

**UNIVERZITA KOMENSKÉHO  
V BRATISLAVE**

---

**Fakulta matematiky, fyziky a informatiky**

**Informačný systém pre športový klub**

**Bakalárska práca**

**Bratislava, 2013**

**Martin Kuchyňár**



UNIVERZITA KOMENSKÉHO V BRATISLAVE  
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

# Informačný systém pre športový klub

Bakalárska práca

**Študijný program:** Informatika

**Študijný odbor:** 2508 Informatika

**Kód práce:** a50a872f-9998-4594-a6ef-a4218436255d

**Katedra:** Katedra informatiky

**Vedúci práce:** doc. RNDr. Andrej Ferko, PhD.



Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky

---

## ZADANIE ZÁVEREČNEJ PRÁCE

**Meno a priezvisko študenta:** Martin Kuchyňár  
**Študijný program:** informatika (Jednoodborové štúdium, bakalársky I. st., denná forma)  
**Študijný odbor:** 9.2.1. informatika  
**Typ záverečnej práce:** bakalárska  
**Jazyk záverečnej práce:** slovenský

**Názov:** Informačný systém pre športový klub

**Cieľ:**  
1. Prehľad problematiky  
2. Špecifikácia funkčnosti  
3. Implementácia

**Literatúra:** Garrett, J.J. The Elements of User Experience: User-Centered Design for the Web and Beyond. New Riders 2010.

**Anotácia:** Cieľom práce je metodicky navrhnuť portál na organizáciu aktivít menšieho športového klubu (napr. badminton, stolný tenis). Funkčnosť vychádza z potrieb organizátorov (napr. tréningy, sústreďenia, tréningové plány). Databázové funkcie v prostredí MySQL s využitím CSS a PHP budú slúžiť na zdieľanie dát a štatistík o predchádzajúcich aktivitách.

**Kľúčové slová:** Informačný systém, vizualizácia

**Vedúci:** doc. RNDr. Andrej Ferko, PhD.  
**Katedra:** FMFI.KAGDM - Katedra algebry, geometrie a didaktiky matematiky  
**Vedúci katedry:** prof. RNDr. Pavol Zlatoš, PhD.

**Dátum zadania:** 11.10.2012

**Dátum schválenia:** 30.10.2012

doc. RNDr. Daniel Olejár, PhD.  
garant študijného programu

---

študent

---

vedúci práce

## **Pod'akovanie**

Týmto by som chcel poďakovať svojej rodine za podporu počas celej doby štúdia a vedúcemu bakalárskej práce Doc. RNDr. Andrejovi Ferkovi, PhD. za odborný dohľad, cenné pripomienky a dobré rady pri písaní bakalárskej práce.

# Abstract

---

*ENG version*

**Author:** Martin Kuchyňár

**Thesis title:** Information System for a Sport Club

**University:** Comenius University, Bratislava

**Faculty:** Faculty of Mathematics, Physics and Informatics

**Department:** Department of Computer Science

**Advisor:** doc. RNDr. Andrej Ferko, PhD.

**Thesis Length:** 37

Jun 2013, Bratislava

In this bachelor thesis we are intending methodically show how to create and test web portal for organizations like a small sport club (badminton, table tennis). Functionality is coming from user needs (training coordination, training plans). Database functions are in MySQL, generated by PHP and presented by CSS. The thesis is divided into 3 chapters deals with problems according to user experience and we offer a Slovak translation of Element's of User Experiences Diagram introduced by J.J. Garrett

**KEY WORDS:** Information system, User-Centered Design, User Experience (UX)

---

# Abstrakt

---

*SVK version*

**Autor:** Martin Kuchyňár

**Názov práce:** Informačný systém pre športový klub

**Škola:** Univerzita Komenského v Bratislave

**Fakulta:** Fakulta Matematiky, Fyziky a Informatiky

**Katedra:** Katedra informatiky

**Vedúci bakalárskej práce:** doc. RNDr. Andrej Ferko, PhD.

**Rozsah:** 37

Jún 2013, Bratislava

V bakalárskej práci sme navrhli ako metodicky vytvoriť a otestovať portál na organizáciu aktivít menšieho športového klubu (napr. badminton, stolný tenis). Funkčnosť vychádza z potrieb organizátorov (napr. tréningy, sústreďenia, tréningové plány). Databázové funkcie v prostredí MySQL s využitím CSS a PHP budú slúžiť na zdieľanie dát a štatistík o predchádzajúcich aktivitách. Práca je rozdelená do 3 kapitol. Práca sa zaoberá problematikou user experience a v jej rámci sme preložili do slovenčiny diagram Element's of User Experience od J. J. Garretta.

**KEÚČOVÉ SLOVÁ:** informačný systém, user-centered Design, user experience (UX)

---

# Zoznam použitých skratiek

---

UX	User experience
MVC	Model-view-controler
CMS	Content management system
DB	Databáza
GUI	Graphical user interface

## Zoznam tabuliek

---

Tabuľka 1: Elementy user experience

18

## Zoznam obrázkov

---

Obrázok 1: Schéma MVC .....	3
Obrázok 2: Pomenovania rovín s dualitou, grafika J.J Garrett, preklad Martin Kuchyňár .....	5
Obrázok 3: Pomenovania rovín, grafika J.J Garrett, preklad Martin Kuchyňár .....	5
Obrázok 4 Počiatočný koncepčný model .....	8
Obrázok 5 Koncepčný model rámcovej roviny .....	9
Obrázok 6 Globálna navigácia [5,120] .....	13
Obrázok 7 Lokálna navigácia [5,121] .....	13
Obrázok 8 Doplnujúca navigácia [5,121] .....	14
Obrázok 9 Kontextová navigácia [5,122] .....	14
Obrázok 10 Zdvorilostná navigácia [5,121] .....	14
Obrázok 11: Ilustrácie emócií [11] .....	20
Obrázok 12: Persóna pre hráčsku skupinu .....	22
Obrázok 13: Persóna pre trénerskú skupinu .....	22
Obrázok: 14 Persóna pre skupinu vedúceho klubu .....	23
Obrázok 15: Strategický konceptuálny model .....	23
Obrázok 16: Tabuľka pred úpravami .....	26
Obrázok 17: Db riešenie pre účasť .....	26
Obrázok 18: tabuľka po úpravách .....	26
Obrázok 19: DB riešenie pre zápasy .....	26
Obrázok 20 Kompletne DB riešenie .....	28
Obrázok 21 Bezvýznamný error template .....	29
Obrázok 22 Koncepčný model domácej stránky .....	29
Obrázok 23 Úrovne softwaru pre GUI [21, str. 247] .....	31
Obrázok 24 Wireframe pre domácu stránku .....	32
Obrázok 25 Wireframe po prihlásení užívateľa .....	33
Obrázok 26: Výsledný vzhľad stránky pred prihlásením .....	34
Obrázok 27: Výsledný vzhľad stránky po prihlásení .....	34

# Obsah

---

<b>ÚVOD</b> .....	<b>2</b>
<b>MODEL-VIEW-CONTROLER</b> .....	<b>3</b>
<b>USER EXPERIENCE</b> .....	<b>4</b>
<b>ELEMENTY USER EXPERIENCE</b> .....	<b>5</b>
STRATEGICKÁ (STRATEGY) ROVINA .....	6
RÁMCOVÁ (SCOPE) ROVINA .....	8
ŠTRUKTURÁLNA (STRUCTURE) ROVINA .....	10
KONŠTRUKČNÁ (SKELETON) ROVINA.....	12
POVRCHOVÁ (SURFACE) ROVINA .....	16
<b>TESTOVANIE POUŽITEĽNOSTI (USABILITY)</b> .....	<b>19</b>
<b>ŠPECIFIKÁCIA</b> .....	<b>21</b>
STRATEGICKÁ ROVINA .....	21
<i>Ciele</i> .....	21
<i>Potreby používateľa</i> .....	21
RÁMCOVÁ ROVINA .....	24
<i>Funkčná špecifikácia</i> .....	24
<i>Požiadavky na obsah</i> .....	25
<b>IMPLEMENTÁCIA</b> .....	<b>29</b>
ŠTRUKTURÁLNA ROVINA .....	29
<i>Dizajn interakcie</i> .....	29
<i>Informačná štruktúra</i> .....	30
KONŠTRUKČNÁ ROVINA .....	31
<i>Dizajn rozhrania</i> .....	31
<i>Dizajn navigácie</i> .....	31
<i>Informačný dizajn</i> .....	32
POVRCHOVÁ ROVINA .....	34
<b>ZÁVER</b> .....	<b>35</b>
<b>ZDROJE</b> .....	<b>36</b>
<b>PRÍLOHY</b> .....	<b>37</b>



# Úvod

---

Vzrastaním konkurencie na internete nastala situácia, kedy sa treba zamýšľať nad tvorbou web aplikácií inak ako pri dizajnovaní všedných vecí. Pri takomto prístupe treba nezohľadňovať iba funkcionálnu alebo estetickú stránku produktu. Existuje praktický rozdiel pri kúpe topánok a pri vyberaní si web aplikácií. Pri kúpe topánok zvažujeme okrem ceny ich účel a estetickú stránku. To isté sa deje aj pri výbere web aplikácie, avšak rozdiel nastáva, keď začneme dané produkty využívať. Kúpené topánky v tomto čase už nemožno vrátiť do obchodu s odôvodnením, že sa vám po 2 minútach vždy rozviažu šnúrky. U internetových aplikácií panuje úplne rozdielna situácia, ak sa mi nepracuje dobre s jednou aplikáciou, jednoducho začnem používať konkurenčnú. Takéto rozdiely často tvoria hranicu medzi úspechom a zlyhaním projektov, t.j. zanechať dobrý dojem z používania produktu, po anglicky user experience.

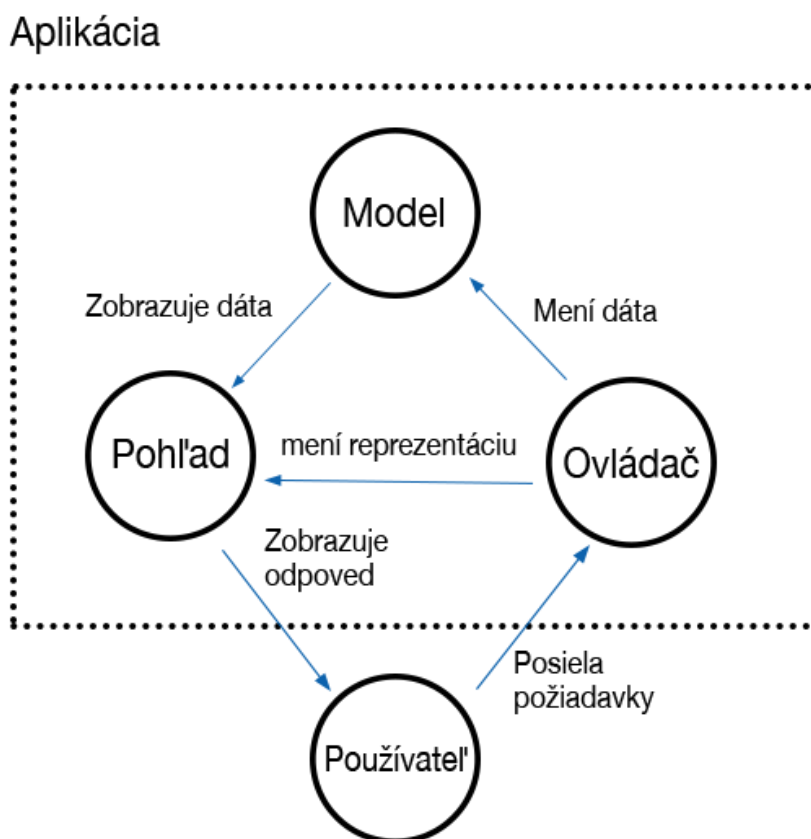
Táto bakalárska práca sa snaží poskytnúť komplexný pohľad na metodiku, pre vytvorenie internetovej aplikácie, pre správu komunikácie menšieho športového klubu. Dôraz pri výstavbe kladieme na zážitky užívateľov (UX) a ich pocity pri používaní aplikácie. Práca má vysvetliť a poskytnúť čitateľovi potrebné know-how pre tvorbu moderných, intuitívnych web produktov. Navyše poskytnúť modelový príklad takéhoto projektu.

Hlavný prínos práce sa nachádza v komplexnosti výkladu, ktorí uvádza čitateľa do teoretických znalostí v danej tematike, ich aplikáciu na reálny produkt a aj uvádza interné záležitosti, princípy fungovania v implementačnej časti.

Práca sa člení do 3 hlavných kapitol. V práca obsahuje anglické ekvivalenty kľúčových slov, uvedené v zátvorkách, ktoré majú čitateľovi pomôcť pri následnom hľadaní podrobnejších informácií alebo literatúry. Prvá kapitola čitateľa uvádza do problematiky UX. V kapitole sú postupne vysvetlené všetky podstatné kroky zohľadňované pri výstavbe ľubovoľnej web aplikácie (väčšina princíпов je aplikovateľná na ľubovoľný produkt). Druhá kapitola je mierená na zadefinovanie konkrétneho problému TJ. Zadanie smerovania web aplikácie, všetky podmienky zadané v prvej kapitole. Posledná kapitola sa zaoberá praktickou implementáciou systému na mieru pomocou programovacieho jazyka PHP.

# Model-view-controler

Pre návrh aplikácií dnes existuje množstvo štandardizovaných návrhových vzorov, ktoré popisujú, ako je aplikácia navrhnutá, aby fungovala. Preto pri začiatku návrhu je vhodné si prejsť tieto štandardné modely fungovania, pretože sa často stáva, že množstvo návrhových vzorov tvoríme, čo zbytočne zaberá čas. Neskôr zistíme, že nie sme až inovatívny ako sme zamýšľali, takýto prístup už existuje dlho. V oblasti internetových aplikácií v poslednej dobe najčastejšie používa návrhový vzor MVC, menej častejšie sa vyskytujú monolitické systémy. MVC separuje interaktivitu používateľa a vnútornú reprezentáciu údajov v našom systéme. Týmto postupom sa systém rozpadá na 3 samostatné časti, zložené z modelu, ovládača (controler), pohľadu (view). Model je časť aplikácie, často prezývaná aj dátová časť, ktorá reprezentuje informácie, používané v aplikácií. Pohľad je časť aplikácie, prezývaná prezentačná časť, ktorá predvádza dátovú časť do podoby vhodnej na prezentáciu a interakciu s používateľom. Ovládač je časť aplikácie, ktorá vykonáva zmeny na modeli a mení pohľad. Výhody takejto architektúry sa nachádzajú pri upravovaní výsledného projektu. Napríklad model môže mať viac pohľadov, a teda je jednoduché ich meniť alebo pridať nový pohľad. Túto architektúru ako prvý popísal Trygve Reenskaug [1] v roku 1979, používa najčastejšie pri vývoji webových aplikácií. Tento princíp sa vyskytuje napríklad v známych frameworkoch ako Zend Framework [2], Symfony [3] alebo CakePHP [4].



