorňuje posloupnosti projektu v čase a ukazuje jeho průběh. Na nástěnce jsou umístěny i nejaktuálnější úkoly a osobní podúkoly, které si uživatel může přiřadit k jednotlivým úkolům.

Myšlenková mapa je nástroj umožňující vytvářet logickou strukturu projektu, pomocí rozdělování na části a podřízené úkoly. Druhá část **Úkoly** nabízí seznam vytvořených úkolů v přehledné tabulce.

Soubory umožňují správu dokumentů potřebných k projektu.

Zprávy umožňují rychlou komunikaci mezi členy projektového týmu. Lze komunikovat s celou skupinou najednou, popřípadě s jednotlivými členy. Lze vytvářet nová konverzační témata k jednotlivým úkolům, aj.

Role a práva umožňují nastavit role uživatelů v projektovém týmu a nastavit práva užívání – editovat a číst. Ve verzi Free role neexistují, všichni uživatelé mají stejná práva.

Finanční řízení umožňuje plánovat finanční zdroje projektu.

3.2 Cena

Aplikace GC je ve verzi Free navržena zejména pro školní projekty pro střední a vysoké školy, proto je přístupná zcela zdarma. Ve verzi Standard je cena stanovena na 990,- Kč/měsíc a za verzi Profi činí 2990 Kč/měsíc. Profi je možné upravit dle požadavků zákazníka (objem dat, maximální počet uživatelů, funkcionality na míru, správa číselníků). Cena se tak bude odvíjet od vzniklého Implementačního projektu.

Ceny jsou stanoveny ve srovnáním s konkurencí a vzhledem k cenám serverových a webhostingových služeb.

3.3 Distribuce

Aplikace GC je nabízena cílovým uživatelům prostřednictvím internetu. Bude zpřístupněna na vlastní doméně. Ve Free verzi je aplikace přístupná po zaregistrování a splnění všeobecných podmínek užívání. Ve verzi Standart a Profi je aplikace přístupná po zaregistrování, splnění všeobecných podmínek užívání a uhrazení ceny produktu.

Bude zajištěn kvalitní webhosting, který zaručuje spolehlivost a bezpečnost, kvůli zamezení jakýchkoli virových aktivit, nevyžádaných reklam, aj. Bude využito zabezpečeného připojení přes HTTPS.

Bude provedena SEO analýza a následně vytvořená strategie, která zabezpečí dostatečnou viditelnost a dostupnost na webu.

3.4 Propagace

Propagace je popsána v následující kapitole 4. Marketingový komunikační mix.

4. Marketingový komunikační mix

WWW prezentace

Mezi základní formy marketingové komunikace bude pro aplikaci GC patřit webová prezentace, tedy webové stránky produktu.

Na webových stránkách se uživatelé dočtou veškeré informace o aplikaci. Bude zde k dispozici online chat pro zodpovězení jakýchkoli dotazů od uživatelů či potenciálních zákazníků. Uživatelé se zde mohou zaregistrovat a prakticky okamžitě začít používat aplikaci ve verzi Free.

Texty na webových stránkách budou mít informativní charakter, zároveň budou mít za úkol zaujmout a prodat. Webová prezentace bude stejně jako aplikace laděná do oranžové a červené barvy. Návrh úvodní obrazovky www stránek naleznete v příloze 2.

Blog

V rámci webové prezentace bude zpřístupněn i blog, kde se budou zveřejňovat novinky ze světa projektového řízení.

Sociální síť

Na sociální síti Facebook bude zřízen profil s názvem Group Cooperation. Profil bude informovat o existenci aplikace a zároveň zde budou uveřejňovány akce, workshopy, přednášky, novinky a informace ze světa projektového řízení. Profil na sociální síti bude propagován pomocí placených bannerů zaměřených na cílovou skupinu.

Aplikace GC bude prezentována i na sociálních sítích Twiter a Instagram. Budou zde uveřejňovány zejména fotky z akcí přednášek a workshopů.

Přednášky

V rámci propagace budou realizovány přednášky, zejména pro vysokoškolské studenty, kde bude kladen důraz především na verzi Free, která je zdarma. Aplikace bude prezentována i na studentských veletrzích a fórech.

Internetové bannery

Budou vytvořeny bannery upozorňující na existenci aplikace GC. Bannery budou umístěovány do diskuzních fór a webových stránek zaměřující se především na management a projektové řízení.

Informační brožury a letáky

Bude vytvořena informační brožura aplikace GC, která bude distribuována zejména na střední a vysoké školy v tištěné podobě.

Přímý marketing

Prostřednictvím e-mailů bude distribuována informační brožura potenciálním zákazníkům – zejména na střední a vysoké školy a malým firmám.

Pro stávající zákazníky bude využíváno přímého marketingu hlavně k dosažení maximální spokojenosti uživatelů. Budou upozorňovány na updaty aplikace, na nové funkcionality, dotazníky spokojenosti, popřípadě na různé akce.

5. Analýza prostředí

5.1 Porterův model pěti konkurenčních sil

Pro zmapování konkurenčního prostředí produktu GC je třeba soustředit se na pár základních otázek dle Porterova modelu pěti konkurenčních sil

Kdo jsou stávající konkurenti?

Aplikací pro řízení projektů je velké množství, převládají ale zahraniční produkty. Dále jsou popsány tři významnější české webové aplikace pro řízení projektů.

Projektově.cz s.r.o.

Produktem této firmy je webová aplikace s názvem Projektově.cz přístupná na adrese <https://www.projektove.cz/> a umožňuje:

- zadávání, sdílení a delegování úkolů,
- myšlenková mapa,
- emailová notifikace a upozornění,
- správa dokumentů,
- evidence stráveného času,
- hlídání rozpočtů,
- česká lokalizace,
- historie komunikace.

Projektově.cz je nabízené ve třech variantách: Pro dva pro jednotlivce, Start pro malé týmy a Standard pro firmy.

Pro Dva <i>pro jednotlivce</i>	Start <i>pro malé týmy</i>	Standard <i>pro firmy</i>
300 Kč <i>/ 2 uživatelé / měsíc</i>	990 Kč <i>/ 5 uživatelů / měsíc</i>	1 990 Kč <i>/ 12 uživatelů / měsíc</i>
<input checked="" type="checkbox"/> Neomezené úkolů	<input checked="" type="checkbox"/> Neomezené úkolů	<input checked="" type="checkbox"/> Neomezené úkolů
<input checked="" type="checkbox"/> Neomezené projektů	<input checked="" type="checkbox"/> Neomezené projektů	<input checked="" type="checkbox"/> Neomezené projektů
<input checked="" type="checkbox"/> Objem uložených dat: 2 GB	<input checked="" type="checkbox"/> Objem uložených dat: 5 GB	<input checked="" type="checkbox"/> Objem uložených dat: 12 GB
<input checked="" type="checkbox"/> Live chat podpora / helpdesk	<input checked="" type="checkbox"/> Live chat podpora / helpdesk	<input checked="" type="checkbox"/> Live chat podpora / helpdesk
<input checked="" type="checkbox"/> Provoz v cloudu	<input checked="" type="checkbox"/> Provoz v cloudu	<input checked="" type="checkbox"/> Provoz v cloudu
<input checked="" type="checkbox"/> Šifrované spojení	<input checked="" type="checkbox"/> Šifrované spojení	<input checked="" type="checkbox"/> Šifrované spojení
<input checked="" type="checkbox"/> Záloha : 1x denně	<input checked="" type="checkbox"/> Záloha : 1x denně	<input checked="" type="checkbox"/> Záloha : 1x denně
<input checked="" type="checkbox"/> Automatické aktualizace softwaru zdarma	<input checked="" type="checkbox"/> Automatické aktualizace softwaru zdarma	<input checked="" type="checkbox"/> Automatické aktualizace softwaru zdarma

Obr. 47 Verze aplikace projektove.cz [28]

Aplikaci Projektově.cz používá přes 2300 uživatelů.