Obrázok 1: Schéma MVC

# User experience

User experience sa dá voľne do slovenčiny preložiť ako zážitok používateľa alebo používateľove skúsenosti (s našim produktom). Najčastejšie ide o každodenné pocity z používania produktov, či už ide o drobnosti v podobe oblých rohov, ale aj v frustrácii z nemožnosti dosiahnuť cieľ. Takéto pocity nedokážeme priamo riadiť, keďže matematici nám ešte neposkytli presný matematický model pre používateľa. Môžeme však ovplyvniť množstvo detailov, pomocou ktorých ovplyvníme skúsenosti našich používateľov žiadaným smerom, čo môže byť výhodné napríklad pri predaji produktov.

Lubovoľný produkt, dizajnovaný pre používanie ľuďmi, dokážeme riadiť na 3 rôznych úrovniach myslenia. Prvá úroveň reprezentuje schopnosť našich používateľov splniť úlohu, pre ktorú bol produkt navrhnutý. V prípade našej aplikácie: dokáže člen športového klubu nájsť výsledky svojich zápasov z tréningu z minulého mesiaca? Táto úroveň UX je nazývaná použiteľnosť (usability) a je presne definovaná pomocou vytýčenia úloh a ich úspešným alebo neúspešným riešením. Túto úroveň možno exaktne merať, štandardne pomocou času. Druhá úroveň sa zaoberá očakávanými pocitmi používateľov, spojenými s produktom. Dobrý príklad poskytuje bankový sektor. Keď používate internet banking systém, očakávate pocit bezpečnosti svojich úspor. Na rozdiel, pri kúpe hernej konzoly očakávate pocit zábavy pri jej používaní. Preto by každodenné veci aj IT produkty mali byť prispôsobené očakávaným pocitom. Tretia úroveň sa zaoberá sociálnym dopadom používania nášho produktu. Napríklad, ak si človek kupuje potraviny cez internetový obchod bude okolím vnímaný ako inovatívny. Takéto vnímania sú dôležité pre vnímanie produktov našej firmy v dobrom svetle. Pri aktuálne využívanom vývoji sa kladie dôraz hlavne na prvú úroveň avšak na ostatné sa pozabúda najčastejšie. Dôsledkom toho sa stáva, že vývojári sa zamýšľajú pri návrhu nad používateľmi a nie reálnymi ľuďmi [7].

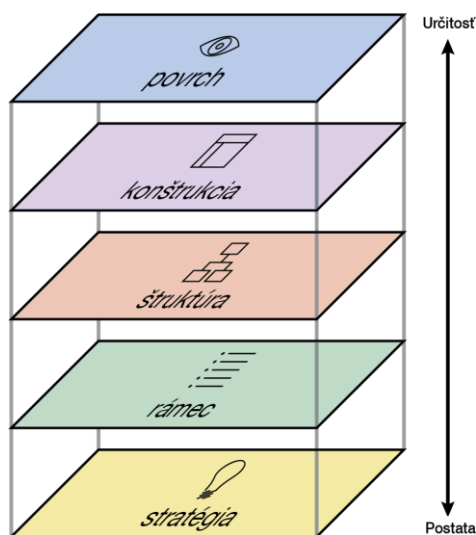
V reálnom svete používajú produkty prostí ľudia, nie modelový používateľia. Preto je dôležité vnímať ako produkt pracuje zvonka, nie vnútorné záležitosti fungovania, ale v čase konfrontácie s reálnym používateľom. Používateľom vnímaným ako človekom. Takýmto spôsobom sa tvoria zážitky používateľov. Vnútorné záležitosti sú doménou programátorov nie používateľov.

*„Takéto zážitky tvoria iba ľudia, používajúci produkt v reálnom svete. Používateľia kladú veľký dôraz na to, čo produkt dokáže (funkcionalita a jej implementácia) . User experience je iná, väčšinou ostáva prehliadnutá časť, a to, ako produkt pracuje, môže často znamenať rozdiel medzi úspechom alebo zlyhaním [5, str. 6].“*

Ďalej predstavíme metodiku, kde sa J. J. Garrett pokúša popísať návrh produktu od nápadu až po konečnú realizáciu ktorým sa priblížime k návrhu UX. Budeme postupovať po 5 rôznych úrovniach. Takáto postupnosť nám dovoľí postupne presnejšie usmerňovať UX až do požadovanej podoby. Rovnaké postupy sa vyskytujú aj v iných odvetviach, ako je vývoj aplikácií. Výbornú metaforu nám ponúkajú krajčíri. Ten si vždy pozve zákazníka na viacero skúšok, kde sa najprv zameriava na základné črty oblečenia, aby zistil akú má postavu. Na ďalších upresňuje aby aj doplnky pasovali na konkrétneho zákazníka. Takto sa zákazník podieľa na vývoji produktu, a preto pocity práve tohto používateľa budú presnejšie navrhnuté.

# Elementy user experience

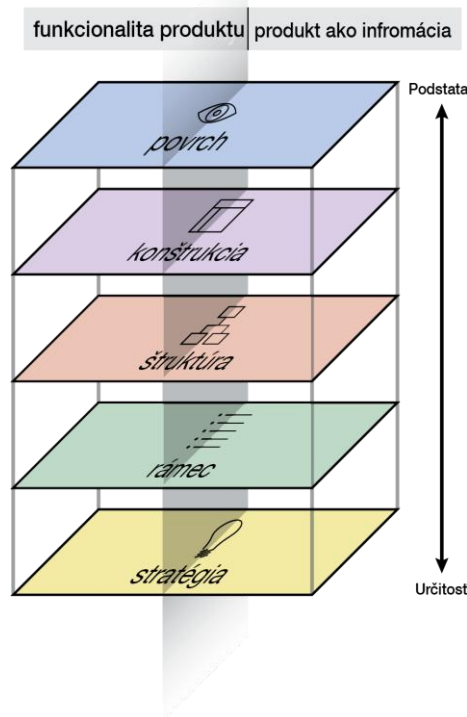
Ovplyvniť UX môže celý rad rozhodnutí pri návrhu produktu, niektoré nepodstatné a niektoré strategické, napríklad čo by mal priniesť produkt používateľovi. Takéto rozhodnutia podstúpené postupne počas návrhu môžu ovplyvniť všetky aspekty UX. Časti návrhu možno rozčleniť do 5 návrhových rovín [5]. Rovinu povrchu, konštrukcie, štruktúry, rozsahu a stratégie. Roviny sú zoradené chronologicky podľa úrovne abstrakcie. Na povrchovej vrstve vidíme produkt (v našom prípade web aplikáciu), ako kompozíciu textu a obrázkov, pomocou ktorých dokážeme porozumieť informácie, prípadne interagovať pomocou nich. Na



Obrázok 3: Pomenovania rovín, grafika J.J Garrett, preklad Martin Kuchyňár

konštrukčnej vrstve vidíme, akým spôsobom rozvrhneme produkt napríklad, kde na televízii je tlačidlo na vypnutie, aby bola efektívnosť používania tohto tlačidla najoptimálnejšia. Štruktúrna vrstva, obrazne povedané sprevádza používateľa po našom produkte. V našom prípade sú to úvahy, akým spôsobom sa používatelia dostanú k svojim informáciám. Rámová rovina definuje, aký rozsah bude mať funkcionálnosť produktu. Strategická je najviac abstraktná rovina, určuje vonkajšie a vnútorné ciele pre ktoré bol produkt navrhnutý. Mala by odpovedať napríklad na otázky typu: Čo chceme dosiahnuť pomocou nášho produktu? Týchto 5 rovín ponúka koncepciu ako uvažovať o problémoch v user experience a prostriedky ako vyriešiť tieto problémy. Na každej rovine sa zapodieваме našim produktom z inej úrovne abstrakcie,

napríklad na povrchovej vrstve riešime výlučne otázku vzhľadu produktu, na konštrukčnej zase rozloženie riadiacich prvkov. Tieto 2 roviny nie sú separované, keďže rozhodnutie rozloženia ovplyvňuje výsledný vzhľad produktu. Preto je každá rovina závislá od abstraktnejšej roviny, napríklad štruktúrna od rámcovej. Z týchto dôvodov budeme budovať náš produkt "od spodu" (bottom up), od najabstraktnejšej vrstvy. Zároveň, pre dôvod prepojenia vrstiev pri vývoji je potrebné neseparovať fázy vývoja na rozdielne fázy. Elementy rozdeľujeme ešte na drobnejšie časti, pretože web bol originálne myslený ako hypertextový informačný priestor, ale vývoj sofistikovanejších front/back-end technológií, ho zmenil na vzdialené softwarové rozhranie (remote software interface). Táto dvojité podoba vyžaduje, aby sme sa na webové aplikácie pozerali z dvoch rozdielnych pohľadov. Z pohľadu funkcionality nášho produktu, sa zapodieваме využitím technológií tak, aby vyhoveli používateľovi (interakcia, funkčná špecifikácia). Tieto 2 rôzne pohľady budeme neskôr spomínať ako princíp duality.



Obrázok 2: Pomenovania rovín s dualitou, grafika J.J Garrett, preklad Martin Kuchyňár



## Strategická (strategy) rovina

Strategická rovina je v podstate určená strategickými cieľmi našej stránky. Základom týchto cieľov je poznanie čo, chceme dosiahnuť pomocou nášho informačného systému. Príklad jedného z našich cieľov: Používateľ chce nájsť informácie (napríklad o predchádzajúcich tréningoch svojich zverencov), my chceme, aby svoj tréningový plán prispôbil ideálne pre našich hráčov.

Chýbajúce odpovede na dve strategické otázky sú častá príčina zlyhania množstva projektov (nie len z prostredia web stránok) nie je technológia, dokonca ani user experience nie. Najčastejšie chyby sa nachádzajú hlavne na tejto rovine v podobe nepremyslenia dvoch jednoduchých otázok. Čo chceme my, ako autori, dosiahnuť pomocou tohto produktu? A čo v skutočnosti chcú od nášho produktu naši zákazníci?

Odpoveďou na prvú otázku je stratégia smerovania stránky a presné pomenovanie cieľov projektu. Často táto stratégia existuje ako nevydiskutované porozumenie jedného pracovníka počas jeho práce na produkte. Avšak rozdielni ľudia majú rôzne názory, ako by mal cieľ projektu vyzerat'. Preto je potrebné mať od začiatku predstavu, čo presne chceme dosiahnuť pomocou vyvíjaného produktu. V tejto časti je taktiež dôležité stanoviť si ciele. Štandardnými cieľmi vo svete vývoja aplikácií je zarobiť peniaze alebo ušetriť pomocou danej aplikácie. Z týchto cieľov nie je jasné, ako to naša aplikácia docieli, preto je nie sú takto formulované ciele najvhodnejšie. Z iného pohľadu sa treba vyvarovať aj prílišne špecifických cieľov, pretože by sme mohli nedodržať stratégiu smerovania stránky. Preto je potrebné vhodne bilancovať medzi týmito dvoma extrémami, aby sme dosiahli úspech v UX. V prípade stanovených cieľov je potrebné vedieť tieto ciele „odmerať“. Pre takéto účely existuje mnoho nástrojov, ktoré nám prezradia, napríklad aký dlhý čas používateľovi zaberie, pokiaľ opustí domácu stránku (a teda za koľko pochopí, kde sa vlastne nachádza), alebo koľko stránok si používatelia prehliadajú priamo z vyhľadávača a na ktoré sa dostanú následne.

Odpoveďou na druhú otázku sú potreby používateľa. Je veľmi jednoduché dizajnovať pre nás samotných, avšak naša snaha je dizajnovať pre ostatných, ideálne pre všetkých čo použijú náš produkt. Aby používatelia mali radi, čo sme vytvorili, musíme vedieť, kto naši používatelia sú, a čo od nás potrebujú alebo očakávajú. Zisťovanie potrieb používateľov môže byť veľmi komplikované, keďže používatelia môžu byť výrazne odlišný, preto je žiadané rozčleniť užívateľov do viacerých cieľových menších skupín. Používateľské prieskumy rozdeľujú používateľov najčastejšie podľa demografických faktorov ako sú: pohlavie, vek, úroveň vzdelania, priemerný príjem, rodinný stav a tak ďalej. Takto vytvorené profily dokážu byť všeobecné (ľudia vo veku 18-47 rokov), ale aj konkrétne (žena vo veku 19-22 rokov, slobodná, zarábajúca 2500 e/mesačne). Demografické fakty nie sú jediný spôsob ako zdefinovať cieľových používateľov. Ďalší spôsob ako sa pozerat' na našich používateľov ponúka psychologická demografia (z anglického Psychographic [5, str. 44]), tá ukazuje, aké majú naši používatelia postoje k našej stránke. psychologická demografia silno koreluje s demografickým faktorom, keďže ľudia podobne starí, zarábajúci, navštevujúci podobné miesta mávajú podobné postoje. Nie je to však pravidlom, len zvyklosťou, môžeme to využiť ako doplnkové informácie k demografickým faktorom. Ďalším dôležitým poznatkom o našich používateľoch, pri navrhovaní web stránky sú údaje: Ako často naši používatelia surfujú na internete? Sú technológie súčasťou ich života? Ako často pracujú s technológiami, ktoré sú obsiahnuté v produkte? Sú nadšenci nových technológií? V závislosti od pochopenia našich používateľov a oboznámenia sa úrovňou ich komfortu pri pracovaní s technológiami, musíme

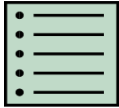
porozumieť, ako veľmi používatelia vyznajú v danej téme. Napríklad technofóbia, technologicki zdatní ľudia, používajú webové aplikácie úplne rozdielnym spôsobom ako sviatoční používatelia internetu. Ďalší podstatný dôvod pre tvorenie cieľových používateľských skupín je, že tieto skupiny nemajú vždy len rozdielne potreby, ale dokonca úplne opačné potreby. Samozrejme, nemôžeme sa vyhovieť všetkým takto vytvoreným skupinám s jediným riešením. Naša možnosť je zamerať sa na jednu z takto vytvorených skupín, alebo ponúknuť dva rozdielne prístupy pre používateľov na vykonanie identickej úlohy. Ktorúkoľvek z týchto možností si vyberieme, toto strategické rozhodnutie bude mať dôsledky pre každé ďalšie rozhodovanie ohľadne user experience. Každú cieľovú používateľskú skupinu budeme ilustrovať pomocou osoby (pre každú skupinu samostatne).

Keď už vieme čo presne tvoríme, je vhodné (hlavne pre ďalší postup) začať tvoriť koncepčný model. Koncepčný model je zjednodušene povedané očakávanie používateľa ako správne použiť daný produkt, ak náš zákazník má nesprávnu predstavu o tom ako v skutočnosti produkt pracuje, je pravdepodobné, že produkt použije spôsobom na ktorý nebol navrhnutý. Inak ovláda produkt bezmyšlienkovite, mechanicky, naslepo[15, str. 39]. Takéto zaobchádzanie často končí frustráciou, pretože používateľ len ťažko dosiahne svoj cieľ.

Doteraz však nemáme dostatok vykonanej práce aby sme vytvorili kompletný koncepčný model, je však dôležité aby sme sa nad ním začali zamýšľať na začiatku návrhu produktu. Pretože nám to ušetrí spätnú prácu na každej ďalšej rovine. Aktuálne disponujeme informáciami čo chceme dosiahnuť a pre koho chceme dané služby ponúkať prioritne. Preto ako výsledok tejto roviny si graficky znázorníme čo vieme pomocou diagramu. Nazveme ho strategicky koncepčný diagram. Takáto prezentácia výsledkov výrazne demonštruje výsledky strategickej roviny a teda pomáha predchádzať nedorozumeniam v ďalšom ponímaní smerovania projektu, čo môže mať fatálne následky v ďalšom návrhu.



Obrázok 4: Počiatočný koncepčný model



## Rámcová (scope) rovina

Vo fáze, keď vieme, akým smerom chceme náš produkt smerovať my a čo očakávajú používatelia, začneme sa zaujímať, akým spôsobom vyhovieť užívateľským potrebám a smerovaniu našej stránky. Strategická rovina sa mení na rámcovú v momente, keď začneme tvoriť konkrétnu funkcionality, ktorá bude prístupná používateľovi. Rámcová rovina určuje rozsah využiteľnosti, alebo funkcionality ktorá bude prístupná na web stránke (všeobecne povedané v produkte). Všetka funkcionality a všetky funkcie zahrnuté na web stránke sú otázkou rámcovej roviny.

Rovinu rozdeľujeme na časť funkcionality a stranu obsahu. Z duálneho pohľadu na web je táto rovina rozdelená na funkčnú špecifikáciu a požiadavky na obsah.

Predtým, ako budeme mať zoznam funkcií, je však nutné zamyslieť sa, čo chceme, aby náš produkt ponúkal, ale aj nad otázkou toho čo nechceme, aby ponúkal. Istá časť funkcionality môže pôsobiť, ako výborný nápad avšak nemusí sa zhodovať so smerovaním stránky, alebo môže tento smer odkláňať. Rozumieť, čo netvoríme značí tiež rozumieť tomu, čo tvoríme aktuálne, pretože ak si zvolíte konkrétne požiadavky pre celý plán návrhu, takéto „nové nápady“, už nebude možné pridať v novších verziách, na druhú stranu ak by sme takéto nápady, zaradovali do plánu veľmi často môže sa stať, že z dôvodu prebytku takejto práce nezostane čas na dosiahnutie hlavných cieľov aplikácie.

Otázka, čo vlastne budujeme, je vcelku jasná, avšak v tímovom pracovnom prostredí tomu nie je úplne tak. Manažér predvedie otázku a odpovede na úrovni strategickú roviny vedúcemu pracovníkovi tímu, ten znova interpretuje svojmu tímu. Po tomto všetkom sa môže nastať situácia, že od očakávania manažmentu, ktorý navrhol smerovanie produktu, vznikne niečo úplne iné. Preto je dôležité definovať požiadavky úplne presne, to nám umožní rozdeliť záväzky na prácu oveľa efektívnejšie.



Obrázok 4 Počiatočný koncepčný model

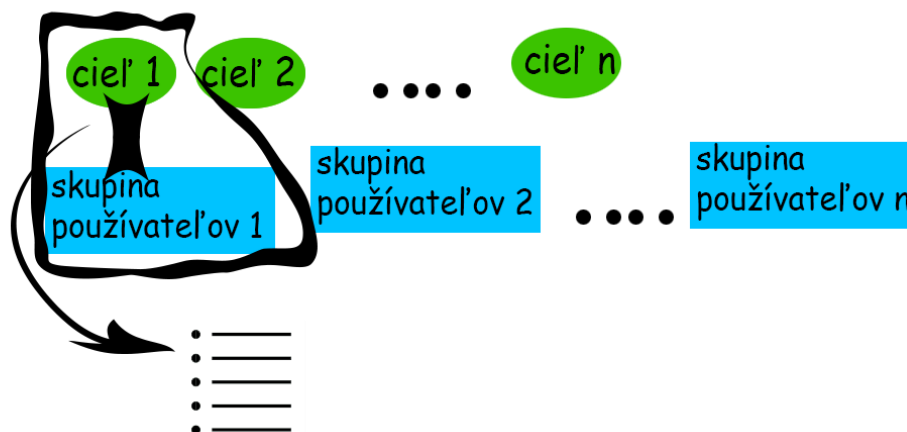
V tejto časti, nám pomôže strategický koncepčný diagram. Obsahuje všetko potrebné na tvorbu presnej špecifikácie, ktorá sa neodkláňa do smerovania stránky. Priamo vytyčuje nám štart a finish projektu. Finish je reprezentovaný cieľmi ktoré sme si vytýčili v strategickú rovina. Na štarte máme naše skupiny užívateľov reprezentované personami. Na tejto rovine je našou úlohou poskytnúť možnosť aby užívateľ mohol splniť všetky ciele. Inak povedané poskytnúť vhodnú funkcionality. Musíme vziať všetky možnosti dvojíc cieľu a skupiny používateľov. Pre takúto dvojicu vieme ľahko navrhnuť funkcie pre umožnenie dosiahnutia cieľa z pohľadu istej skupiny používateľov.

Táto časť je vyššie spomínaná ako funkcionality čo chceme tvoriť, aby sme ale zistili čo nechceme tvoriť je potrebné tento zoznam zoradiť podľa priority. Takýmto postupom vytvoríme prioritnú frontu pre každú dvojicu cieľu a skupiny používateľov. Funkcie na



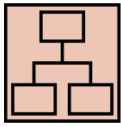
s najväčšou prioritou, budeme plniť ako prvé, ostatné zostanú a budú sa plniť podľa zostávajúceho času. Takýmto spôsobom sme poskytli jednej skupine používateľov prístup (most) k ich cieľovým požiadavkám. Následne musíme zvoliť inú dvojicu cieľu a skupiny používateľov a vytvoriť ďalšiu prioritnú frontu, pokiaľ nevyčerpáme všetky možnosti. Takýmto spôsobom vytvorená špecifikácia dokáže poskytnúť všetkým skupinám používateľov všetky ciele vyplývajúce zo strategickej roviny.

Požiadavky na obsah nie sú iba texty, ktoré budeme vidieť na stránke. V tejto rovine ide hlavne o systém, ktorý bude prezentovať. Väčšina webových aplikácií používa systém na správu obsahu (content management systém) [9]. CMS sa vyskytujú v rôznych škálach od komplexných systémov až po voľne dostupné systémy. Ich podstatou je možnosť meniť obsah stránok aj bez znalostí technológie. Okrem toho potrebujeme zabezpečiť aby naše už naplánované funkcie mali dostatočnú technickú podporu. Tento problém budeme znova vizualizovať. Vezmeme už vytvorenú prioritnú frontu a postupne zvrchu budeme plniť nároky pre fungovanie jednotlivej funkcionality. Napríklad ak chceme používať CMS musí mať podporu nejakej formy databázového systému pre ukladanie dát. Takýmto spôsobom musíme podporiť každú funkciu ktorá pripadá do úvahy aj v ďalekej budúcnosti, pretože by mohli nastať 2 nemilé záležitosti. Prvá nepríjemná je zmena už naprogramovaných funkcií kvôli pridaniu novej funkcie, ďalej môže nastať, že istú funkciu nebude vôbec možné implementovať, lebo to zaberie viac času ako môžeme zaplatiť, čo je problém. Množstvo funkcií sa dá podporovať rôznymi technológiami, preto musíme zobrať do úvahy, všetku (aj takú ktorá sa možno nebude realizovať) funkcionality. Takto vyberieme tú technológiu, ktorá je najideálnejšia pre náš projekt.



Obrázok 5 Konceptný model rámcovej roviny





## Štruktúrálna (structure) rovina

ak, sme si vytýčili ciele, stratégie a zoradili ich podľa priority, teraz máme jasný obraz toho, ako by mal náš výsledný produkt vyzerat' a aké požiadavky musí spĺnat'. Na žiadnej z rovín sme však nepopísali, akým spôsobom budú pracovať v jednom celku. Na štruktúrálnej rovine budeme budovať koncepčnú štruktúru nášho produktu (v našom prípade web aplikácie), práve táto rovina 5 vrstvovom modeli tvorí hranicu medzi abstraktným plánovaním a konkrétnejšími úlohami, spojenými s vývojom user experience. Z pohľadu funkcionality produktu do tejto roviny patrí dizajn interakcie, ktorý je podľa tradičného vývoja softvéru zaradený iba ako podmnožina práce v oblasti dizajnu rozhrania. Avšak v modernom vývoji sa dizajn interakcie separuje od dizajnu rozhrania a vytvára novú, samostatnú oblasť práce. Z druhého pohľadu, pohľadu informácií, rieši otázku štruktúrovania v user experience informačná architektúra. Dizajn interakcie aj informačná štruktúra majú rovnakú myšlienku a to definovať súvislosti, zákonitosti v akých budú prezentované používateľovi. Dizajn interakcie sa však sústreďuje na oblasť úspešného plnenia a ukončenia úloh, ktoré chce užívateľ vykonať, na rozdiel od informačnej štruktúry, ktorá sa snaží ponúknuť informácie na správnom mieste, napríklad tam, kde ich bude používateľ očakávať.

Dizajn interakcie sa sústreďuje na opis správania sa používateľa, a priamo definuje, ako bude produkt reagovať na akcie. Vždy, keď niekto používa náš produkt, by mala vznikať súhra medzi akciami používateľa a produktu, efektívna postupnosť akcií používateľa a následných reakcií produktu. Pre tvorenie takejto bezproblémovej interakcie je dôležitý používateľský prieskum, čo nám dáva len ďalší dôvod stavať tieto roviny odspodu nahor, a teda postupovať od abstraktnejších problémov po konkrétne. Problémom tvorenia takejto interakcie často bývajú samotní programátori, ktorí viac uvažujú nad technickou efektívnosťou riešenia ako nad prívetivosťou k vzorke cieľových používateľov. Riešenia ktoré najviac vyhovujú používanej technológii, večneou nie sú najpríjemnejšie pre používateľov. Používateľské pocity z interaktívnych komponentov, ktoré tvoríme na tejto rovine, sú známe ako konceptuálne modely. Napríklad typickým príkladom konceptuálneho modelu je nákupný košík, každý keď pridá do košíka produkt zpozoruje oznámenie o pridání produktu do košíka (podobne pre odobratie z košíka), a systém musí ponúkať funkcionality pre takéto akcie. Takéto modely sú často výsledkom ustálených zvykov, väčšina komerčných stránok sa chce držať takto „ustálených“ pravidiel.

Informačná architektúra má základy v samotnej ľudskej komunikácii. Ak potrebujeme preniesť informácie od jedného človeka k druhému, musíme zvážiť, akým spôsobom ich budeme interpretovať tak, aby im ostatní porozumeli správne. Inak povedané informačná štruktúra je zameraná na proces, akým spôsobom ľudia spoznávajú informácie. Snaha je očividná, a to prezentovať informácie tak, aby boli informácie porozumené správne. Najdôležitejšie sa to ukazuje pri stránkach ktorých účelom je práve sprostredkovať informácie, pretože užívatelia môžu ostať frustrovaný z neporozumenia. Základnou funkciou informačnej štruktúry je navrhnuť organizačnú a navigačnú schému, ktorá umožní používateľovi prechádzať obsahom stránky efektívne. Existujú 2 hlavné prístupy a to prístup zhora-nadol a zdola-nahor.

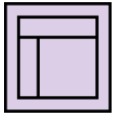
Zhora-nadol je forma prístupu k tvorbe informačnej architektúry vychádzajúc priamo zo správneho porozumenia strategickej roviny, inak povedané poznania presného smerovania stratégie a potrieb používateľa. Informačnú architektúru konštruovať od najširšieho okruhu

kategórií, ktoré postupne rozdeľuje na menšie logické podsekcie. Takáto hierarchia predstavuje šablónu, podľa ktorej bude reprezentovaný samotný obsah.

Prístup zdola-nahor taktiež tvorí informačnú architektúru pomocou vytvárania sekcií a príslušných podsekcii, ale nie je priamo vychádzajúci zo zadefinovania strategickej roviny. Tento prístup je založený na analýze rámcovej roviny, a teda sa odvíja od analýzy obsahu a funkčnej špecifikácie. Začína s materiálmi, ktoré už existujú a zlučuje ich postupne do nízko úrovňových sekcií, a tie ďalej spájame do vyššie úrovňových sekcií, pokiaľ neposkladá štruktúru reprezentujúcu ciele, zadefinované v strategickej rovine.

Žiadny prístup nie je vhodnejší ako druhý. Pri prístupe zhora-nadol sa často stáva, že používateľ prehliadne časť dôležitých informácií, v prístupe zdola-nahor je problém vo flexibilitate. Obsah web stránok sa neustále mení a takýto prístup, aj pri malej zmene, je potrebné zmeniť celú „starú štruktúru“, čo môže byť časovo veľmi náročné.

V tejto rovine máme všetko potrebné nato aby sme vytvorili koncepčný model. Preto ako výstup tejto roviny budem celkový koncepčný diagram spolu výberom prístupu k informačnej architektúre (Information Architecture Diagram).



## Konštrukčná (skeleton) rovina

Pod povrchovou rovinou sa nachádza kostra našej stránky, na tejto vrstve budeme definovať základné problémy typu ako funkcionality zadefinovaná v predchádzajúcej vrstve bude vyzeráť vo výslednej podobe.

Z pohľadu funkcionality produktu na tejto rovine definujeme pomocou dizajnu rozhrania, a teda pomocou rozostavenia kontrolných prvkov, formulárov, blokov informácií s textom, fotiek, tlačidiel. Takáto konštrukcia je dizajnovaná pre dosiahnutie ideálneho rozostavenia elementov pre maximálnu efektívnosť. Napríklad logo na komerčných web stránkach, je zobrazované vo vrchnej časti stránky, každej stránky, aby bolo ľahko zapamätateľné a charakterizujúce spoločnosť. Z druhého pohľadu, pohľadu informácií, definujeme konštrukčnú vrstvu pomocou navigačného dizajnu. Navigačný dizajn je špecializovaná forma rozhrania prispôbena pre prezentovania polohy a pohybu cez informačnú štruktúru definovanú v predchádzajúcich vrstvách. Nakoniec, z oboch stránok definujeme štruktúrnú rovinu pomocou informačného dizajnu, čo značí pochopiteľnú reprezentáciu oboch pohľadov. Tieto 3 časti sú prepojené do veľkej miery napríklad, správny návrh navigácie nepomôže, ak bol nepresne navrhnutý informačný dizajn.

Dizajn rozhrania definuje akým spôsobom budeme získavať spätné väzby (vstupy) od používateľov. Pre konkrétnu úlohu budeme voliť rozostavenia na ploche tak, aby bola jasne rozoznateľná ich funkčnosť, a formu vyžadovaného vstupu, aby poskytovala jednoduchú manipuláciu s ním. Pri riešení jednej úlohy treba často prejsť cez viaceré odkazy (podstránky) vyžadujúce rôzne vstupy a výstupy kontrolované používateľom. Problém výberu funkcie spustenej určitou interakciou je záležitosťou dizajnu interakcie a tá je súčasťou štruktúrnej roviny, ale spôsob ako je táto funkcionality realizovaná je oblasťou dizajnu rozhrania. Pre úspech dizajnu rozhrania je dôležité rozčleniť elementy (oznamy, veci) a informácie na tie významné, vzhľadom na ciele určené v strategickú rovine. Významné informácie musia byť viac pozorovateľné, ako tie označené ako menej dôležité, na druhú stranu elementy ktoré sú označené ako nepodstatné by mali byť v pozadí alebo úplne vynechané. Ďalej je potrebné zvoliť aké elementy budú použité. Tu sa musíme spoliehať na štandardné ovládací prvky, nové sa totiž zavádzajú avšak veľmi pomalým tempom. Uvedieme príklady štandardných ovládacích prvkov.