Easy Project

Firma Easy Software s.r.o. nabízí webovou aplikaci Easy Project zpřístupněnou na adrese „<https://www.easypoint.cz>“. Software pro komplexní řízení projektů s rozšířeními pro řízení zdrojů, financí a zákazníků. Aplikace nabízí:

- komplexní řešení projektů,
- rozšíření pro řízení zdrojů, financí a zákazníků,
- možnost cloudového nebo serverového řešení,
- důraz na osobní implementaci a podporu,
- správa projektů a úkolů,
- osobní kalendář s úkoly,
- emailová notifikace,
- optimální využití zaměstnanců,
- kontrola a řízení rozpočtu,
- neomezené aktualizace,
- profesionální implementace,
- technická podpora.

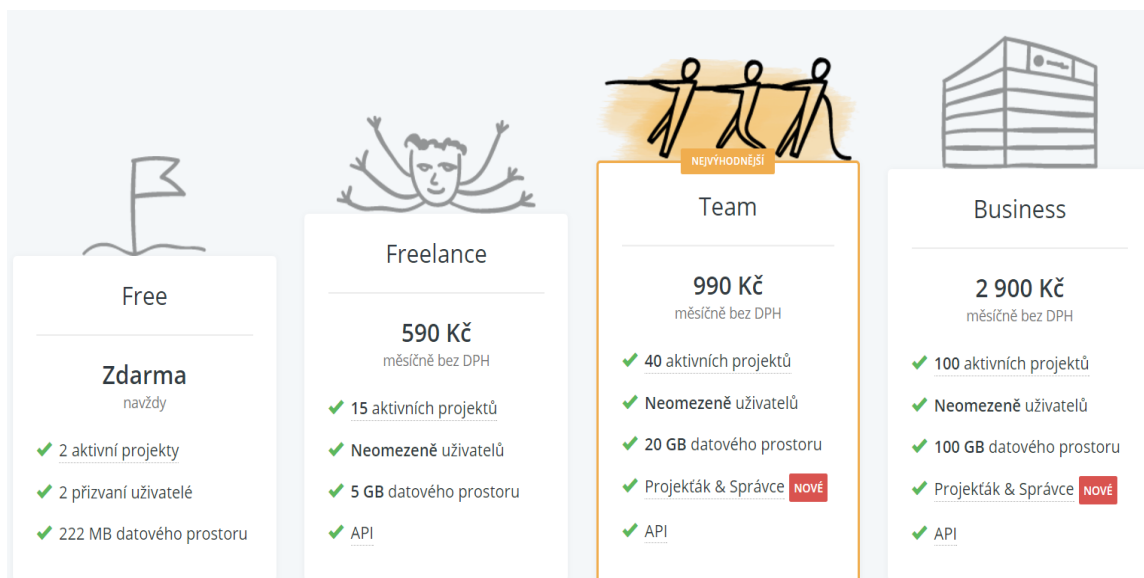
<p>BASIC</p> <p>10 uživ.</p> <p>1 490 Kč / měsíc (účtováno ročně nebo 1 937 Kč měsíčně)</p>	<p>STANDARD</p> <p>20 uživ.</p> <p>2 490 Kč / měsíc (účtováno ročně nebo 3 237 Kč měsíčně)</p>
<p>PROFESSIONAL</p> <p>50 uživ.</p> <p>4 990 Kč / měsíc (účtováno ročně nebo 6 487 Kč měsíčně)</p>	<p>ENTERPRISE</p> <p>50+ uživ.</p> <p>Kontaktujte nás! (nabídneme Vám výhodnou cenu)</p>

Obr. 48 Verze aplikace easyproject.cz [29]

Freelo

Webová aplikace Freelo je zpřístupněná na adrese <https://www.freelo.cz>. Aplikace se zaměřuje na plánování úkolů a komunikaci v týmu. Freelo je nabízené v několika variantách, viz obr. 49. Na rozdíl od předchozích nabízí verzi, která je zcela zdarma. Verze Free ale umožňuje pouze dva aktivní projekty, dva uživatele a pouze 222MB datového prostoru.

Výhodou oproti aplikaci Group Cooperation je možnost sledování výkazů práce a vedení fakturace od základní verze. Další výhodou je napojení Freela přes API na další zákazníkovu aplikaci. Aplikace Freelo umožňuje i ve verzi Tým a Business upravovat role a práva uživatelů. Uživatelé rozdělují na Vlastníka, Projektáky a dále umožňuje přizvat další osobu k projektu, která vidí pouze své vlastní výkazy. Takovou osobou může být například zákazník.



Obr. 49 Verze aplikace freelo.cz [30]

Kdo jsou Potenciální konkurenti?

Vzhledem k tomu, že trh s vývojem webových aplikací je velice produktivní odvětví, existuje velké množství potenciálních konkurentů. Na trhu s webovými aplikacemi neexistují bariéry vstupu.

Kdo jsou dodavatelé?

GC je webová aplikace a pro její provoz je potřeba pronajímat serverové služby a webhostig. Poskytovatelů těchto služeb je na českém trhu mnoho, např. WEDOS Internet, a.s., VSHosting s.r.o., SYSTEM4U s.r.o., a další.

Kdo jsou kupující?

Kupující, či zákazníci pro produkt GC pro verzi Start jsou zejména studenti, pracující na malých projektech v rámci studia, pro které je produkt zdarma. Verze Standard a Profi jsou kupující malé firmy, startupy po celé České Republice.

Co mohou být substituty?

Produkt GC může být nahrazen, kterýmkoli konkurenčním výrobkem, kterých je na trhu velké množství.

5.2 SWOT analýza

Silné stránky	Příležitosti
<ul style="list-style-type: none">• intuitivní ovládání• jednoduchá registrace• online podpora• přehled o vývoji projektu• vývojové grafy• emailová notifikace• verze zdarma• automatické aktualizace• automatické zálohování	<ul style="list-style-type: none">• vlastní aplikační a databázový server• další přídavné moduly• akce a veletrhy s tematikou projektového řízení• rozsáhlý internetový marketing
Slabé stránky	Hrozby
<ul style="list-style-type: none">• nevhodné pro velké projekty• v nižších verzích absence finančního plánování projektu a určení rolí a práv uživatelů	<ul style="list-style-type: none">• neustále rozšiřující se konkurence• rostoucí ceny serverových služeb• nedostatek zákazníků• webové útoky – ztráta dat

6. Shrnutí a závěr

V rámci diplomové práce Návrh realizace a marketingová strategie byla navržena webová aplikace, pro řízení projektu malého rozsahu, zejména pro malé firmy, startupy, semestrální vysokoškolské projekty, aj.