1. Checkbox: Dovoľuje vybrať práve jednu možnosť zo skupiny
2. Radio button: Dovoľuje vybrať ľubovoľný počet možností zo skupiny
3. Textové pole
4. Rozbalovací zoznam: rovnaká funkcionality ako radio button, ale umožňuje efektívnejšie využívanie priestoru
5. List boxes: Obdobu checkboxu, vďaka posuvníku je lepšie použiteľný na veľký počet prvkov.
6. Tlačidlo: vykonávajú rôznorodé funkcie, typicky ide o potvrdenie alebo odoslanie informácií

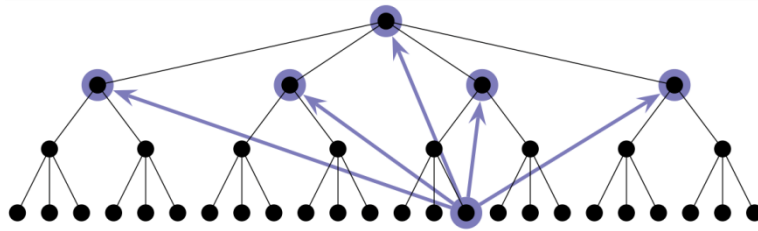
Všetky tieto elementy ponúkajú veľmi podobnú funkcionality, avšak vyberanie medzi nimi vždy vyžaduje kompromisy. Pre user experience dizajn je dôležitý rozdiel či používateľ bude vidieť všetky možnosti, ako to je v prípade použitia elementu checkbox, alebo budú niektoré skryté a zostane miesto na viacej dôležité informácie, ako v prípade list boxes.

Ďalšou súčasťou je dizajn navigácie. Navigácia je funkcionality nachádzajúca sa vo všetkých webových aplikáciách. Navigácia má 3 hlavné ciele:

1. Poskytnúť spôsob pohybu medzi jednotlivými stránkami. Štandardne je nepraktické aby boli v navigácii všetky stránky. Navigácia by mala poskytnúť pohyb tak, ako ho očakáva reálny používateľ, zároveň všetky linky musia byť funkčné.
2. Dizajn navigácie musí znázorniť aj rozdiel medzi elementmi v nej, nestačí poskytnúť vhodnú množinu linkou. Sú niektoré stránky dôležitejšie ako iné?
3. Dizajn navigácie sa musí prispôbovať, aktuálnej pozícii používateľa, aby bolo jasné kde sa práve nachádza.

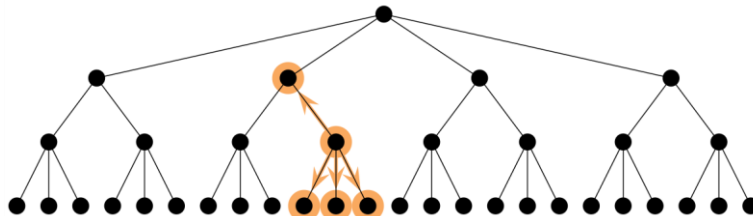
Väčšina súčasných webových aplikácií ponúka dizajn navigácie zložený z viacerých elementov postavených podľa zopár vzorov:

1. Najčastejšie sa vyskytuje globálna navigácia, ktorá ponúka prístup do hlavných častí celej aplikácie. Takáto navigácia sa nemusí nachádzať na každej stránke, avšak často to tak nebýva.



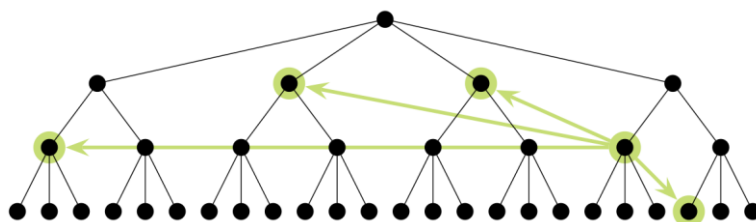
Obrázok 6 Globálna navigácia [5,120]

2. Lokálna navigácia poskytuje podrobnejšie informácie ako globálna navigácia, pre príklad ak by sme mali striktnu hierarchickú informačnú štruktúru, tak lokálna navigácia by, obsahovala elementy s odkazom na nadradenú, susednú sekciu a pod sekcie toho, kde sa užívateľ aktuálne nachádza.



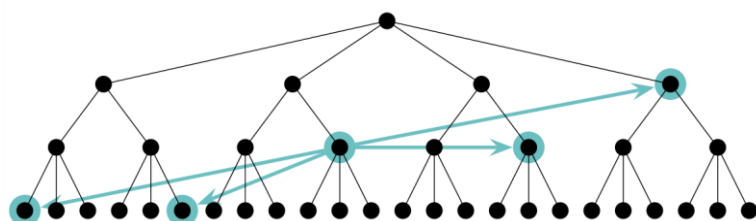
Obrázok 7 Lokálna navigácia [5,121]

3. Doplnujúca (Supplementary) navigácia obsahuje skratky ktoré možno priamo podľa hierarchie nie sú prepojené, ale je predpoklad, že užívateľ by mohol mať záujem o takéto prepojenie. Najlepší príklad nachádzame v internetových, obchodoch kde ak si prezeráte ponuku produktu, zvyčajne na tej istej stránke nájdete aj skratky s ponukou príslušenstva pre daný produkt.



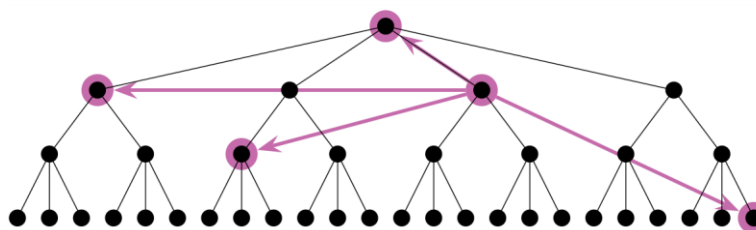
Obrázok 8 Doplňujúca navigácia [5,121]

4. Kontextová navigácia je vsadená priamo do obsahu, často je vizuálne vyznačená iba hypertextovým prepojením priamo v texte elementu. Používa sa za kvôli objasneniu, niekedy je vhodné vysvetliť isté pojmy v texte, aby používateľ mohol plnohodnotne porozumieť textu.



Obrázok 9 Kontextová navigácia [5,122]

5. Zdvorilostná (Courtesy) navigácia umožňuje prístup k informáciám, ktoré používateľ primárne nepotrebuje ale je slušné ich poskytnúť. Ide napríklad o informácie ako sú otváracie hodiny, túto informáciu si užívateľ vie sám vyhľadať pomocou inej časti navigácie ale je vhodné túto informáciu zobrazíť znova. Pri komerčných web aplikáciách sa v tomto type navigácie nachádzajú elementy typu kontaktného formuláru alebo podmienok používania.



Obrázok 10 Zdvorilostná navigácia [5,121]

V prípade že zlyhajú klasické spôsoby navigácie, užívatelia sa obracajú na staršie nástroje navigácie. Časté používanie takýchto nástrojov však značí chybu v návrhu dizajnu navigácie, reprezentujú však záchrannú brzdu pre stratených užívateľov. Najčastejšia forma je site mapa, ktorá popisuje hierarchickú štruktúru s linkami na hlavné časti obsahu zobrazenej kompaktne na jednej stránke bez iného informačného charakteru.

Spoločnú oblasť konštrukčnej roviny tvorí informačný dizajn. Obrazne povedané je to lepidlo spájajúce predchádzajúce dve časti. Základnou myšlienkou je prezentácia zrozumiteľnou formou. Prvá funkcia informačného dizajnu je wayfinding [5, str. 127], čo značí pomôcť používateľovi pochopiť kde sa nachádza a kam sa môže dostať. Takéto znaky môžeme ľahko nájsť aj v bežnom živote, napríklad rôzne farby na dopravných značkách už prezrádzajú ich význam. Tieto znaky je vhodné zaviesť aj pri tvorbe dizajnu webových aplikácií, aby pomohli

naším používateľom nadobudnúť jasný pocit (experience) vedomia, kde sa nachádzajú, aké majú možnosti kam sa dostať na našej web aplikácii pre dosiahnutie ich cieľa. Wayfinding znaky nevykonávajú navigačnú funkciu, ich funkcia je čisto informačná. Reálna forma takýchto znakov sa objavuje najčastejšie ako ustálené ikonky pre email alebo menej častejšie ako frázu „teraz sa nachádzate na:“. Ďalšou zjednocujúcou funkciou informačného dizajnu je rozloženie stránky (Wireframe [5, str. 128]), to je miesto kde je potrebné zapracovať všetko, čo sme si v tejto vrstve doposiaľ navrhli, a teda všetky druhy dizajnu navigácie, všetky elementy potrebné pre splnenie funkcionality atď. Skĺbenie všetkých návrhov do jedného návrhu stránky sa nazýva Wireframe. Pre menej komplexné projekty vystačíme s jedným takýmto rozdelením, naopak pre komplexné projekty je treba navrhnuť viacej rozložení. Wireframe je prvý krok k návrhu vizuálneho dizajnu.



## Povrchová (surface) rovina

Na najkonkrétnejšej povrchovej rovine budeme zvečňovať, čo používateľ vníma na našej stránke ako prvé, vizuálny dizajn. V tejto vrstve pozorujeme spojenie 2 strán produktu, tak ako zo stránky obsahu aj zo stránky informácií so zohľadnením estetiky. Takýto dizajn musí spĺňať všetky ciele rovín, ktoré sú opísané v abstraktnejších 4 vrstvách. Pre príklad v predchádzajúcej rovine sme sa sústredili na rozpoloženie elementov na stránke, ich interakciu a spôsobom ako správne reprodukovať informácie porozumiteľne, aby ich používateľ pochopil v čo najlepšom spôsobe, akým sme to určili na abstraktnejších rovinách. Keďže sme sa posunuli ku konkrétnejšej rovine, budeme riešiť otázku ako najvhodnejšie od prezentovať takto zoradené informácie. Všetky user experiences spojené s touto sú spracovávané pomocou všetkých 5 zmyslov. Čuch a vôňa býva dôležitou zložkou napríklad predaja produktov v potravinárstve, avšak v UX nevyužívajú. Ďalší zmyslom je hmat, v tejto vrstve avšak neriešime otázky napríklad, koľko by mal náš produkt mať tlačidiel, to je práca v oblasti dizajnu interakcie, riešime napríklad ich tvar, budú hladké na povrch alebo drsné aby boli takýmto spôsobom zvýraznené? Sluch a zrak sú najdôležitejšie vnemy týkajúce sa user experience. Zvuk je používaný v mnohých smeroch, napríklad ako druh upozornenia pri odosielaní alebo prijímaní správ. Zvuk však nie je vhodný iba na upozornenia, dá sa pomocou neho personifikovať produkt, tak ako je to štandardné pri mobilných telefónoch (nastavenie zvonenia, prijímania správ, haptic feedback). Nakoniec ostáva zrak, ktorí je z hľadiska user experiences najdôležitejší. Spočiatku si všetci myslia, že to je iba oblasť estetiky, alebo ako „cool“ náš produkt vyzerá a to je všetko o čo sa snažíme na tejto rovine. O čo však vo skutočnosti snažíme je aby náš dizajn čo najefektívnejšie plnil ciele vytýčené vo všetkých abstraktnejších rovinách, to znamená, že sa zameriavame na to ako dobre je náš produkt funguje. Pre príklad vezmeme príklad novej firmy tvoriacej si meno ako výrobca najvoňavejšieho chleba v meste, v takomto prípade použitie chladných farieb nebude správna voľba, preto nie vždy budeme riadiť iba estetickým hľadiskom ale aj tým aké ciele máme zadefinované skôr. Najjednoduchším spôsobom ako hodnotiť výsledný dizajn je sledovať používateľa, konkrétne jeho oči. Takto môžeme ľahko zistiť ktoré elementy získavajú ako prvé pozornosť používateľa a na druhú stranu ktoré elementy si ani nevšimne. V súčasnosti existuje množstvo profesionálneho vybavenia ktoré presne určuje kam sa používateľ pozerá, aký čas tam strávi [10], to však je veľmi finančne náročné. Ak iba doladíme výsledný dizajn je často vhodné previesť testovanie užívateľov pomocou iba toho, že ich požiadame, aby rozmýšľali nahlas [11] zadáme im štandardné úlohy ktorým je prispôbený dizajn, takýto prístup nedosahuje presnosť sledovania očí, avšak môžeme za veľmi prijateľnú cenu zistiť najťažšie problémy a pokúsiť sa ich odstrániť. Pri takomto testovaní si je potrebné dobre si všimnúť reakcie používateľa. Viac si o nízko rozpočtovom testovaní povieme v kapitole o použiteľnosti. Existujú dva hlavné faktori ktoré by sme mali všimnúť pri študovaní pohybu očí:

1. Ak používatelia vravia, že sa im dizajn zdá preplnený, v skutočnosti sú ich oči zaneprázdnené veľkým množstvom elementov ktoré sú zvýraznené a nevšimajú si celkový dizajn stránky.
2. Existujú možnosti ktoré môžu pomáhať plniť stanovené ciele, pomocou rôznych maličkostí ktoré môžu užívateľovi pomôcť k naplneniu. Napríklad v konštrukčnej vrstve sme definovali pomocou wayfinding bodov. Dôležitejšie ako existencia týchto bodov je to, aby tieto body nespôsobili zmätok užívateľov.