Pro aplikaci byl vymyšlen název Group Cooperation, který vznikl spojením dvou slov „Skupina“ a Spolupráce“, což vystihuje cíl webové aplikace. Ve spojení s názvem se pojí heslo: „Snadná cesta k efektivnímu řízení“, které bude sloužit i jako marketingový slogan.

Aplikace Group Cooperation je navržena tak, aby pokryla nejdůležitější požadavky pro řízení projektu – plánování, rozvržení projektu a následné delegování úkolů, efektivní komunikaci, správu souborů a v neposlední řadě podává aktuální informace o průběhu projektu, prostřednictvím přehledných grafů.

Praktická část diplomové práce je převážně věnována implementačnímu projektu, který bude sloužit pro budoucí implementaci dané aplikace. V implementačním projektu byly definovány funkční a nefunkční požadavky, dále byla věnována velká pozornost drátěným modelům aplikace, tedy jednoduchým návrhům jednotlivých obrazovek systému, s důrazem na aplikování doporučení UX. Síla aplikace je v její jednoduchosti a intuitivnosti.

Implementační projekt se věnuje především základní verzi aplikace Free, která je cílená především pro malé projekty, například vysokoškolské semestrální projekty. Vzhledem k cílové skupině, je webová aplikace zcela zdarma. Studenti nebo začínající startupy mohou řídit své projekty efektivně bez starostí z vysokých nákladů na software. Další verze byly definovány pouze v základních bodech a v budoucnu budou předmětem další analýzy a implementace. Návrh aplikace, UML diagramy, datový model a návrhy wireframů byly konzultovány s analytikem softwaru na míru a s programátorem z firmy Gist, s.r.o.

Druhá část diplomové práce se zabývá Marketingovým a komunikačním mixem webové aplikace. Cílem práce tedy bylo aplikaci nejen navrhnout, ale také definovat, jakým způsobem se aplikace dostane do povědomí cílové skupiny lidí. Marketingový mix detailně popisuje produkt, včetně návrhu loga aplikace. Marketingový komunikační mix určuje hlavní propagační cesty s důrazem na webové stránky, blog, sociální sítě a prezentace na studentských veletrzích a fórech.

Seznam obrázků

Obr. 1 Vodopádový model [1]	4
Obr. 2 Spirálový model [1]	5
Obr. 3 Rational Unified Process [6]	7
Obr. 4 Agilní metodiky vs. tradiční přístupy [1].....	7
Obr. 5 Vývojové fáze FDD [1]	12
Obr. 6 Vývojové fáze TDD [1]	12
Obr. 7 Relace [6].....	17
Obr. 8 UML diagramy [8].....	18
Obr. 9 Aktér (Vlastní).....	22
Obr. 10 Use Case (Vlastní).....	23
Obr. 11 Zobecnění aktéra (Vlastní)	24
Obr. 12 Zobecnění případů užití (Vlastní)	25
Obr. 13 Relace Include (Vlastní)	25
Obr. 14 Relace Extend (Vlastní).....	25
Obr. 15 Analytická třída (Vlastní)	26
Obr. 16 Zpřesňování analytického modelu do návrhového [9].....	28
Obr. 17 Atributy [9]	28
Obr. 18 Typy viditelnosti [6].....	29
Obr. 19 Primitivní datové typy UML [6]	29
Obr. 20 Závislost (Vlastní)	30
Obr. 21 Asociace (Vlastní)	30
Obr. 22 Agregace a Kompozice (Vlastní).....	30
Obr. 23 Generalizace (Vlastní).....	31
Obr. 24 User experience [25]	32
Obr. 25 Maslowova pyramida webdesignu [24]	33
Obr. 26 4P Marketingového mixu [16].....	35
Obr. 27 Produkt [14]	36
Obr. 28 Tvorba cenové strategie [14]	37
Obr. 29 Jednotná marketingová komunikace [16].....	40
Obr. 30 Analýza marketingového prostředí [21].....	44
Obr. 31 Cílený marketing [16]	46
Obr. 32 Use Case model aplikace (Vlastní).....	55
Obr. 33 Analytický model tříd aplikace (Vlastní).....	56
Obr. 34 Návrhový model tříd (Vlastní)	57
Obr. 35 Datový model aplikace (Vlastní).....	58
Obr. 36 Wireframe - Přihlíšení (Vlastní)	59
Obr. 37 Wireframe - Homepage aplikace (Vlastní)	60
Obr. 38 Wireframe - Nástěnka (Vlastní).....	62
Obr. 39 Wireframe - Myšlenková mapa (Vlastní).....	64
Obr. 40 Wireframe - Úkoly (Vlastní).....	64
Obr. 41 Wireframe - Detail úkolu (Vlastní)	66
Obr. 42 Wireframe - Zprávy (Vlastní)	67
Obr. 43 Wireframe - Soubory (Vlastní)	68

Obr. 44 Wireframe - Nastavení (Vlastní).....	69
Obr. 45 Logo Group Cooperation (Vlastní).....	71
Obr. 46 Verze aplikace Group Cooperation (Vlastní).....	72
Obr. 47 Verze aplikace projektove.cz [28]	76
Obr. 48 Verze aplikace easyproject.cz [29]	77
Obr. 49 Verze aplikace freelo.cz [30]	78

Seznam příloh

Příloha č. 1 – Funkční a nefunkční požadavky aplikace

Příloha č. 2 – Návrh webových stránek pro aplikaci

Seznam použité literatury

Monografie:

- [1] KADLEC, Václav. *Agilní programování: metodiky efektivního vývoje softwaru*. Brno: Computer Press, 2004. ISBN 80-251-0342-0.
- [6] 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, 567 s. ISBN 978-80-251-1503-9.
- [7] BOOCH, Grady., James. RUMBAUGH a Ivar. JACOBSON. *The unified modeling language user guide*. Reading Mass.: Addison-Wesley, c1999. ISBN 0-201-57168-4.
- [9] Ing. Pavel Čech, Ph.D. *Úvod do objektového modelování*. (přednáška) Univerzita Hradec Králové: Fakulta informatiky a managementu
- [10] KRAVAL, Ilja. *Analytické modelování informačních systémů pomocí UML v praxi*. 1. vydání © RNDr. Ilja Kraval, 2010, 140 s. ISBN 978-80-254-6986-6
- [12] Ing. Kamil Svoboda, Ph.D. *Objektové modelování I*. (přednáška) Univerzita Hradec Králové: Fakulta informatiky a managementu
- [13] KOTLER, Philip. *Marketing*. Praha: Grada, 2004, 855 s. ISBN 978-80-247-0513-2
- [14] HÁLEK, Vítězslav a kolektiv. *Marketing XXL*. 1.vydání. Bratislava: Donau Media, 2010. 667 s. ISBN 978-80-89364-14-5.
- [16] KOTLER, Philip. *Moderní marketing: 4. evropské vydání*. 1. vyd. Praha: Grada, 2007, 1041 s. ISBN 978-80-247-1545-2.
- [17] ROŠICKÝ S. a kol. *Marketing XXL*. 1. vyd. Bratislava: DonauMedia , 2010, 672 s. ISBN 978-80-89364-34-3.
- [19] CHMEL, Zdeněk. *Propagace, public relations, media*. Vyd. 1. Brno: Ante, 1997. 432 s. ISBN 80-902-4042-9
- [20] PELSMACKER, Patrick de. *Marketingová komunikace*. Praha: Grada, 2003. 581 s. ISBN 80-247-0254-1.
- [22] SOMMERVILLE, Ian. *Softwarové inženýrství*. Brno: Computer Press, 2013. ISBN 978-80-251-3826-7.
- [25] SAFFER, Dan. *Designing for interaction: creating innovative applications and devices*. 2nd ed. Berkeley: New Riders, c2010. Voices that matter. ISBN 978-0-321-64339-1.