Pohyb očí je výsledok komplexných inštinktov ľudského vnímania, teda nie je náhodný. Našťastie pre dizajnérov, sme aspoň sčasti schopný rôznymi vizuálnymi technikami upriamovať pozornosť očí užívateľov. Prvou takouto technikou je práca s kontrastom. Dizajn

bez kontrastu pripomína masu elementov a oči každého používateľa by bezcieľne hľadali cieľové informácie. Kontrast teda pomáha používateľovi identifikovať vzťahy v informačnej štruktúre použitej na stránke. Dizajnové rozdielne elementy, práve tie kontrastujúce, upriamujú pozornosť používateľa, táto skutočnosť nám pomáha efektívnejšie prezentovať informácie, ktoré používateľ nesmie preskočiť. Tradičnou kontrastnou informáciou vo webových aplikáciách sú chybové upozornenia (hlášky), odlišujú sa farbou textu (napr. červenou) alebo použitím väčšieho fontu. Aby táto stratégia bola úspešná, rozdiel (použitý kontrast) musí byť dostatočne význačný, aby každý používateľ tento rozdiel zbadal a upriamil pozornosť. Ak by bolo ošetrenie dvoch chybových upozornení podobné avšak od seba dostatočne rozdielne, používateľa to zmätie. Začne sa zamýšľať, prečo boli tieto elementy rozdielne, prípadne začne byť frustrovaný z nepochopenia. Preto je potrebné pre rovnakú skupinu informácií jednotný dizajn. Pri tejto technike je dôležité mať vybranú paletu používaných farieb. Farby sú dôležitou súčasťou pri tvorení firemnej identity, väčšina farieb sa odvíja od farby produktov prezentovaných na stránke, napríklad firmu ponúkajúcu kolový nápoj budú farby palety biela (ako bublinky v nápoji) a hnedá (samotná farba nápoja). Tieto základné farby teraz rozlíšime či sú elementy v popredí, teda s vysokým kontrastom, alebo naopak v pozadí. Vo väčšine prípadov, jasnejšie a výraznejšie farby sú používané v popredných elementoch, ktoré majú priťahovať pozornosť. Viac tlmené (šedé)farby bývajú používané v pozadí. Ďalšou technikou, prebratou z tlačiarenského priemyslu, je mriežkové rozloženie (grid-based layout [5, str. 141]). Vytvorí základnú mriežku, kde následne umiestňuje elementy rôznej veľkosti, ale vždy dodržiava základnú mriežku. Hlavná nevýhoda nastáva pri rôznych rozlíšeníach mriežky (obrazovky) pretože mriežka je statická a v súčasnosti sa používajú zariadenia rôznych rozšírení obrazovky.



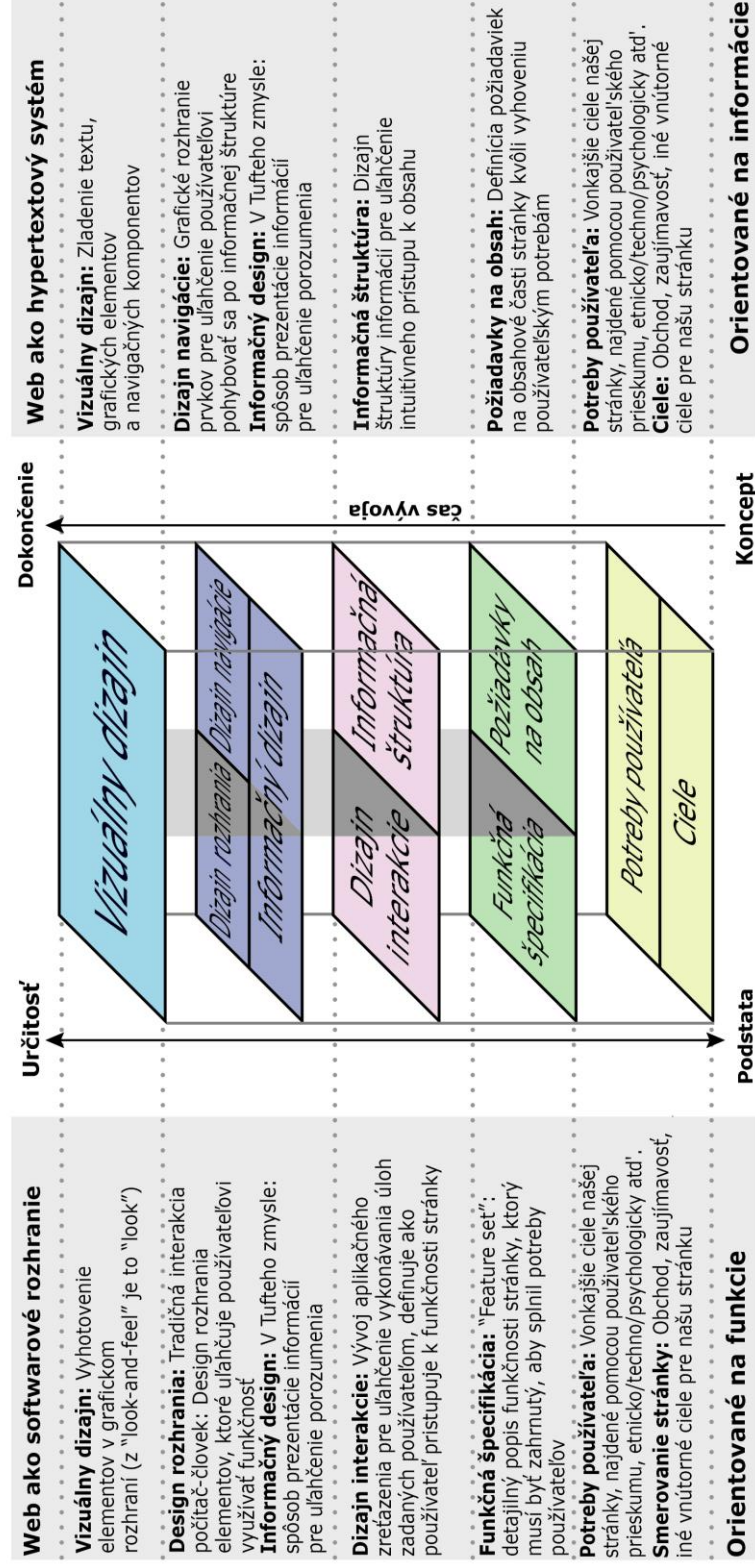
# Elementy "User Experience"

Jesse James Garrett  
jig@jig.net

30 March 2000

**Základ duality:** Web bol originálne myslený ako hypertextový informačný priestor, ale vývoj sofistikovanejších front/back-end technológií ho zmenil na remote software interface (vzdialené softwarové rozhranie). Táto dvojitá podoba vedie k výraznému zmätku, keďže používateľia "user experience" sa pokúšali adaptovať ich profesionálnu terminológiu na prípady nad rámec pôvodného využitia. Cieľom tohto dokumentu je definovať niektoré výrazy k prislúchajúcim kontextom a ozrejmiť podstatu vzťahov medzi jeho viacerými elementami.

Slovenský preklad  
vyhotovil  
Martin Kuchynář 2013



**Obrázok je nekompletný:** Opisovaný model neobsahuje druhotné úvahy (ktoré, napríklad, vznikajú počas tvorby obsahu či samotného programovania) to môže mať vplyv na rozhodnutia počas vývoja "user experience". Model taktiež neopisuje proces vývoja, ani konkrétne úlohy vo vnútri tímu "user experience". Model sa skôr usiluje definovať kľúčové úvahy, ktoré sa v súčasnosti používajú pri vývoji "user experience" na webe.

© 2000-2013 Jesse James Garrett

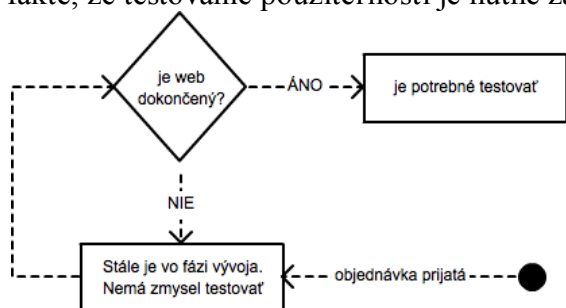
<http://www.jig.net/ia/>

# Testovanie použiteľnosti (usability)

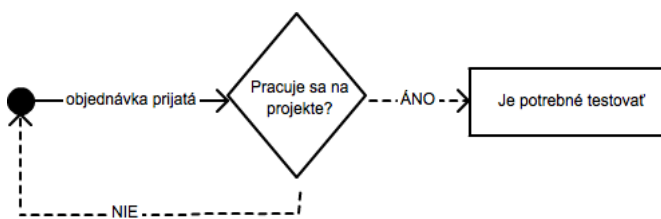
Použiteľnosť je silno spätá s UX. Pokiaľ sa UX sa zaoberá tým aké pocity ostávajú pri práci s našim produktom (v našom prípade web aplikácií), tak použiteľnosť študuje len to či sú užívatelia schopní vykonať úlohy, ktoré im zadáme pomocou nášho produktu. Možnosť vykonať štandardné úlohy býva kľúčová napríklad v internetových obchodoch, jednoducho ak zákazník (používateľ) nájde produkt a chce ho zakúpiť, nepodari sa mu to lebo formulár ktorým sa snaží odoslať. Používateľ to skúsi jedenkrát, dvakrát potom si pomyslí, že nevie niečo spraviť. Následne prichádza z jednej strany frustrácia používateľa, ktorý má pocit, že je nešikovný alebo hlúpy, to však jednoznačne nie je jeho vina. Tieto chyby (mimochodom veľmi zbytočné chyby) nie sú chybami používateľa ,ale dôsledkom chybného dizajnu produktov [15]. Pocity frustrácie jednoznačne nie sú pocity ktoré by sme radi dosiahli. Z druhej strany si zákazník nezakúpi nič a zisk jednoducho uniká. Pomocou úloh sa skryté snažíme zistiť 5 hlavných otázok podstatných pre použiteľnosť [16]:

1. Naučiteľnosť - Aké zložité je pre používateľa splniť základné úlohy, ak sa po prvýkrát stretol s produktom?
2. Zapamätateľnosť - Ak sa používateľ vráti k dizajnu ktorý už niekedy používal, aké budú jeho zručnosti teraz?
3. Efektivita – Ak je používateľ oboznámený s dizajnom, ako rýchlo vie dokončiť úlohy?
4. Chybovosť – Koľkých chýb sa používateľ dopustí, ako sa o tom používateľ dozvie, ako efektívne bude vedieť opraviť chyby?
5. Satisfakcia – Ako je spokojný s použitým dizajnom?

Toto sú okruhy ktoré zodpovedá práve použiteľnosť, ako ale prísť k odpovediam? Jedinou cestou je testovanie, samozrejme je potrebné brať v úvahu atribúty aj pri návrhu produktu. V tejto fáze máme dve možnosti. Prvá je vyrobiť produkt a následne ho nechať kompletne otestovať [17]. Druhá možnosť je príbežne testovať produkt počas produkcie a odstraňovať najhrubšie chyby okamžite [11]. Vzhľadom na nízkonákladovosť nášho projektu volíme druhú možnosť. Aby sme mohli začať testovať musíme mať niečo čo v skutočnosti funguje, toto nie je pravda. Ak sa odborníci z oblasti použiteľnosti v niečom zhodujú, tak je to vo fakte, že testovanie použiteľnosti je nutné začať čo najskôr [11, str. 35].



Obrázok 9: Nesprávne zmýšľanie



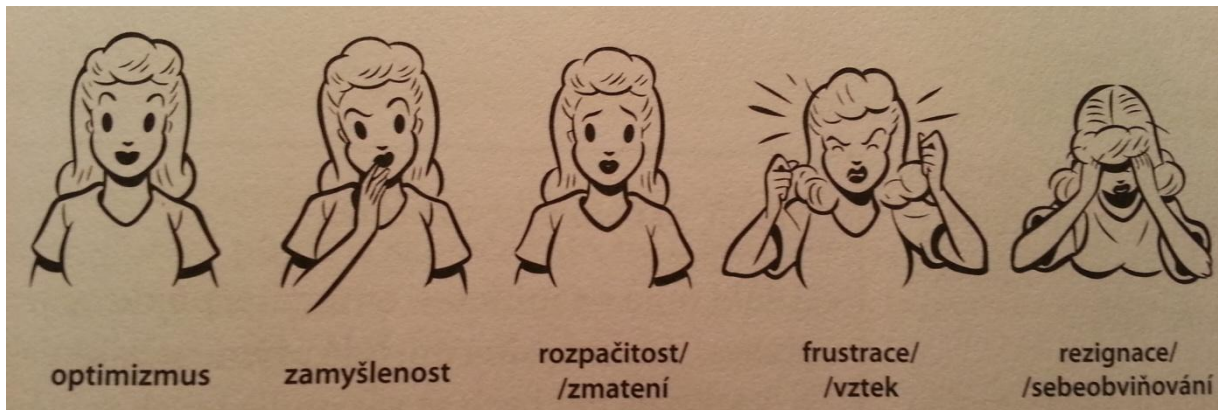
Obrázok 10: Správne zmýšľanie

Testovať môžeme každú z rovín, práve tento fakt nám dovoľuje začať testovať už v čase, keď sme presvedčení, že nemáme čo. Napríklad pri testovaní koncepčného modelu (štruktúrálna rovina) budeme postupovať nasledovne. Od prezentujeme aktuálnu predstavu koncepčného modelu, napríklad formou náčrtku na papierový obrúsok [19]. Následne klásť otázky: “Môžete mi povedať ako tomu rozumiete? Čo si myslíte, ako by to malo byť správne?” [11, str. 38]. Takýmto spôsobom vieme odstraňovať problémy, ktorých neskoršie nájdenie by mohlo spôsobiť veľké problémy. Navyše test takéhoto charakteru zaberie minimum času (od 5 minút do 10 minút na jedno sedenie).

Následne ak už testujeme rovinu vizuálneho dizajnu pristupujeme ku komplexnejšiemu testovaniu. Pri príprave testu je najdôležitejšie precízne pripraviť úlohy ktoré budú účastníci testu plniť [11, str. 59]. Tu je dôležité vedieť aké sú najdôležitejšie úlohy, mali by priamo vychádzať zo strategickej roviny. Následne prepísať úlohy do scenárov. Scenár je obdobný ako sa používa v herectve, musí vás oboznámiť s tým kto ste a čo sa snažíte dosiahnuť. Napríklad „Vy ste zákazník v internetovom obchode po dlhom monitorovaní ceny kosačiek na trhu ste sa rozhodli kúpiť si model X. Chcete odoslať objednávku na tento model, ale zároveň sa informovať o dodacích podmienkach.“ Dôležité je aby úlohy boli jasne zadané a zároveň neobsahovali detaily čo by naviedli účastníka k splneniu úlohy. Napríklad: „Vy ste zákazník v internetovom obchode po dlhom monitorovaní ceny kosačiek na trhu ste sa rozhodli kúpiť si model X. Chcete odoslať objednávku, *pomocou vloženia tovaru do košíka a vyplnenia dát*, ale zároveň sa informovať o dodacích podmienkach *v sekcii pripomienky*.“ Keď máme pripravené scenáre môžeme začať testovanie na reálnych účastníkoch, avšak pred samotným zadaním úloh je potrebné oboznámiť účastníka testu so základnými pravidlami:

- Nepoužívajte vyhľadávanie – Aby používatelia prirodzene hľadali potrebné informácie
- Zostaňte na tomto webe – Chceme testovať tento dizajn nie iný
- Rozmýšľajte nahlas – najdôležitejšie je aby sme vedeli ako rozmýšľa, aby sme vedeli či sa správa tak ako sme to naplánovali pri tvorbe
- Budete nahratí – nahrávať budeme iba pohyb myši po obrazovke a zvuk v miestnosti

Po oboznámení s pravidlami testu nasleduje Exkurzia po domovskej stránke, v tejto fáze sa snažíme zistiť či je na prvý pohľad jasné čo stránka reprezentuje. Následne začneme s plnením scenárov. Scenár vždy pomaly prečítame, uistíme sa, že účinkujúci pochopil úlohu a prenecháme im dostatok času na splnenie úlohy. Najdôležitejšie je byť pozíciou neustranného pozorovateľa, v žiadnom prípade nedávať nápovede ako by mal postupovať, ďalej je potrebné dôkladne sledovať ako sa účastník počas testu tvári.