Internetové zdroje:

[2] Manifest Agilního vývoje software, <http://agilemanifesto.org> [online], Dostupné z <http://agilemanifesto.org/iso/cs/>

[3] Aliance pro agilní vývoje software, The Agile Alliance [online], Dostupné z <http://www.agilealliance.org>

[4] The Scrum Development Process, popis metodiky Scrum na webových stránkách společnosti Mountain Goat Software [online], Dostupné z <http://www.mountaingoatsoftware.com/scrum>

[5] *Agile Software Development: A gentle introduction* [online]. 2009 [cit. 2016-12-29]. Dostupné z: <http://www.agile-process.org/>

[8] *UML* [online]. 2016 [cit. 2016-12-29]. Dostupné z: <http://www.itnetwork.cz/navrhove-vzory/uml>

[11] *UML tutorials* [online]. [cit. 2016-12-29]. Dostupné z: http://www.sparxsystems.com/resources/uml2_tutorial/

[15] *Marketingový mix 4C* [online]. 2016 [cit. 2016-12-29]. Dostupné z: <https://managementmania.com/cs/marketingovy-mix-4c>

[18] *Propagace (Promotion)* [online]. 2016 [cit. 2016-12-29]. Dostupné z: <https://managementmania.com/cs/propagace>

[21] *SWOT analýza* [online]. 2016 [cit. 2016-12-29]. Dostupné z: <https://managementmania.com/cs/swot-analyza>

[23] *Úvod do webdesignu* [online]. 2012 [cit. 2016-12-29]. Dostupné z: <http://www.inflow.cz/uvod-do-webdesignu-0>

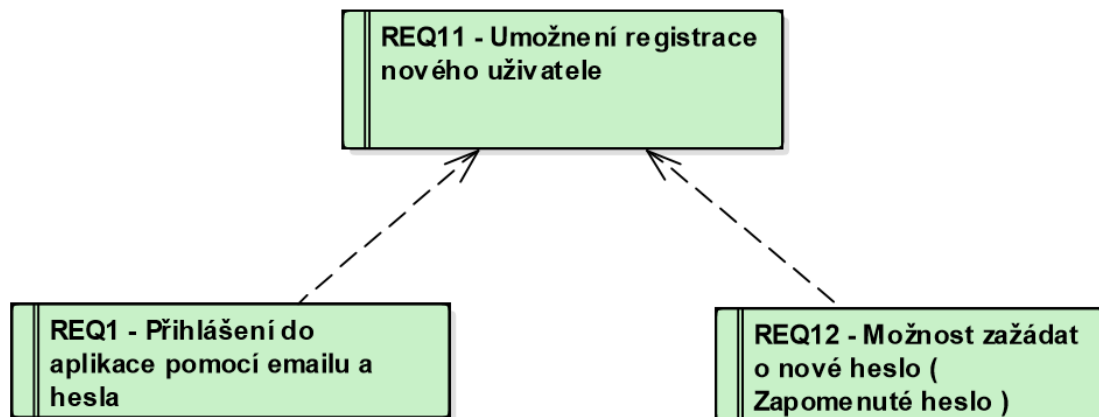
[24] ŘEZÁČ, Jan. *Úvod do webdesignu pro zadavatele* [online]. 2012 [cit. 2016-12-29]. Dostupné z: <http://www.slideshare.net/janrezac/vod-do-webdesignu-pro-zadavatele>

[26] NIELSEN, Jakob. *10 Usability Heuristics for User Interface Design* [online]. 2015 [cit. 2016-12-29]. Dostupné z: <https://www.nngroup.com/articles/ten-usability-heuristics/>

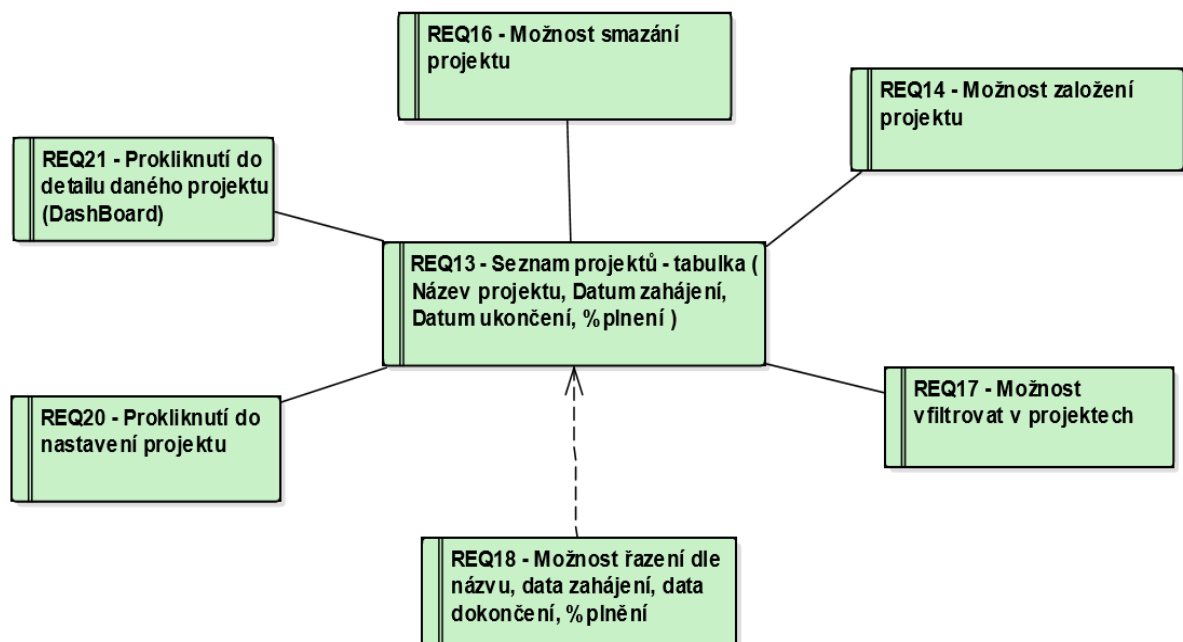
[27] NIELSEN, Jakob. *Top 10 Mistakes in Web Design* [online]. 2011 [cit. 2016-12-29]. Dostupné z: <https://www.nngroup.com/articles/top-10-mistakes-web-design/>