Obrázok 11: Ilustrácie emócií [11]

Neustále je potrebné pripomínať, aby testujúci účastník rozmýšľal nahlas. Je potrebné, aby vývojári mohli sledovať priamo testovanie (v horšom prípade iba záznam), potom je vhodné usporiadať poradu a názorne si rozobrať len najdôležitejšie problémy a navrhnúť ich riešenia.

# Špecifikácia

---

V tejto kapitole sa začneme zaoberať praktickou realizáciou teórie, opísanej v predchádzajúcej kapitole. Pri špecifikácii v tradičnom vývoji sa zaoberáme výberom vhodnej technológie pre daný produkt, a kompletizovaním funkcionality, ktorú očakávame od produktu. V tejto práci sa však pokúšam prezentovať a predviesť nový metodický prístup k tvorbe internetových produktov, konkrétne portálov. Preto sa mierne odkloníme od štandardného modelu špecifikácie, a prikloníme sa k 5 vrstvovému návrhu J. J. Garretta ktorá popisuje celý vývoj takýchto portálov.

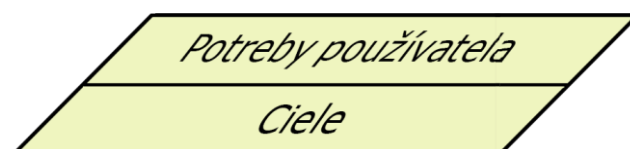
Kompletne zadefinujeme strategickú rovinu, rozsahovú rovinu. Nakoniec načrtujeme štruktúrnu rovinu, ktorá tvorí pomyselné rozhranie medzi plánovaním (a teda špecifikáciou projektu) a reálnou implementáciou. V ďalších podkapitolách sa budem zaoberať výhradne výstavbou konkrétneho informačného klubu pre športový klub.



## Strategická rovina

---

Na tejto rovine sme definovali dve najdôležitejšie otázky, na ktoré hľadáme odpovede.



### Ciele

---

Čo chceme my dosiahnuť pomocou tohto produktu?

1. Ľahký prístup k štatistikám o tréningoch
2. Vzrast súťaživosti hráčov
3. Teambuilding hráčov klubu
4. Zjednodušenie koordinácie trénerov TJ. tréningov, turnajov, sústredení
5. Zhromažďovanie vhodných cvičení, tréningových metód

### Potreby používateľa

---

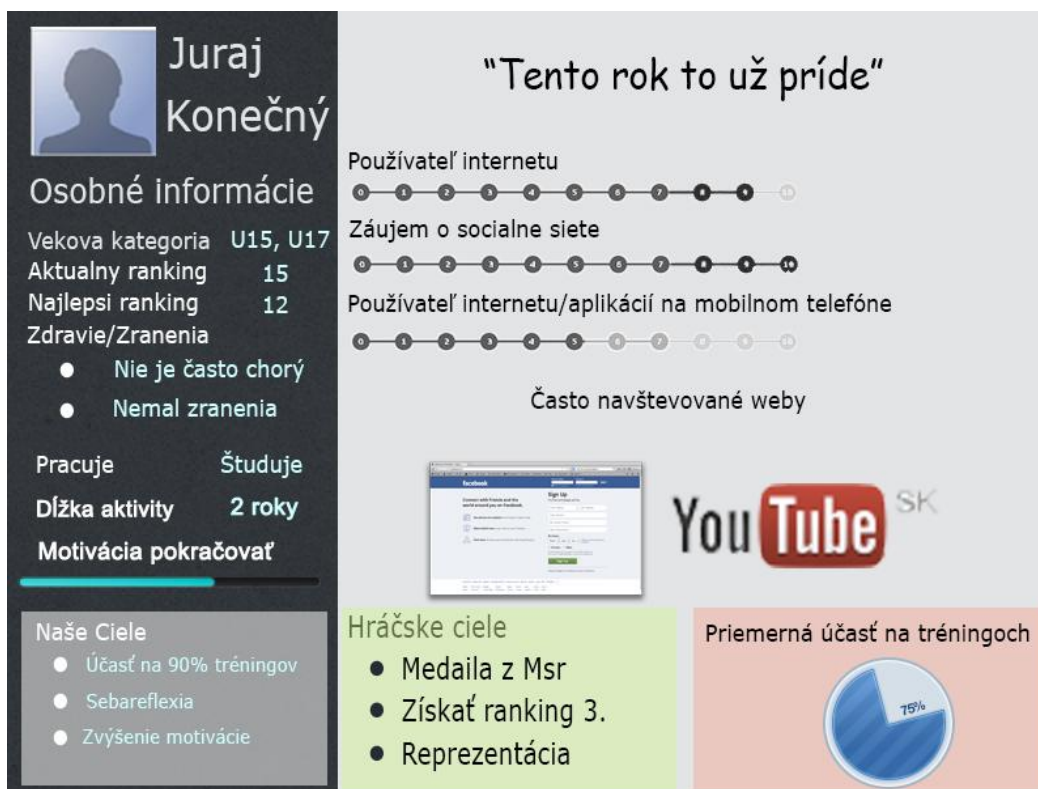
Čo naši zákazníci v skutočnosti chcú od nášho produktu?

1. Potreby používateľov je v našom prípade jednoduché rozdeliť do 3 charakteristických skupín používateľov
  - a. Hráči
  - b. Tréneri
  - c. Vedúci klubu

Každú skupinu vyčerených užívateľov demonštrujeme pomocou osoby.



**Hráči** - primárne portál nepotrebujú, ale chcú vedieť ako sa darí ich spoluhráčom, prípadne súperiť (gamification), chcú mať vždy isto či sa tréning koná alebo nie.




Obrázok 12: Persóna pre hráčsku skupinu

**Tréneri** - potrebujú medzi sebou koordinovať tréningy, potrebujú motivovať hráčov, chcú zlepšovať svoje metódy.



Obrázok 13: Persóna pre trénerskú skupinu

**Vedúci klubov** potrebujú evidovať a vyplácať trénerov, rozdeľovať výjazdy medzi trénerov, mať prehľad o výkonnosti hráčov



**Filip Bystrý**

**Osobné informácie**

Veková kategória **dospelý**

Trénerská trieda **1**

Príprava na trening **20 min**

Na administráciu klubu potrebujúe **2hod/deň**

Pracuje **Študuje**

Dĺžka aktivity **10 rokov**

Motivácia pokračovať

---

**Naše Ciele**

- Kontakt na hráčov, trénerov
- Podporiť súťaživosť hráčov
- Maximálna štatistika diania na tréningoch


**"Nestíham všetko vybaviť, ale klub musí fungovať bezchybne."**

Používateľ internetu  
0 1 2 3 4 5 6 7 8 9 10

Záujem o socialne siete  
0 1 2 3 4 5 6 7 8 9 10

Používateľ internetu/aplikácií na mobilnom telefóne  
0 1 2 3 4 5 6 7 8 9 10

Často navštevované weby



**Vedúceho ciele**

- Víťaztvá hráčov
- Jednoduchá administrácia, presná štatistika
- Mať všetko pod kontrolou

Obrázok: 14 Persóna pre skupinu vedúceho klubu

Z týchto informácií vytvoríme strategický diagram obsahujúci všetky rozhodnutia na tejto rovine.





**Hráč**



**Tréner**



**Vedúci klubu**

Obrázok 15: Strategický konceptuálny model



# Rámcová rovina

---

Rovinu rozdeľujeme na časť funkcionality a stranu obsahu. Z duálneho pohľadu na web je táto rovina rozdelená na funkčnú špecifikáciu a požiadavky na obsah.



Zo strategického konceptuálneho modelu postupne dostávame kompletný zoznam očakávaní funkcionality s priradenými prioritami (uvedené v zátvorkách, 1 je najvyššia priorita). Pre korektnosť nakoniec uvedieme formálny príklad špecifikácie jednej zo zoznamu názvov.

## Funkčná špecifikácia

---

- a. Udržiavanie štatistík o tréningoch, zápasoch (1)
    - i. Pravidelne tréningy
    - ii. Editácia, pridávanie, mazanie
    - iii. Zápasy medzi ľubovoľnými počtami hráčov
  - b. Vedenie štatistík o práci trénerov, pre vyplácanie plátov (2)
  - c. Hra o kráľa klubu (gamification) (1)
  - d. Posielanie správ medzi používateľmi (2)
    - i. Dohováranie odchodov na turnaje, vedúcich na turnaje (hromadná sprava)
    - ii. Klasický chat
    - iii. Plánovanie tréningového procesu
- Online tréner (3)
- iv. Tréneri môžu pridávať cvičenia
    - 1. Možnosť určiť náročnosť cvičenia pomocou usporiadanej trojice technika, taktika, kondícia
  - v. Hľadanie cvičení podľa zadaných atribútov
  - vi. Generovanie tréningu podľa zadaných požiadaviek

## Špecifikácia C) hra o hráča klubu

---

Vstup: Evidencia účasti na tréningoch, záznamy o zápasoch.

Výstup: Zoradený zoznam hráčov podľa výkonnosti.

## Požiadavky na obsah

---

Podľa požiadaviek vyplývajúcich z požiadaviek definovanej na strategickej rovine, sa budeme snažiť, aby kompletný obsah mohol byť dynamický a ľahko editovateľný používateľmi (priblíženie sa k web 2.0[12]). Z tohto dôvodu, spolu s faktom, že tvoríme informačný systém pre menší športový klub je vhodné použiť content management systém, aby aj bežný používateľ bez potreby znalosti takéhoto systému mohol riadiť a meniť systém. Takéto organizácie totiž so pravdepodobnosťou blížiacou sa k jednotke nebudú mať prostriedky na správu systému vlastným programátorom. Na výber sa nám ponúka množstvo voľne dostupných systémov pre príklad spomením open source systém Joomla[13], ďalšou možnosťou je vývoj vlastného CMS. Použitie systému typu Joomla nám plne nevyhovuje pretože takýto systém je pre naše účely príliš prepracovaný, čo zväčšuje náklady na hostingové služby, aj predlžuje vývoj samotnej aplikácie, najviac sa systém vďaka komplexnej štruktúre (ktorá musí pokryť všetky typy web stránok od propagačných až po internetové obchody) stáva neprehľadnou. Nie je štandardom podporovať dátové štruktúry ktoré podporujú resp. sú použiteľné na ukladanie zápasov alebo tréningov, preto je potrebné ich doprogramovať aj pri takýchto systémoch.

Pre projekt takéhoto typu vyberiem súkromný CMS systém, ktorí neponúka také veľké množstvo predprogramovaných možností avšak jeho úprava pre konkrétny účel je jednoduchšia. Pre potreby nášho informačného systému som sa rozhodol vyžiť jadro CMS napísaného Michalom Novomeským, ktoré bolo súčasťou jeho bakalárskej práce[8]. Avšak aj súbory jadra systému som sa rozhodol modifikovať triedu obsluhujúcu databázové spojenia[8, str. 18], z dôvodu zastarania verzie PHP jazyka. Jadro systému je napísané v PHP5 ja som ho aktualizoval pre PHP5.2 praktický rozdiel je napríklad pri prístupe k databáze, kde je možné využívať plne objektovo orientovanú PDO[14] triedu pre prácu s rôznymi databázovými systémami.

Výhody takéhoto objektu sú napríklad aj pri bezpečnosti a rýchlosti vykonávania databázových dotazov (query's). PDO objekty nám dovoľujú tvorenie parametrizovaných databázových dotazov. Ich výhoda sa nachádza v 100% bezpečnosti voči injekciám (query injection) [20] v dotazoch.

Dosahuje sa to rozdielnym prístupom spracovania dotazu, a to tak ,že databáze ešte pred dosadením hodnôt pripraví všetko čo dokáže bez dosadenia hodnôt. Následne databáze do pred kompilovaného kódu odošle hodnotu na dosadenie.

Nevýhody takéhoto prístupu sú v požiadavkách na server kde musí byť dostatočne aktualizovaná verzia jazyka PHP aspoň vo verzii 5.1.0 alebo je nutnosť inštalácie prídavných modulov[14].

Ďalšie zmeny boli vykonané na databázovom modeli. Kde som pridal bezpečnostný stĺpec soľ ku tabuľkám používateľom (v implementácii je to tabuľka Users). Význam reprezentujem príkladom.



id_user	enabled	loginname	nickname	password
1	1	Nubian	Nubian	c03eafb5c40234d0a34269846975ac8
2	1	admin	admin	c03eafb5c40234d0a34269846975ac8
4	1	mato	Mato	c03eafb5c40234d0a34269846975ac8

Obrázok 16: Tabuľka pred úpravami

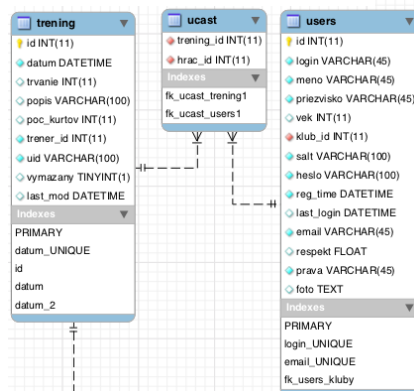
Na výpise z databázy je ľahko nahliadnuteľné, že všetci užívatelia majú rovnaké heslo, útočník však nevie aké to heslo keďže vidí iba jednosmerne vytvorený hash. Ak by bol útočníkom napríklad užívateľ Nubian, a získal by takýto výpis, automaticky sa dozvie hesla ostatných užívateľov. Mohol by sa prihlásiť ako administrátor systému a vykonávať zmeny na ktoré nie je oprávnený.

id	login	meno	priezvisko	v	k	salt	heslo
1	Nubian	Nubian	Nubian	1	1	5159aa8231035	189d259163113901c27b87fd6ccad81e
2	Kuchy	Kuchy	Kuchy	1	1	5159aa8231183	983a2543459238765t2sagegi3crtd56q
4	mato	mato	mato	1	1	5150abe76352b	b875c4a86375c5ec444d0d22fe731d1a

Obrázok 18: tabuľka po úpravách

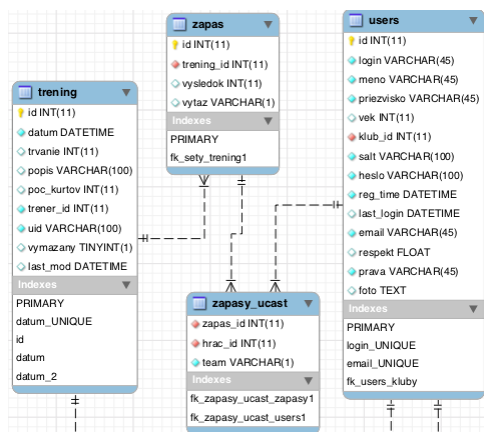
Riešenie je prídanie ďalšieho stĺpca ktorý bude unikátny pre každého používateľa. nazvime ho soľ (salt), keď následne tvoríme šifru hesla napísanú užívateľom modifikujeme šifru vytvoríme pomocou nie iba zadaného hesla, ale spojenia soľ a hesla, následného zašifrovania. Preto aj ak majú všetci užívatelia rovnaké heslo, hash dostanú iný. Ak by prišlo k úniku informácií z databázy útočník nedokáže rozšifrovať heslo inak ako skúšaním všetkých možností.