Funkční požadavky

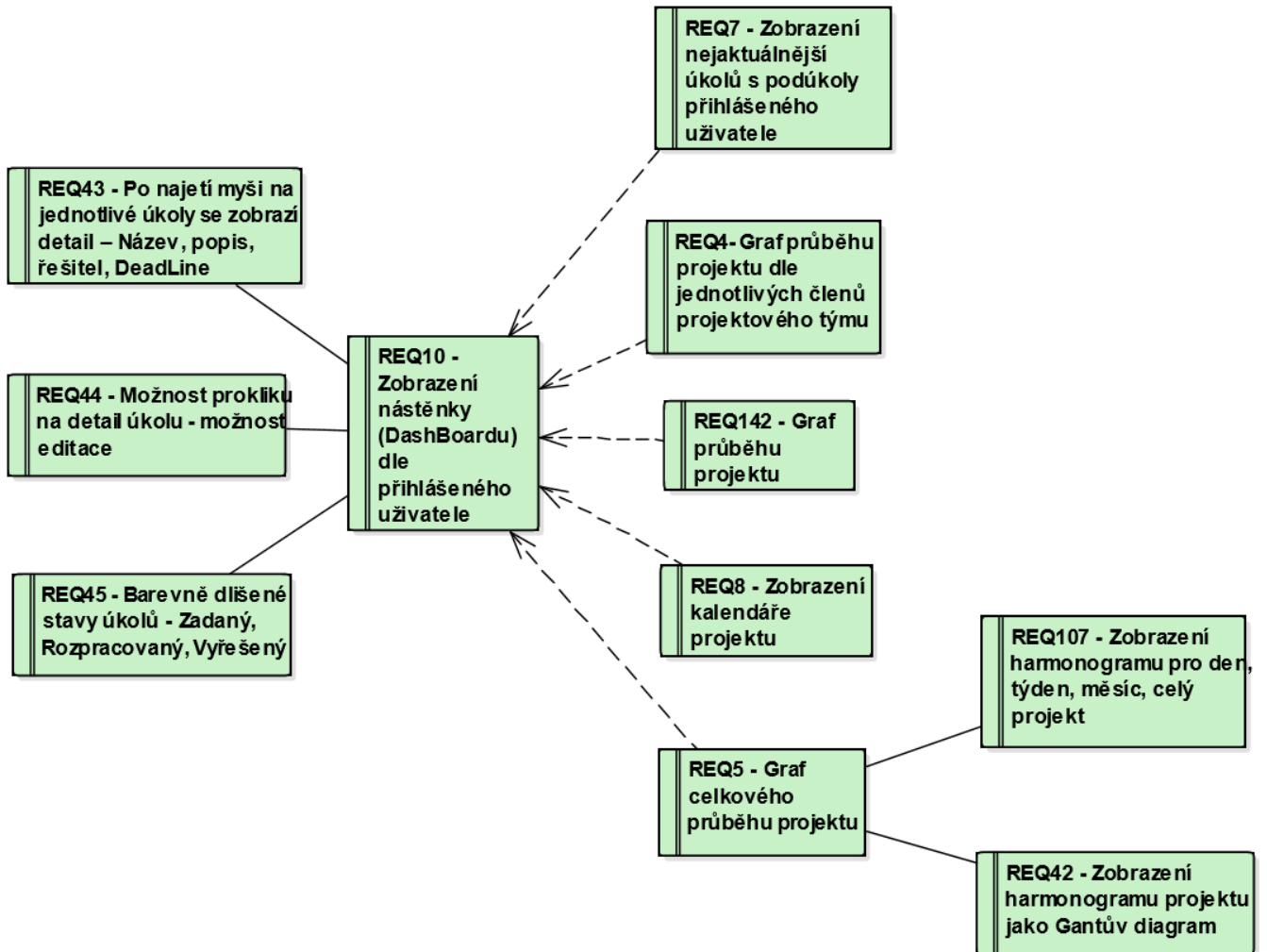
1) Přihlášení do aplikace



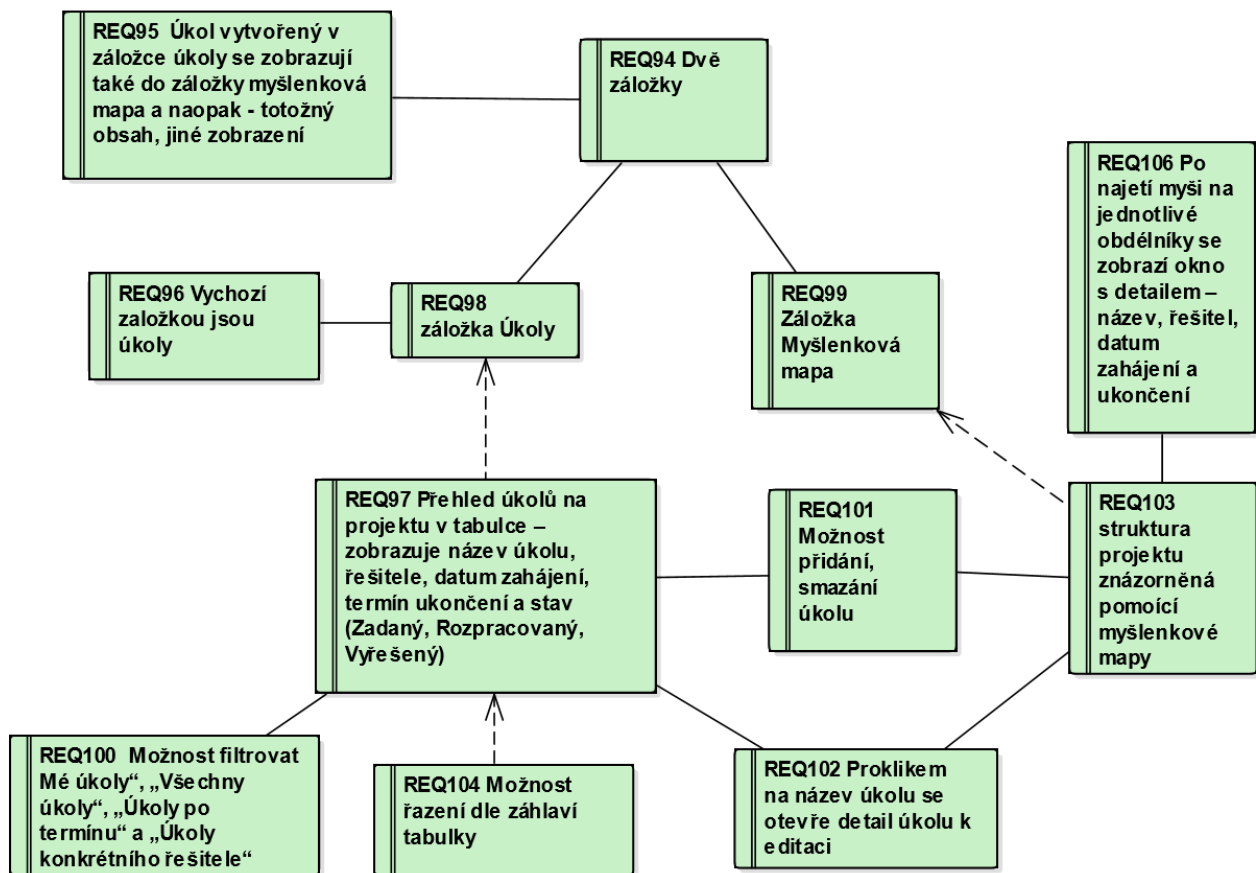
2) Úvodní stránka aplikace



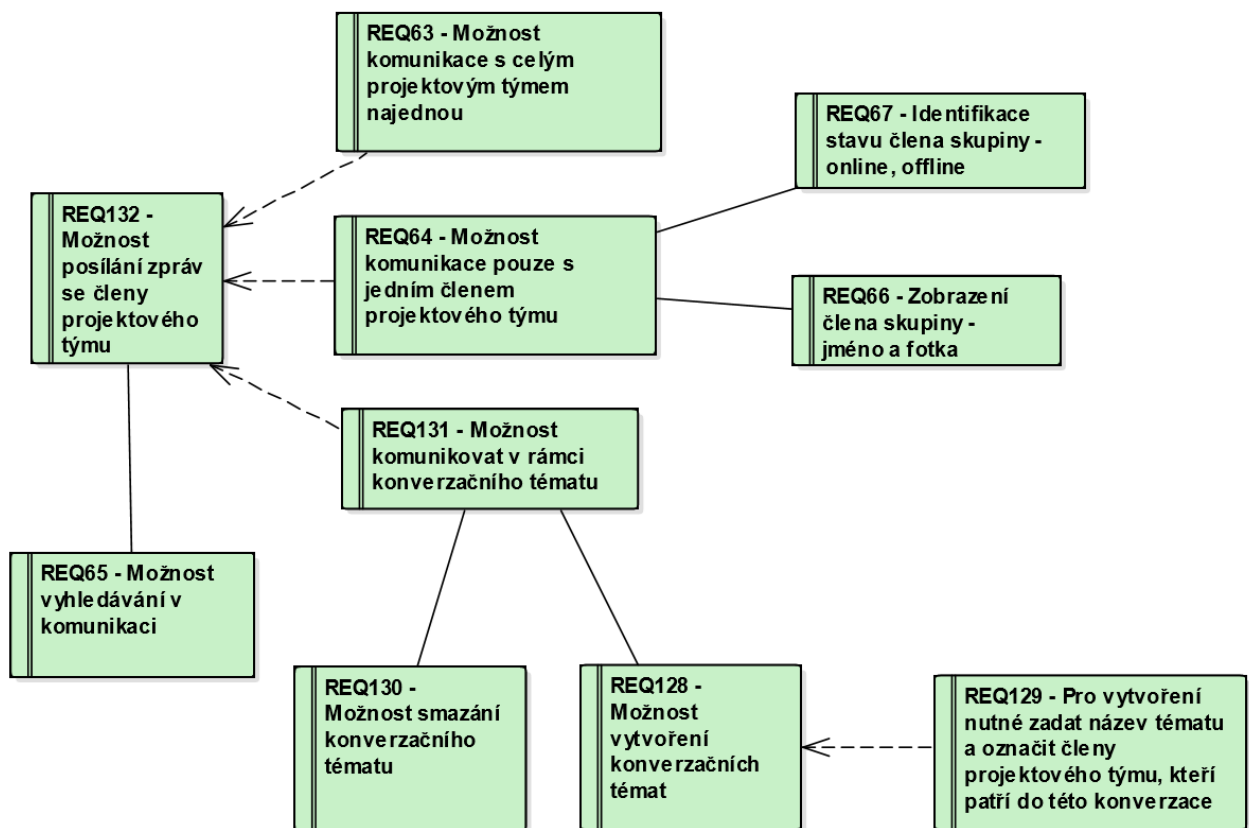
3) Nástěnka



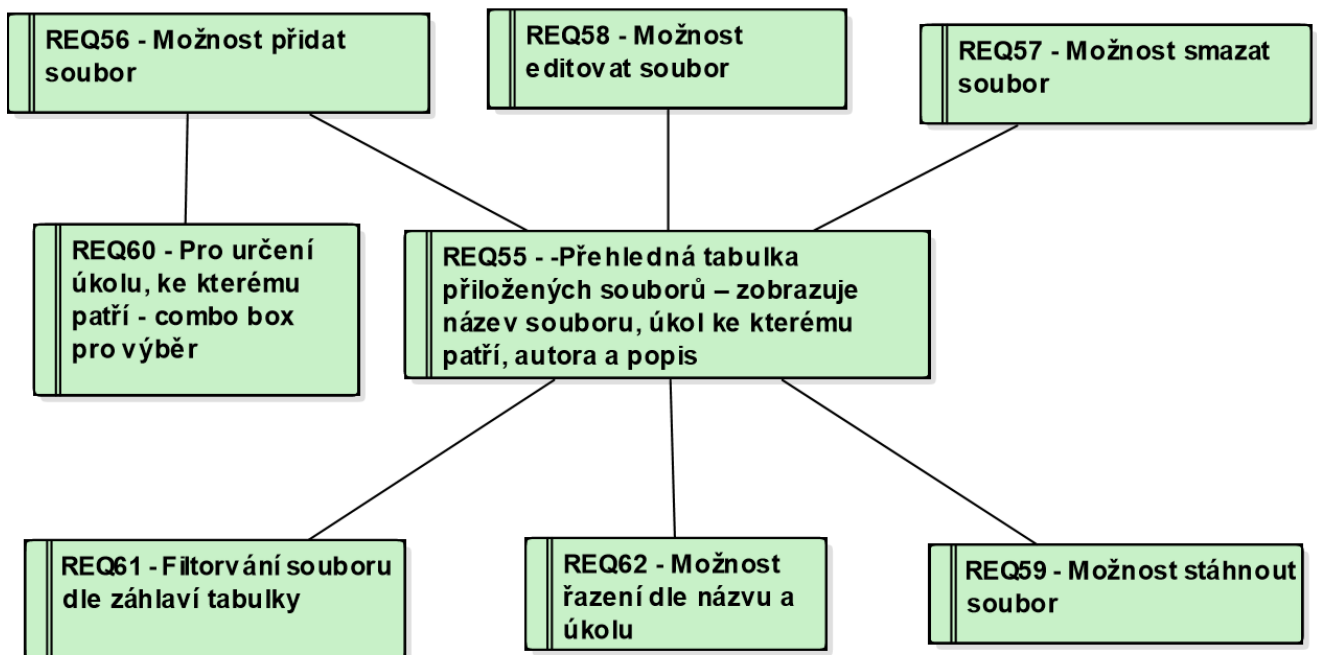
4) Úkoly



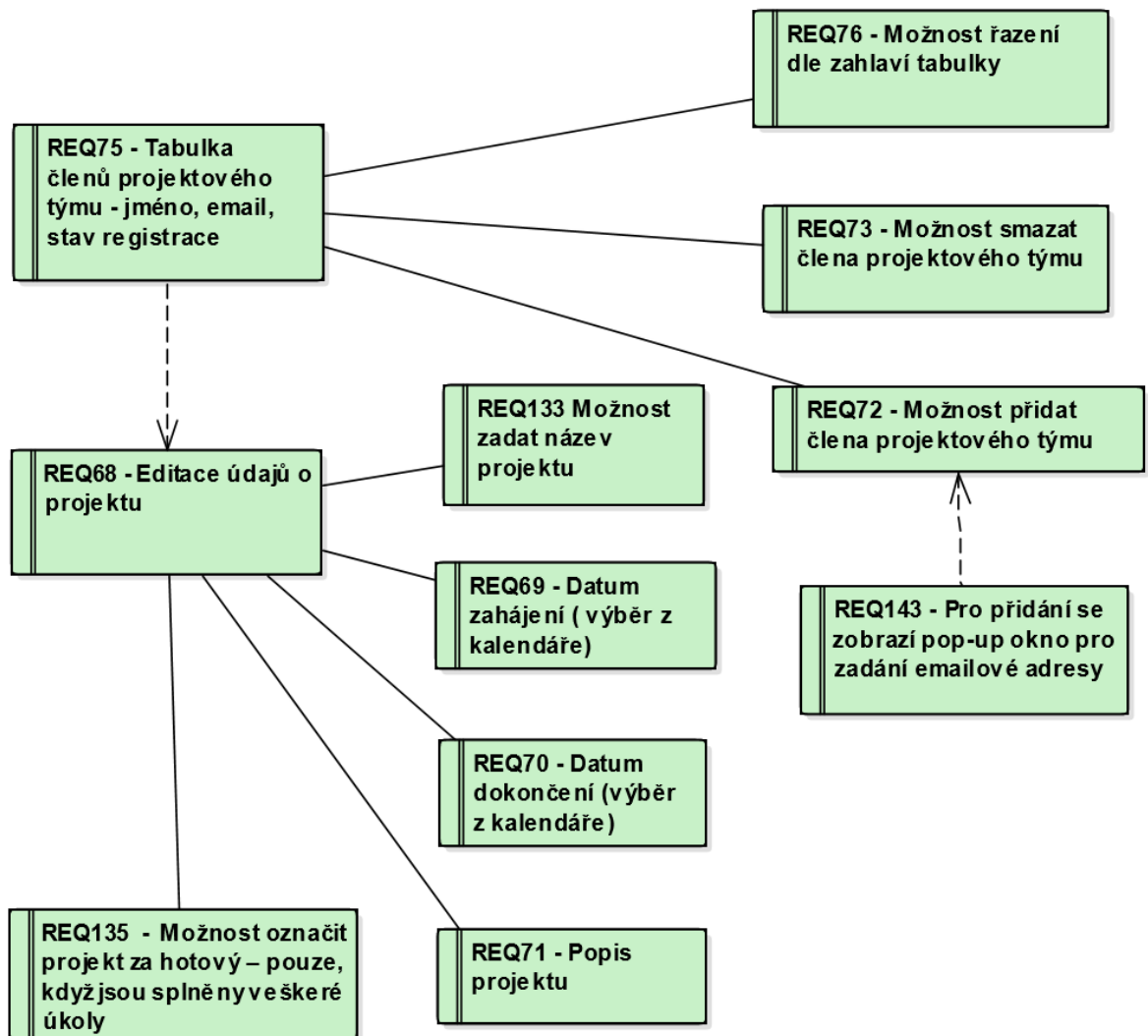
5) Zprávy



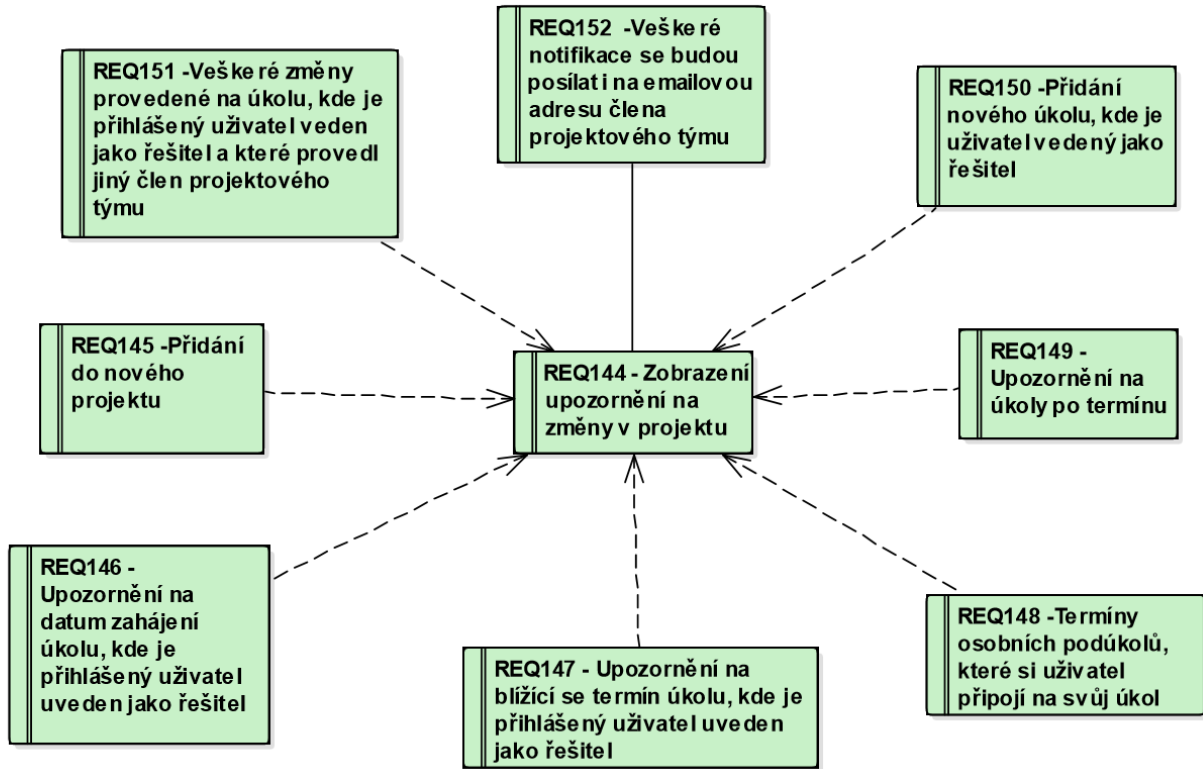
6) Soubory



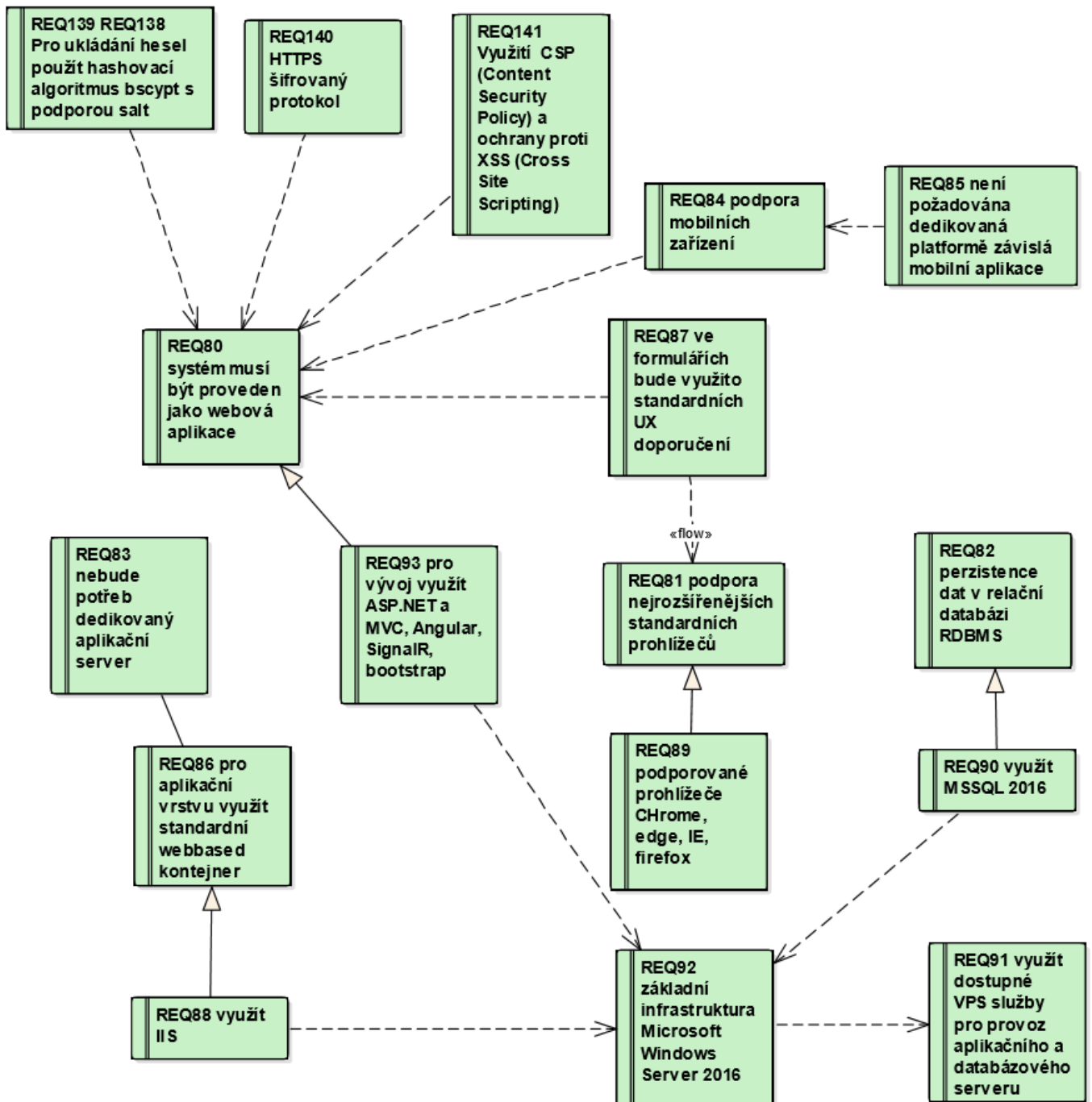
7) Nastavení



8) Notifikace



Non-Funkční požadavky



Grafický návrh webové aplikace

Group Cooperation | Snadná cesta k efektivnímu řízení projektu

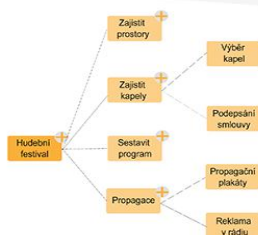
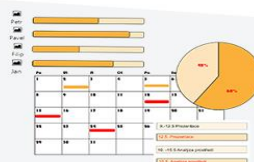
Úvod Aplikace Ceník Blog Kontakt

- Zvýšení efektivity řízení projektu
- Snadné plánování
- Rychlejší komunikace
- Přehled na všech činnostech projektu



Přehled nad průběhem projektu

Exceper sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Quisque porta. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Mauris dolor felis, sagittis at, luctus sed, aliquam non, tellus. Aenean fermentum risus id tortor. Fusce suscipit libero eget elit. Mauris metus. Cum tincidunt scelerisque libero. Mauris dictum facilisis augue.