Ďalšie modifikácie nastali spolu s úpravou štandardu html 4 do štandardu html 5.



Obrázok 17: DB riešenie pre účast'

Ďalšou, už vyššie spomínanou časťou práce na tejto rovine je poskytnúť potrebné zázemie pre plánované funkcie. V špecifikácii sa nachádzajú štatistiky o tréningoch. Pre ich výrobu budeme potrebovať viacero tabuliek. Prvou bude tabuľka hovoriaca o samotnom tréningu. Na každom tréningu je teoreticky možná účasť ľubovoľného počtu hráčov a ich zápasov, preto použijeme ďalšie tabuľky ktoré previažeme pomocou cudzích kľúčov (FOREIGN KEY's). Tabuľka „účasť“ bude reprezentovať účasť jednotlivého hráča na jednotlivom tréningu. Tabuľka bude poskytovať referenciu, že používateľ s istým Id sa zúčastnil na tréningu s istým Id. Takáto tabuľka nám dovolí zaevidovať ľubovoľný počet hráčov na tréningu, a zároveň samotný databázový systém

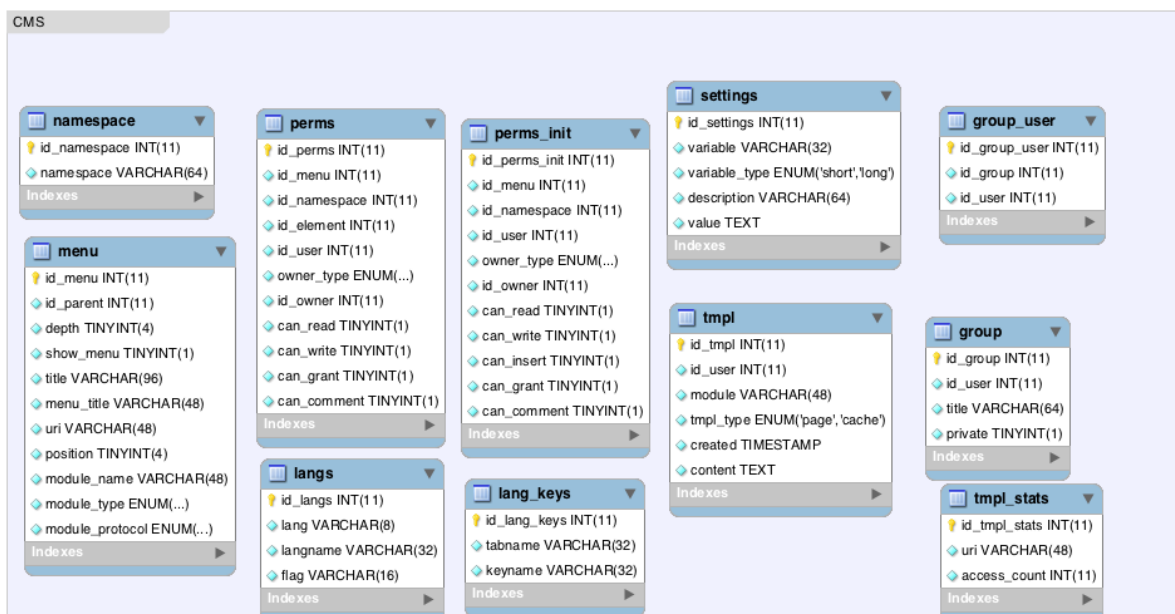
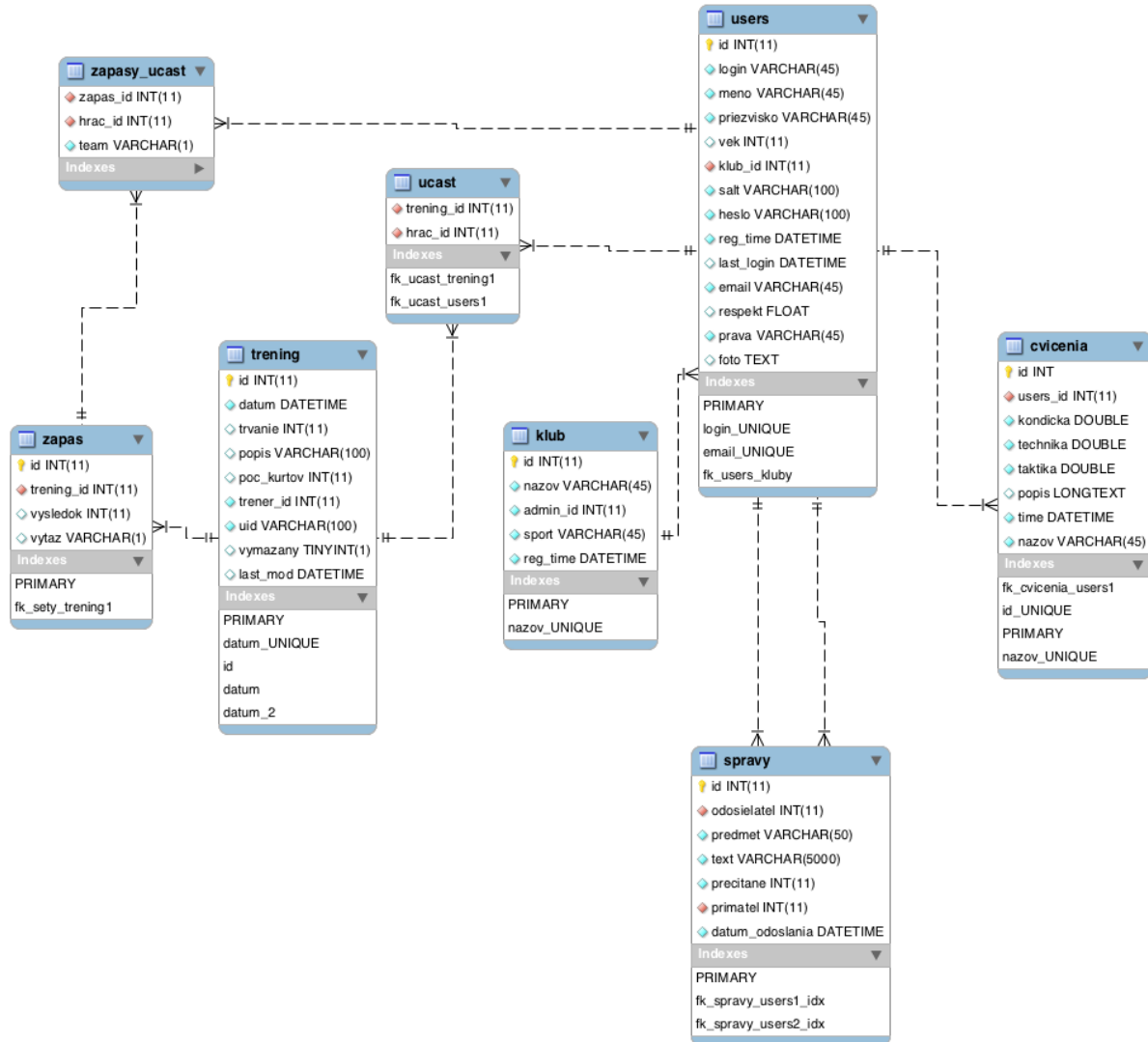


Obrázok 19: DB riešenie pre zápasy

nedovolí zadať účasť hráča aký neexistuje, na tréningu ktorí sa neuskutočnil (rovnako to funguje pri editácii a mazaní záznamov).

Obdobným problémom sú zápasy, taktiež ich môže byť ľubovoľný počet na tréningu, takže zvolíme rovnaký postup. Pri návrhu zápasov však musíme počítať s možnosťou jednak ľubovoľného množstva zápasov na tréningu ale taktiež musíme počítať s možnosťou rôznych druhov zápasov. Napríklad v badmintone poznáme 3 druhy disciplín a to 2hru 4hru a zmiešanú 4hru, pritom pri tréningoch sa veľmi často používajú zápasy 1hráča proti 2 alebo 3 hráčom. Informačný systém preto aby mohol byť používaný musí vedieť podporovať všetky druhy takýchto zápasov. Preto som vytvoril ďalšiu tabuľku referujúcu na zápas a hráča v ňom hrajúceho spolu s tímom v akom štartoval. Takýto architektonický krok nám poskytuje nástroj na vytváranie zápasov s ľubovoľným počtom hráčov súperiacich proti sebe.

Konečná databázová schéma pre náš informačný systém je nasledovná.

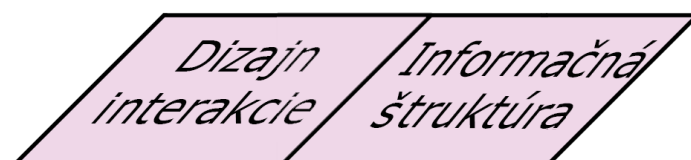


Obrázok 20 Kompletné DB riešenie

# Implementácia



## Štrukturálna rovina



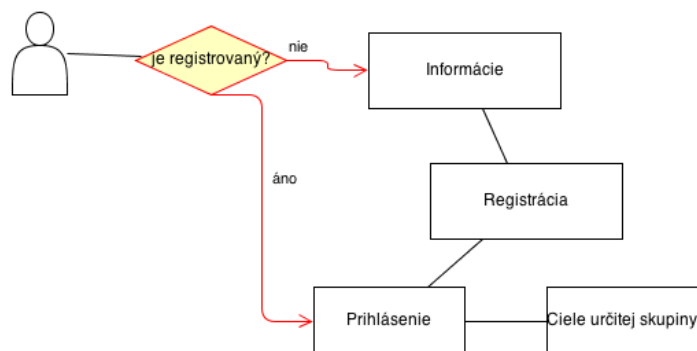
### Dizajn interakcie

Hlavnou úlohou tejto časti roviny je vytvoriť koncepčný model, a teda predstavu akú bude mať užívateľ o fungovaní nášho systému. Táto predstava nemusí ani v najmenšom odpovedať skutočnému fungovaniu, je však dôležité aby odpovedala predstave používateľa. Musíme však počítať aj so situáciami keď sa užívateľ dostane do úzkych alebo sa vymykne takémuto modelu. Docielime to dostatočným informovaním používateľa v takýchto situáciách (Error handling), pomocou chybových hlásení. Chybové hlásenia musia nielen zabrániť užívateľovi v chybných akciách ale aj jasne objasniť prečo je takáto akcia chybná. Oznámenie zobrazené na obrázku užívateľovi nepovie nič len sa niečo stalo. Takýto prístup je často pre užívateľov frustrujúci lebo nerozumejú čo sa deje, a logicky sa cítia ako hlúpy a vinný za chyby dizajnu takýchto hlásení. Ďalšou závažnou chybou takýchto hlásení je snaha používať jedno upozornenie, na veľa rôznych prípadoch, kde nie je povedaná presná

príčina problému. V našej aplikácii rozdelíme chyby do hlavných sekcií ktoré budú mať rovnaký vzor zobrazovania (error template), a rozdielny obsah chyby. Hlavné sekcie budú: validačná, práva, potvrdenie. Použitím vzorov docielime okamžité rozoznanie druhu chyby a keby užívateľ aj nečítal konkrétny text popisujúci chybu (čo sa stáva viac ako bežne) vie približne kde je chyba.



Obrázok 21  
Bezvýznamný error  
template

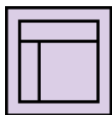


Obrázok 22 Koncepčný model domácej stránky

## **Informačná štruktúra**

---

Informačnú architektúru budeme tvoriť pomocou prístupu zhora-nadol, z dôvodu veľkej flexibility. Systém nie je určený pre konkrétny športový klub, preto je dôležité aby bolo možné neskôr ju ľahko meniť.



# Konštrukčná rovina

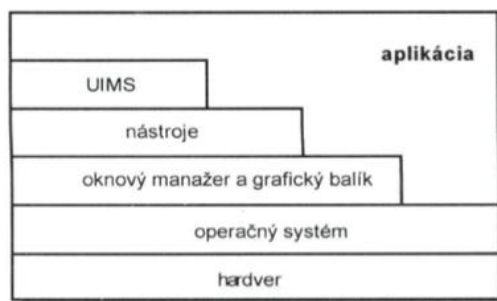


## Dizajn rozhrania

Dizajn rozhrania definuje dizajn ako používateľ zadáva vstupy pre riadenie funkcií. Systém má veľké množstvo funkcií, preto budeme prezentovať jednu vybranú funkciu. Ako príklad uvedieme posielanie správ medzi používateľmi (chat). V tejto funkcionalite zvolíme konzervatívny prístup. Preberieme rozhranie na ktoré sú užívatelia zvyknutý. V strategickej rovine sme od segmentovali 3 používateľské skupiny, z ktorých sú 2 familiárne s používaním sociálnych sietí a z toho dôvodu budeme tvoriť tak tvoriť rovnaký dizajn rozhrania. Takýmto kompromisom docielime zlepšenie naučiteľnosti u väčšiny používateľov. Ostatné funkcie je možné odsledovať z implementácie (V prílohe práce).

## Dizajn navigácie

Navigáciu navrhne podobne ako informačný dizajn. V dizajne navigácie budeme hľadať miesto a konkrétny druh navigácie. Konceptný model nám rozdeľuje obsah na 2 časti preto aj navigácia bude rozdielna na Domácej stránke a stránke po prihlásení. Je zbytočné aby neprihlásený používateľ mal v ponuke obsah ku ktorému mu systém nedovolí prístup, preto zavedieme na domácu stránku iba tučnú pätičku (big footer), ktorá bude v pozadí a bude odkazovať na menej potrebné informácie o projekte. Po prihlásení navigácia zmení podobu na klasickú globálnu navigáciu. Táto navigácia obsahuje konkrétne funkcie. Hierarchiu zobrazovania funkcií nebudeme určovať, tieto rozhodnutia môžu vykonávať vedúci klubu pomocou systému správy užívateľského rozhrania (UIMS) GUI [21, str. 246]. Hlavným prínos spočíva v možnosti zoradenia web stránok do ľubovoľných sekcií, takýmto konaním používateľ sám vytvorí štruktúru navigácie tak ako ju sám používa.



Obrázok 23 Úrovne softwaru pre GUI [21, str. 247]

# Informačný dizajn

Budeme postupovať presne podľa koncepčného modelu zadaného na predchádzajúcej rovine. Z neho vyplýva, že systém je primárne určený pre používateľov so založeným kontom v istej používateľskej skupine, preto budeme tvoriť dve úplne rozdielne rozhrania. Prvé je založené na prvej časti koncepčného modelu, používateľ bez registrácie, druhé bude založené po prihlásení používateľa do systému.

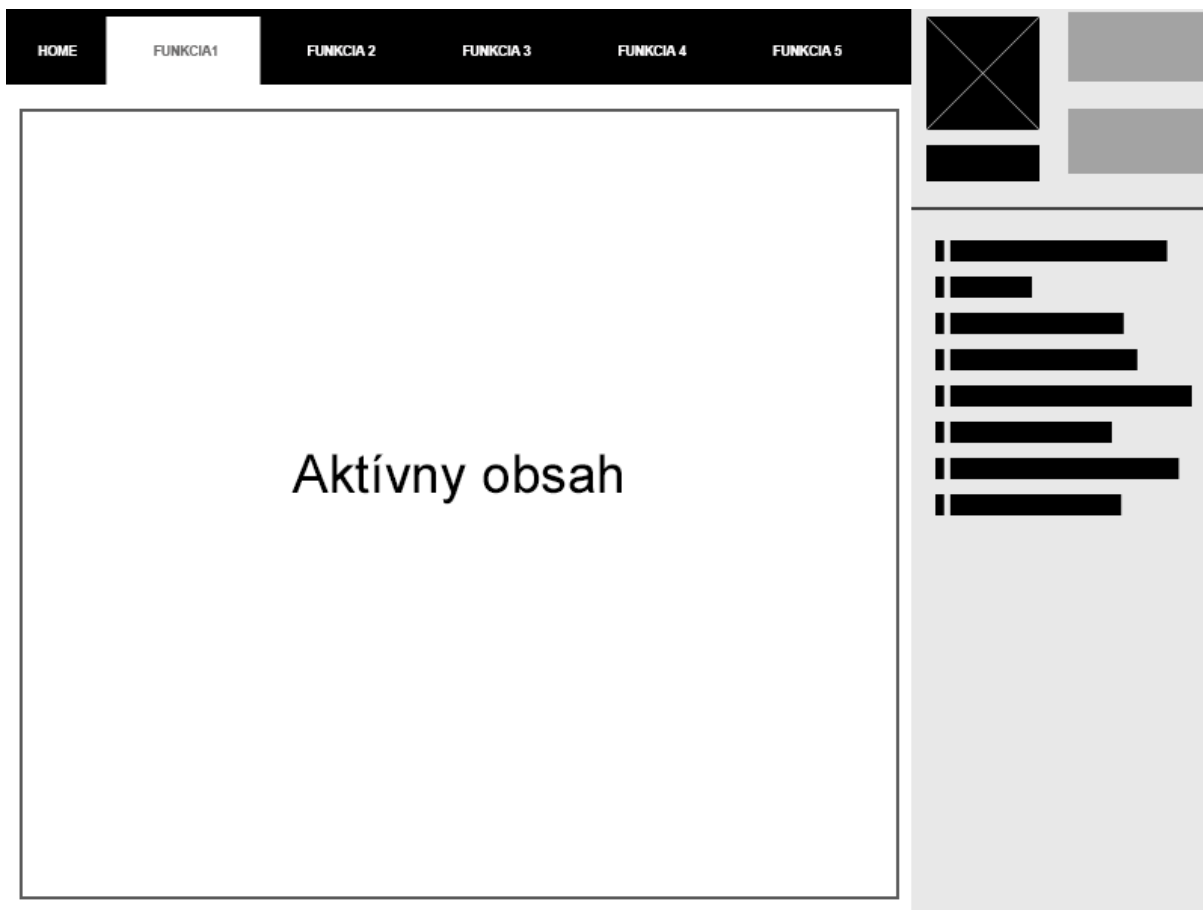
Prvý dizajn bude založený na jednoduchosti, podľa koncepčného modelu musí ponúknuť jasnú cestu neregistrovanému používateľovi k registrácii a prihláseniu. Preto rozdelíme domácu stránku na časť prihlásenia/registrácie a časť informačnú, ktorá vedie k registrácii používateľa. Informačná časť bude výraznejšia, pretože je určená pre nových používateľov. Prihlasovacia časť bude menej výrazná pretože je používaná iba už zaregistrovaných používateľmi.

Stránka po prihlásení bude mať úplne inú štruktúru. Podľa koncepčného modelu v tejto časti je nutné poskytnúť prístup ku všetkým informáciám pre určitú skupinu používateľov, preto zvolíme kde rozdelíme prvky na statické a dynamické. Statickými prvkami bude chat, navigácia a používateľský panel, k týmto elementom potrebuje používateľ neustáli prístup. Dynamický obsah bude závisieť od práve prehladanej funkcie.

Celkový výsledok takejto roviny budeme reprezentovať pomocou Wireframov.



Obrázok 24 Wireframe pre domácu stránku



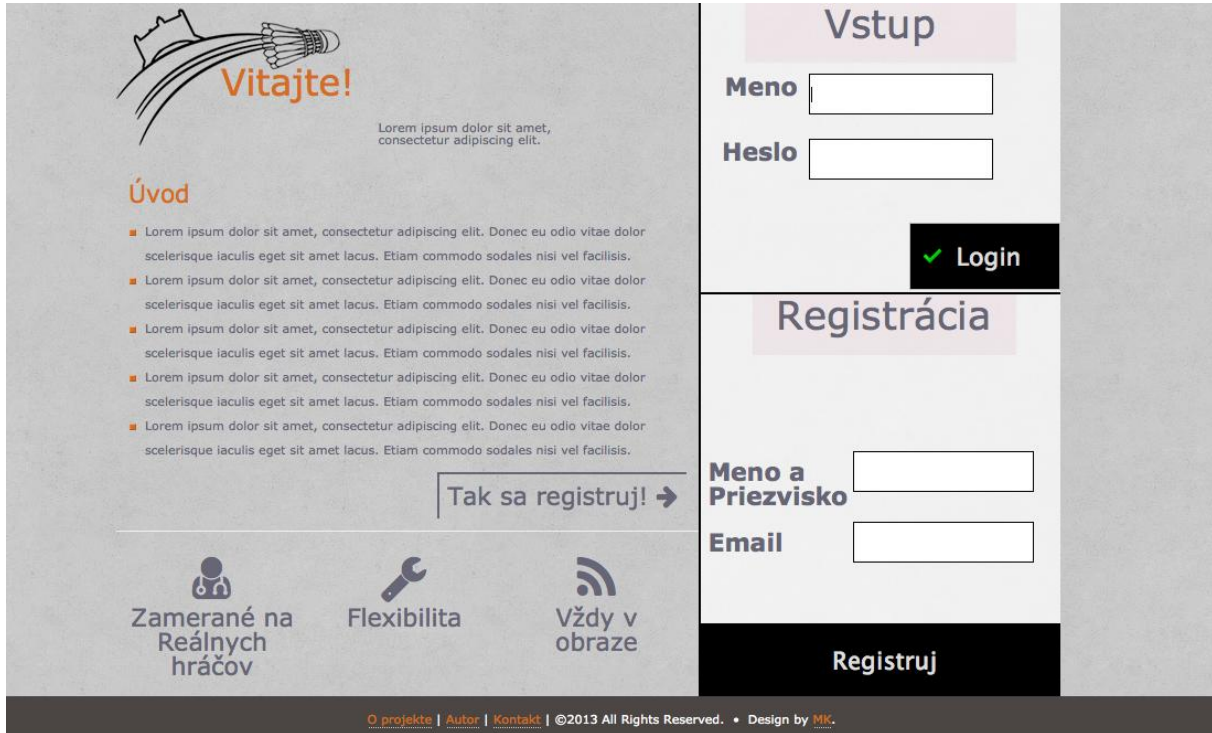
Obrázok 25 Wireframe po prihlásení užívateľa



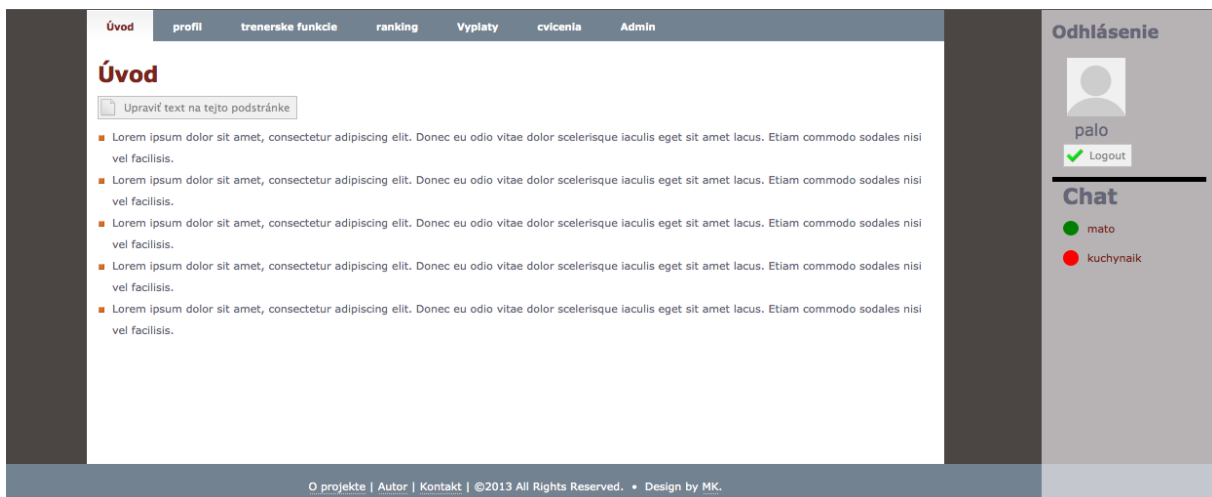


# Povrchová rovina

Rozhodnutia na povrchovej rovine demonštrujeme, ukázkami výslednej stránky. Celý projekt je dostupný v prílohe. Na základe zistení v strategickej rovine, sme sa rozhodli vytvoriť zodpovedajúci dizajn určený pre 2 rôzne platformy: počítače a mobilné zariadenia [23].



Obrázok 26: Výsledný vzhľad stránky pred prihlásením



Obrázok 27: Výsledný vzhľad stránky po prihlásení

# Záver

---

Táto práca ponúka prehľad pojmov v oblasti UX, ich metodické zaradenie do chronologického vývoja web projektov. Udáva príklad ako sériu rozhodnutí pri aplikovaní takejto metodiky ako aj kompletnú implementáciu modelového projektu. Ďalej sme navrhli slovenský preklad diagramu Element's of User Experience od J. J. Garretta.

Implementácia, obsiahnutá v práci, bola pred začatím písania web aplikácia v podobe neflexibilného systému určeného jednému športovému klubu. Úprava nebola možná. Dnes tento systém je plne editovateľný používateľom, zameraný na presne vymedzené používateľské skupiny, pre ktoré spĺňa ich ciele, najviac je systém použiteľný pre ľubovoľný šport.

Verím, že práca bude prínosom pre čitateľov, ktorí nie sú familiárni s tvorbou používateľsky zameraných (User-Centered Design) web systémov a na jej základe objavia nový rozmer vývoja aplikácií, pre ich efektívnejšie využitie v reálnom svete. Slovenský preklad diagramu Element's of User Experience bol ponúknutý jeho autorovi na uverejnenie.

V budúcnosti by som rád prácu podrobil testovaniu na reálnych užívateľoch, zároveň dobudoval funkcionality vyplývajúcu z testovania, a vytvoril plne zodpovedajúci dizajn [23].

# Zdroje

---

- [1] Reenskaug, T. 1979. *MODELS - VIEWS - CONTROLLERS*[online]. [cit. 2013-05-19]. Dostupné na: <http://heim.ifi.uio.no/~trygver/1979/mvc-2/1979-12-MVC.pdf>
- [2] GUTMANS, A. 1999. [cit. 2013-05-19]. Portál: <http://framework.zend.com/>
- [3] OPEN SOURCE. 2007. [cit. 2013-05-19]. Portál: <http://symfony.com/what-is-symfony>
- [4] TATARYNOWICZ. M. [cit. 2013-05-19]. Portál: <http://cakephp.org/>
- [5] GARRETT, Jesse James. *The elements of user experience: user-centered design for the Web and beyond*. 2nd ed. Berkeley, CA: New Riders, c2011, xviii, 172 p. Voices that matter. ISBN 03-216-8368-4.
- [6] NIELSEN, J. *Web design*. Vyd. 1. Preklad Ladislav Valík. Praha: SoftPress, 2002, 382 s. ISBN 80-864-9727-5.
- [7] BLYTHE, M, et al. *Funology from usability to enjoyment*. [Online-Ausg.]. Dordrecht: Kluwer Academic, 2003. ISBN 14-020-2967-5.
- [8] NOVOMESKÝ M. 2008. *Podporný komunitný portál medzinárodného študijného programu MEi: CogSci* [online]. [cit. 2013-05-19]. Dostupné na: <http://nubian.sk/bakalarka/spec.pdf>
- [9] *What is content management?* [online]. [cit. 2013-05-19]. Dostupné z: <https://www.vasont.com/resources/what-is-content-management.html>
- [10] NIELSEN, J & PERNICE, K. *Eyetracking web usability*. Berkeley, CA.: New Riders, 2010, xix, 437 p. ISBN 978-032-1498-366.
- [11] KRUG, S. *Nenuťte uživatele přemýšlet!: praktický průvodce testováním a opravou chyb použitelnost webu*. Vyd. 1. Brno: Computer Press, 2010, 165 s. ISBN 9788025129234.
- [12] DINUCCI, D. *Fragmented future*. 1999. [cit. 2013-05-19]. Dostupné z: [http://www.darcyd.com/fragmented\\_future.pdf](http://www.darcyd.com/fragmented_future.pdf)
- [13] Joomla. [online]. 2005 [cit. 2013-05-19]. Dostupné z: <http://www.joomla.org/>
- [14] PHP Data Objects [online]. 2013 [cit. 2013-05-19]. Dostupné z: <http://php.net/manual/en/book.pdo.php>
- [15] NORMAN, Donald A. *Design pro každý den*. 1. vyd. v českém jazyce. Praha: Dokořán, 2010, 271 s. ISBN 978-80-7363-314-1.
- [16] Usability 101: Introduction to Usability. *Nielsen Norman Group* [online]. 2012 [cit. 2013-05-19]. Dostupné z: <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>
- [17] NIELSEN J, 1999. *Designing Web Usability: The Practice of Simplicity ??*
- [18] NORMAN, D. *Design pro každý den*. 1. vyd. v českém jazyce. Praha: Dokořán, 2010, 271 s. ISBN 978-80-7363-314-1.
- [19] ROAM, D. *Nápady na ubrousku: řešte problémy a prezentujte myšlenky pomocí obrázků*. Vyd. 1. V Brně: Jan Melvil, c2009, 287 s. ISBN 978-809-0391-291.
- [20] *SQL injection*. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-05-19]. Dostupné z: [http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)
- [21] RUŽICKÝ, E & Andrej F. *Počítačová grafika a spracovanie obrazu*. Bratislava: SAMOSATO, 2012. ISBN 978 – 80 – 89464 – 08 – 1.
- [22] TUFTE, Edward R. *The visual display of quantitative information*. 2nd ed. Cheshire: Graphics Press, 2007, 197 s. ;. ISBN 978-0-9613921-4-7.
- [23] MARCOTTE, Ethan a [foreword by Jeremy KEITH]. *Responsive web design*. New York: A Book Apart, 2010. ISBN 978-098-4442-577.

# Prílohy

---

K tomuto dokumentu je priložené CD so samotným systémom. CD taktiež obsahuje návod na inštaláciu systému.