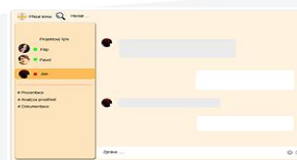


Logická struktura projektu

sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. In laoreet, magna id viverra tincidunt, sem odio bibendum justo, vel imperdiet sapien wisi sed libero. Integer pellentesque quam vel velit. Nullam lectus justo, vulputate eget mollis sed, tempor sed magna. Phasellus et lorem id felis nonumplacemat. Quisque

Komunikace na jednom místě

Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet ut et voluptates repudiandae sint et molestiae non recusandae. Mauris dolor felis, sagittis at, luctus sed, aliquam non, tellus. Aenean fermentum risus id tortor. Fusce suscipit libero eget elit. Mauris metus. Cum tincidunt scelerisque libero. e.



Správa dokumentů

Integer pellentesque quam vel velit. Nullam lectus justo, vulputate eget mollis sed, tempor sed magna. Phasellus et lorem id felis.

Základ
Zdarma

Objem dat: 1 GB

Maximální počet uživatelů na jeden projekt: 4

Neomezené množství projektů

Automatická aktualizace aplikace zdarma

Online podpora

Moduly:
Nástěnka
MindMap
Úkoly
Harmonogram
Soubory
Zprávy

Standard
990 Kč/měsíc

Objem dat: 5 GB

Maximální počet uživatelů na jeden projekt: 10

Neomezené množství projektů

Automatická aktualizace aplikace zdarma

Online podpora

Moduly:
Nástěnka
MindMap
Úkoly
Harmonogram
Soubory
Chat
Role a práva

Profi
2990 Kč/měsíc

Objem dat: 500 GB

Maximální počet uživatelů na jeden projekt: 20

Neomezené množství projektů

Automatická aktualizace aplikace zdarma

Online podpora

Moduly:
Nástěnka
MindMap
Úkoly
Harmonogram
Soubory
Zprávy
Role a práva
Finanční řízení rojektu

O nás

Aenean vel massa quis mauris vehicula lacinia. Duis condimentum augue id magna semper rutrum. Phasellus faucibus molestie nisi. Nullam sit amet magna in magna gravida vehicula. Duis pulvinar. Etiam sapien elit, consequat eget, tristique non, venenatis quis, ante. Fusce tellus odio, dapibus id fermentum quis, suscipit id erat.

Blog

Název článku 1

Název článku 2

Kontakt

Hradecká 1200
Hradecké Králové 503 27
Tel.: +420 666 505 606
Email: gc@gc.cz

Sledujte nás

Podklad pro zadání DIPLOMOVÉ práce studenta

PŘEDKLÁDÁ:	ADRESA	OSOBNÍ ČÍSLO
Andrlová Denisa	Hvozdnice 63, Hvozdnice	11414

TÉMA ČESKY:

Návrh, realizace a marketingová strategie webové aplikace

TÉMA ANGLICKY:

Design, implementation and marketing strategy of the web application

VEDOUCÍ PRÁCE:

Ing. Karel Mls, Ph.D. - KIT

ZÁSADY PRO VYPRACOVÁNÍ:

Úvod
Cíl a metoda výzkumu
Rešerše literatury
Funkční a non-funkční požadavky
Návrh aplikace
Marketingová strategie aplikace
Závěr a zhodnocení

Cílem diplomové práce je návrh webové aplikace, sloužící jako nástroj pro týmovou spolupráci a její následná marketingová strategie.

SEZNAM DOPORUČENÉ LITERATURY:

O'REILLY, Tim. What is Web 2.0: Design patterns and business models for the next generation of software. Communications & strategies, 2007, 1: 17.
CONSTANTINIDES, Efthymios; FOUNTAIN, Stefan J. Web 2.0: Conceptual foundations and marketing issues. Journal of direct, data and digital marketing practice, 2008, 9.3: 231-244.
FERNANDEZ, Adrian; INSFRAN, Emilio; ABRAHÃO, Silvia. Usability evaluation methods for the web: A systematic mapping study. Information and Software Technology, 2011, 53.8: 789-817.
FUTRELL, Robert T, Donald F SHAFER a Linda SHAFER. Quality software project management. Upper Saddle River, NJ: Prentice Hall, c2002, xxxvi, 1639 p. ISBN 0130912972.
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, 567 s. ISBN 978-80-251-1503-9.

Podpis studenta:

Datum:

Podpis vedoucího práce:

Datum: