



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

UŽIVATELSKÉ ROZHŘANÍ PRO SPRÁVU DIGITÁLNÍCH PŘEDNÁŠEK

USER INTERFACE FOR MANAGEMENT OF DIGITAL LECTURES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DANIEL KOŠTIALIK

VEDOUCÍ PRÁCE

SUPERVISOR

prof. Ing. HEROUT ADAM, Ph.D.

BRNO 2016

Abstrakt

Cielom tejto práce je navrhnuť a implementovať užívateľské rozhranie pre správu digitálnych prednášok a ich sprievodných materiálov. Podstatou môjho riešenia je vytvorenie sieťových štruktúr pozostávajúcich z jednotlivých videí, ktoré spolu navzájom súvisia a poskytujú možnosť touto štruktúrou prechádzať. Aplikácia ďalej umožňuje užívateľovi vytvárať vlastné zoznamy videí a pracovať s nimi ako s reálnymi záložkami na nástenke. Testy na cieľovej skupine dokázali, že návrh prináša nový pohľad na usporiadanie a prehliadanie videí. Hlavným prínosom tejto práce je, že navrhnuté rozhranie umožňuje lepšie využiť potenciál video prednášok a takto zefektívniť proces učenia.

Abstract

The aim of the thesis is to design and program user interface for administration of digital lectures and their accompanying materials. The fundament of my approach creates network structure of individual videos that are related with each other with ability to look through them. The application allows user to create its own video lists and work with them like with real bookmarks on a notice board. Trials on the target group prove that it brings new view on arrangement and video watching. The main contribution of the thesis is that designed interface provides better utilisation of video lectures and thus makes the process of learning more efficient.

Klíčová slova

Užívateľské rozhraní, digitální přednášky, video přednášky, HTML, PHP, testování uživatelského rozhraní

Keywords

User Interface, digital lectures, video lectures, HTML, PHP, user interface testing

Citace

KOŠTIALIK, Daniel. *Užívateľské rozhraní pro správu digitálních přednášek*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Herout Adam.

Uživatelské rozhraní pro správu digitálních přednášek

Prohlášení

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením prof. Ing. Adama Herouta, Ph.D. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Daniel Košťalik
13. května 2016

Poděkování

Chcel by som poďakovať svojmu vedúcemu prof. Ing. Adamovi Heroutovi, Ph.D. za jeho užitočné rady, vedomosti a trpezlivosť pri riešení tejto bakalárskej práce.

© Daniel Košťalik, 2016.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1 Úvod	2
2 Teoretický základ	3
2.1 Digitálne prednášky	3
2.2 Existujúce a podobné riešenia	3
2.3 Užívateľské rozhrania a ich návrh	5
2.4 Testovanie užívateľského rozhrania	9
2.5 Implementačné nástroje	11
3 Analýza cieľového užívateľa	16
4 Návrh riešenia	18
4.1 Forma digitálnych prednášok a vzťahy medzi nimi	18
4.2 Návrh funkcionality aplikácie	19
4.3 Užívateľské rozhranie	20
5 Implementácia	25
5.1 Výber nástrojov	25
5.2 Dátová štruktúra	25
6 Testovanie	27
6.1 Úpravy na základe výsledkov testov	28
7 Návrhy na pokračovanie	29
8 Záver	30
Literatura	31

Kapitola 1

Úvod

Používanie moderných technológií je pre nás každodennou rutinou a náš život si bez nich už len ťažko vieme predstaviť. Tieto technológie zasahujú takmer do všetkých oblastí nášho života akými sú domácnosť, priemysel, marketing, zdravotníctvo, doprava a tak isto neobišli ani oblasť vzdelávania. V oblasti vzdelávania je pokrok na veľmi vysokej úrovni. Dnes už štúdium bez notebooku a internetového pripojenia je nepredstaviteľné. Tieto vymoženosti ponúkajú mnoho neprebádaného priestoru pre tvorbu nových spôsobov vzdelávania. Jedným z týchto spôsobov je aj využívanie multimedialného obsahu.

Moja práca sa zameriava na spôsob podania tohto obsahu, konkrétne digitálnych prednášok študentom spolu so sprievodnými materiálmi. V práci riešim problémy ako ponúknuť študentovi tento obsah, aby bol využitý potenciál video prednášok čo najlepšie. Taktiež som navrhol spôsob prepojenia prednášok, ktoré spolu tematicky súvisia a pomáhajú užívateľovi plne pochopiť danú problematiku. Ďalej v práci riešim možnosť vytvárania si vlastných albumov, ktorých obsah si môže užívateľ navoliť podľa potreby a pracovať s nimi ako s reálnymi záložkami na svojej nástenke alebo tabuly. V práci som sa nezameral len na tvorbu užívateľského rozhrania, ale aj na jeho testovanie. Testovanie prebiehalo na cieľových užívateľoch. Na základe testov som vytvorený prototyp upravoval.

Celkovým cieľom práce je navrhnúť rozhranie, ktoré odstráni čo najviac negatívnych aspektov pri pozeraní digitálnych prednášok a vyzdvihne ich pozitíva. Do akej miery sa mi to podarilo a akým spôsobom sa môžete dočítať nižšie.

Kapitola 2

Teoretický základ

2.1 Digitálne prednášky

Digitálne prednášky sú nahrávky z prednášky, besedy, konferencie uchovávané v digitálnom formáte. Obsahujú mnoho informácií, ktoré sú cenným zdrojom pre každého študenta pri tvorbe projektov, učením sa pred skúškou alebo pre vlastnú potrebu. Video prednášky sú dnes dostupné už na mnohých univerzitách. Ich používanie je však často diskutabilné a nie všetci vyučujúci s nimi súhlasia. Problém je, že študenti nechodia na prednášky a vyučujúci nemajú komu prednášať. Aj keď podľa počtu stiahnutí je možné zistiť, koľko študentov prednášku videlo, vytráca sa čaro prednášať pred „živým“ publikom. Sledovanie prednášok zo záznamu má svoje výhody aj nevýhody.

Medzi hlavné výhody patrí lepšia organizácia času. Študent sa nemusí časovo prispôbovať. Mnohokrát sa prednášky prekrývajú s cvičeniami. Taktiež, ak študent z nejakého dôvodu vymešká prednášku, môže si zameškanú látku pozrieť neskôr. Prehrávanie zo záznamu prináša možnosti ako: pozastaviť, preskočiť, zrýchliť záznam či opätovne prehrať pri neporozumení prednášanej látky. Ďalšou výhodou je, že záznam je možné uchovať a po čase sa k nemu vrátiť.

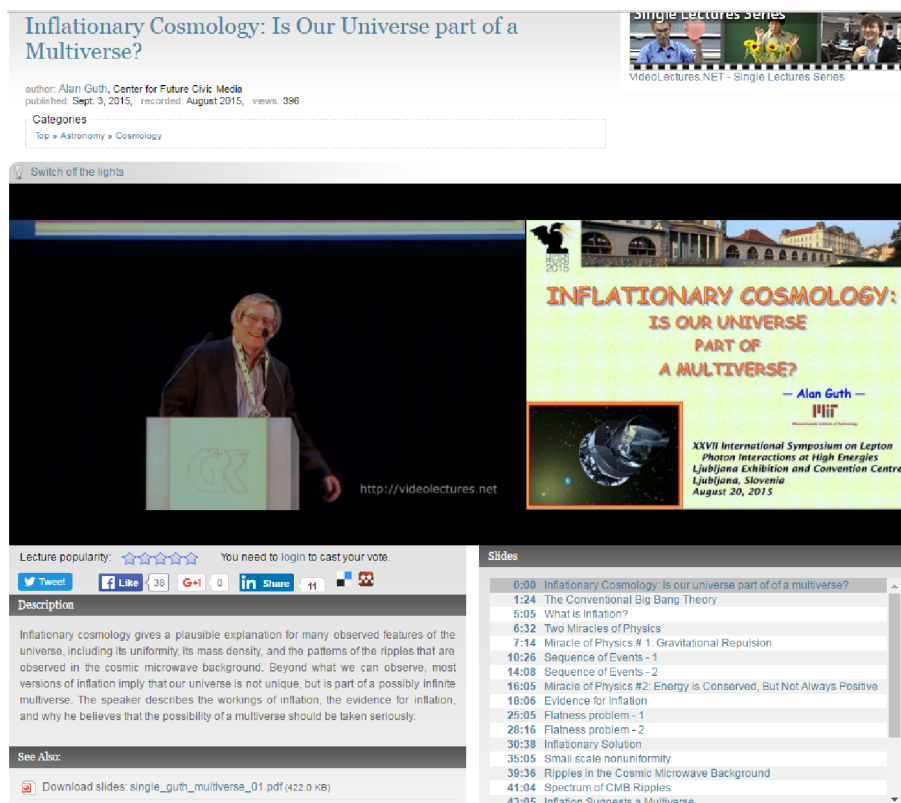
Avšak video prednášky nemajú len kladné stránky. Medzi nevýhody, ktoré prináša sledovanie zo záznamu patrí aj to, že študent nemá možnosť položiť dotaz. Videá je často potrebné sťahovať, čo môže zabráť určitý čas a miesto na disku. Nie je garantovaná ani kvalita videa alebo sa môže stať, že technika zlyhá. Pre sprístupnenie video prednášok je dôležité taktiež potrebné vybavenie, obsluha a údržba.

2.2 Existujúce a podobné riešenia

Pred samotným návrhom rozhrania som vyhladal existujúce alebo podobné riešenia, ktoré som analyzoval a čerpal z nich inšpiráciu. Prvé rozhranie, ktoré som analyzoval, je webová stránka na našej fakulte. Tu je možné si videá stiahnuť vo formáte MP3 alebo MP4. Sťahovanie videí má výhodu v tom, že po stiahnutí je možné si prednášku pozrieť aj neskôr bez internetového pripojenia. Medzi výhody patrí aj to, že videá sú utriedené podľa dátumu a predmetu. Ak študent vynechá prednášku, presne vie, kde si ju môže vyhľadať a stiahnuť. Hlavnou nevýhodou je to, že obsah prednášky sa dozvieme až po stiahnutí. Sťahovanie videí môže zabráť nejaký čas a miesto na disku. Ak študent hľadá danú informáciu v stiahnutej prednáške, trvá dlhšiu dobu pokiaľ sa k nej dostane. Toto rozhranie je dosť neefektívne a nevyužíva naplno potenciál, ktorý video prednášky majú.

Lynda.com¹ je komerčná stránka ponúkajúca užívateľom tutorialy. Videá sú zaradené v jednotlivých tutorialoch podľa špecifickej témy. Tutorial je zložený z jednotlivých videí, ktoré majú logickú nadväznosť. Samotné tutoriály sú taktiež zaradené do obširnejšej vednej oblasti. Tieto oblasti sa dajú filtrovať.

Videlectures² je Webová stránka ponúkajúca veľké množstvo prednášok. Veľkou výhodou je doprovodná prezentácia k videu. Ďalším prvkom, ktorý sa mi páči je že sú ponúknuté ďalšie videá, ktoré súvisia s aktuálne pozeraným videom. Tak isto ma zaujala aj možnosť vkladať komentáre k danému videu.



Obrázek 2.1: Screen z Videlectures.net.

Trello³ je komerčná webová stránka, ktorá nemá s videami nič spoločné. Slúži na organizáciu záložiek. Je to jedna veľká nástenka, kde je možné si pripieňovať poznámky, upomienky a pracovné záležitosti, ktoré už máme hotové, na ktorých ešte pracujeme a aj tie ktoré nás ešte len čakajú. Na tomto rozhraní sa mi páči, že si užívateľ svoje záznamy nechováva len podľa názvu, alebo čísla, ale aj samotným umiestnením na ploche. Pôsobí to fyzicky a užívateľ má oveľa lepšiu predstavu o svojich záložkách. Tento prvok som sa rozhodol zakomponovať do svojho návrhu.

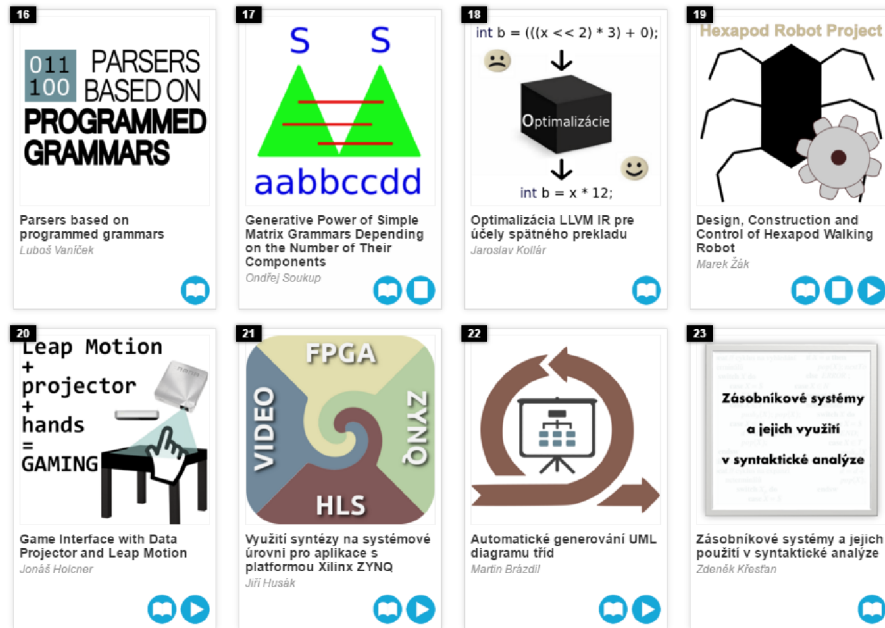
Zborník Excel-Fit⁴ je časť webovej stránky. Ponúka zoznam projektov prihlásených do školskej súťaže. Na tomto návrhu sa mi páči karta, ktorá reprezentuje celú prácu. Je v nej uvedený názov, autor a grafický obrázok naznačujúci tému projektu.

¹ www.lynda.com

² www.videlectures.net

³ www.trello.com

⁴ <http://excel.fit.vutbr.cz/sbornik/>



Obrázek 2.2: Zborník Excel Fit.

2.3 Uživatelské rozhrania a ich návrh

Rozhranie vo všeobecnosti môžeme chápať ako vrstvu medzi dvoma systémami, ktoré spolu navzájom interagujú a vymieňajú si informácie. Uživatelské rozhranie je druh rozhrania, kde na jednej jeho strane je užívateľ a na druhej systém. Je to spôsob akým môžu medzi sebou komunikovať. V informačných technológiách je to najčastejšie rozhranie medzi užívateľom a počítačom. Táto komunikácia je možná len určitými spôsobmi. Je obmedzená zmyslami človeka a technologickou vyspelosťou daného systému. Výmena informácií môže prebiehať oboma smermi. V súčasnosti sa využívajú len niektoré komunikačné kanály a sú to hlavne:

počítač → užívateľ

- Obraz (zrak)
- Zvuk (sluch)

užívateľ → počítač

- Pohyb (hmat)
- Zvuk (hlas)

V informačných technológiách je uživatelské rozhranie spolu s funkčnosťou a grafikou veľmi dôležitý aspekt pri tvorbe stránok, aplikácií, hier a podobne. Hlavným cieľom dobre navrhnutého rozhrania je predpokladať, čo potrebuje užívateľ so systémom robiť a pomôcť mu vykonať danú vec najrýchlejšie a najefektívnejšie.

2.3.1 Rozhranie príkazovej riadky

Je uživatelské rozhranie[5], v ktorom užívateľ komunikuje s programami pomocou príkazov, ktoré zadáva do príkazového riadku. Oproti textovému a grafickému uživatelskému rozhraniu, nepoužíva myš ani menu a nedokáže pracovať s celou plochou obrazovky (terminálu). Užívateľ zadáva príkazy napísaním ich názvu a stlačením enter. Interpret príkazového riadku daný príkaz prijme a na základe analýzy spustí príslušný program. Spustený príkaz

môže na terminál vypísať informácie o svojom behu, výsledok operácie, vyžiadať ďalší vstup a iné. Po ukončení daného príkazu je opäť možné zadávať ďalšie príkazy. Príkazový riadok je tradičným spôsobom ovládania vo všetkých unixových systémoch.

2.3.2 Textové užívateľské rozhranie

Textové užívateľské rozhranie (TUI – Text User Interface)[8] je užívateľské rozhranie, ktoré predstavuje medzistupeň medzi rozhraním príkazového riadku a grafickým užívateľským rozhraním. Pracuje v textovom režime, kde je obrazovka pevne rozdelená na stĺpce a riadky, pričom na každú pozíciu je možné zobraziť najviac jeden znak z vopred danej množiny. Pomocou špeciálnych znakov sú zostavené podobné ovládacie prvky, ako v GUI, takže prostredie obsahuje okná, tlačidlá, rolovacie zoznamy a podobne. Príkladom TUI sú programy, ktoré boli súčasťou operačného systému DOS.(Norton Commander, Turbo Pascal, edit.com, a iné)

2.3.3 Grafické užívateľské rozhranie

Grafické užívateľské rozhranie, GUI (graphical user interface)[2] je počítačové prostredie, ktoré uľahčuje interakciu užívateľa s počítačom tým, že reprezentuje programy, príkazy, zložky, a iné možnosti ako vizuálne elementy, ikony, menu, tlačidlá, okná a dialógové boxy. Výberom jednej z týchto grafických prvkov môže užívateľ iniciovať rôzne aktivity, ako je napríklad spustenie programu alebo tlač dokumentu. Pred zavedením GUI prostredia väčšina programov, ktoré komunikovali cez užívateľské rozhranie bola textovo orientovaná a vyžadovala, aby užívateľ poznal sadu zložitých príkazov, ktoré špecifikovali daný program. Prvé GUI bolo vyvinuté v roku 1970 firmou Xerox Corporation, ale populárne sa stalo až v roku 1980 so vznikom počítača Apple Macintosh. Dnes najznámejšie GUI rozhrania sú operačné systémy v počítačoch Apple Macintosh a Microsoft Windows.

Softwarové aplikácie často používajú balíky GUI zabudované do operačného systému, do ktorých pridávajú svoje vlastné prvky. Výhodou toho je, že poskytuje štandardné metódy pre splnenie daných úloh. Väčšina balíkov, ktoré sú založené na rovnakom operačnom systéme, sú postavené aj na rovnakých princípoch a ak je užívateľ oboznámený s GUI prvkami jedného, je ľahšie potom pracovať v inom balení.

GUI rozhranie zvyčajne ponúka viac ako jednu metódu pre začatie určitej akcie. Ak chce užívateľ napríklad vytlačiť dokument, môže zvoliť možnosť tlač z menu alebo ako alternatívu použiť klávesovú skratku. Užívateľ má možnosť si vybrať najpohodlnejší spôsob. GUI je zároveň aj nástrojom ľahkého sprístupnenia internetu jednotlivcom pomocou prehliadačov.

2.3.4 Základné princípy pri tvorbe užívateľského rozhrania

Pri tvorbe užívateľských rozhraní existuje mnoho pravidiel a princípov, ktoré by sa mali dodržiavať. Nasledujúcich osem pravidiel vychádza z [1]. Netreba však tieto pravidlá brať úplne doslova. Pri špecifických situáciách je možné pravidlá porušiť alebo upraviť.

- Konvencie. Pri návrhu rozhrania treba dávať pozor, aby sme neporušili základné konvencie a nezmenili všeobecne zaužívané rutiny. Predstavme si, že vytvoríme novú aplikáciu a tlačidlo zavrieť umiestnime v pravo dole. Užívateľia by boli zmätení a pri prepínaní s inými aplikáciami by boli nútení zakaždým premýšľať, kde sa nachádza tlačidlo k zavretiu okna. Dodržiavanie konvencií urýchľuje zorientovanie sa v novom

rozhraní, urýchljuje samotnú prácu a prispieva k celkovému dobrému dojmu z rozhrania.

- Široká škála užívateľov. Pred samotným návrhom je dobré si ujasniť, aká skupina užívateľov bude aplikáciu používať a na základe týchto poznatkov jej rozhranie patrične prispôbiť. Jedným z najdôležitejších faktorov je lokalizácia. Napríklad v Japonsku sú užívatelia zvyknutí na iné konvencie ako v Európe. Ďalšími významnými faktormi môžu byť vek, pohlavie, vzdelanie.
- Spätná väzba. Slúži k informovaniu užívateľa o vykonaných úkonoch. Netreba ju však preháňať. Člení sa na dva druhy a to: Silnú a slabú. Silná spätná väzba znamená, že užívateľ musí potvrdiť vzatie správy na vedomie alebo musí vybrať z ponúkaných možností. Slabá je vtedy, keď užívateľ nemusí nič potvrdzovať alebo vyberať.
- Navigácia užívateľa. Pri väčšom počte úkonov alebo zanorení sa v členení aplikácie je potreba udržiavať užívateľa informovaného v akej časti sa v daný moment nachádza. Pomáha mu udržiavať kontext.
- Predchádzanie chybám. Užívateľské rozhranie by malo minimalizovať možné chyby užívateľa. Ak dôjde k chybe je potreba užívateľa informovať a zároveň je dobré ukázať príčiny a naviesť na riešenia. Technické hlášky nie sú pre užívateľa veľmi veľavravné a môžu viesť k zmäteniu.
- Možnosť vrátiť sa späť. Zásadne prispieva k pohodlnému používaniu aplikácie. Užívateľ sa cíti oveľa pohodlnejšie a nebojí sa, že urobí chybu, pretože sa môže kedykoľvek vrátiť späť. Vníma rozhranie prívetivejšie keďže toleruje jeho chybu a umožňuje mu ju napraviť.
- Predvídateľné rozhranie. Užívateľ musí byť riadiacim prvkom rozhrania. Nemal by plniť rozkazy aplikácie, ale práve naopak. Nepredvídateľné rozhranie spôsobuje to, že užívateľ stráca nadvládu nad aplikáciou. Ak sa po vykonaní akcie stane úplne niečo iné ako očakával, užívateľ nevie či sa chyba stala na jeho strane alebo na strane systému.
- Nenútiť užívateľa si pamätať. Nesmie byť nútený si rozhranie pamätať. Platí pravidlo 7+- 2. To znamená že si človek krátkodobo zapamätá 5 až 9 údajov

2.3.5 Priama manipulácia, „Drag and Drop“ a „Look and Feel“

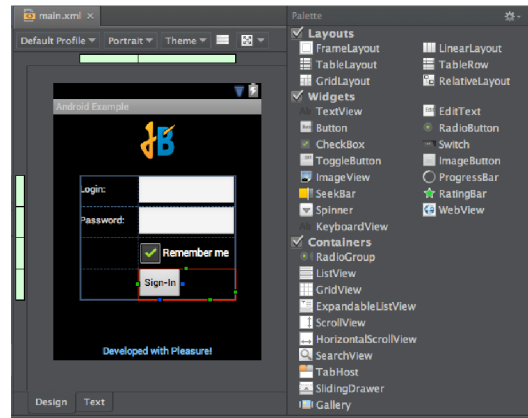
Pri návrhu rozhrania je potreba brať ohľad na to načo sú užívatelia zvyknutí a čo je im prirodzené. Predpokladom tomu je možnosť interakcie medzi užívateľom predmetmi zobrazenými na displeji počítača. Človek je z bežného života zvyknutý na priamu manipuláciu s rôznymi predmetmi pomocou úkonu ako uchopenie, jeho prenesenie z miesta na miesto, na používanie iných nástrojov, na vyhadzovanie nepotrebných predmetov do odpadkového koša, listovanie stránok v knižke a podobné činnosti. Prirodzené chovanie užívateľského rozhrania má dve kladné dôsledky. Prvým je, že ochota a schopnosť neškolených ľudí pracovať s počítačom vybaveným s prirodzeným chovaním. Tento dôsledok je veľmi dôležitý pre celkový dojem z práce na počítači. Druhým je znížená záťaž užívateľov pri práci s počítačom. Znížená záťaž j vysvetliteľná skúsenosťou užívateľa z bežného života a jeho podvedomými návykmi.

Jedným z prvých princípov tohto typu, ktorý bol zavedený do rozhrania je princíp „drag and drop“, ktorý dovoľoval manipuláciu s objektmi len pomocou myši. Neskôr bol tento

princíp zovšeobecnení vo forme „look and feel“, pod ktorým sa rozumie, že užívateľské rozhranie sa dá používať intuitívnym spôsobom. vychádzajúcim z podobnosti s prácou s predmetmi v reálnom živote.

2.3.6 Tvorba rozhrania grafickými editormi a zdrojovým kódom

V súčasnosti pri tvorbe užívateľských rozhraní je možné využiť tri spôsoby tvorby. Prvým je za pomocou zdrojového kódu. Ten definuje vzhľad rozhrania. Pri webových aplikáciách je to najčastejšie HTML a CSS. Výhodou je že vzhľad je možné nadefinovať ľubovoľne a tvorca nie je ničím limitovaný. Nevýhodou je však pomalá tvorba a to že celkový výsledok je vidieť až po naprogramovaní, takže modifikácia je komplikovanejšia. Druhou možnosťou je využiť grafické editory. Tu je možné si vzhľad nie naprogramovať, ale nakresliť. Výhodou je, že tvorca má priamo pred sebou podobu rozhrania, ktoré uvidí aj užívateľ. Vďaka editoru je možné rozhranie rýchlo modifikovať. Nevýhodou je zložitá definícia dynamického správania. Tretou, v súčasnosti veľmi využívanou možnosťou sú prostriedky integrujúce editory rozhraní a zdrojového kódu. Je to spojením prvých dvoch možností, pričom spája výhody a odstraňuje ich nedostatky. Príkladom sú programy Glade, Eclipse alebo JetBrains. Tvorba



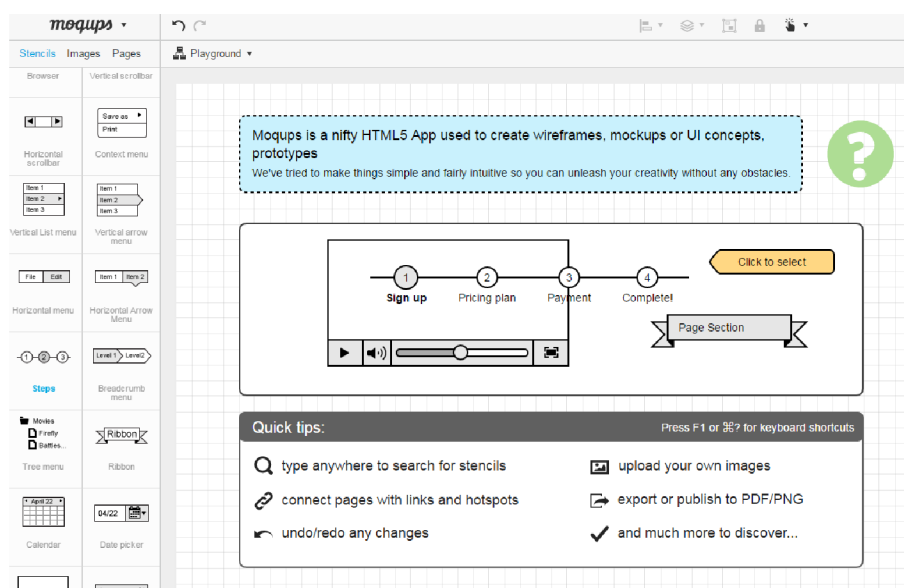
Obrázek 2.3: Príklad tvorby GUI v IntelliJ.

rozhrania takouto formou je veľmi efektívna. Má výhody v tom, že tvorca má priamo pred sebou podobu rozhrania a navyše môže upravovať aj zdrojový kód. Objekty v grafickom editore a zdrojový kód sú generované paralelne. To znamená, že po úprave v editore vidíme zmeny v zdrojovom kóde a naopak. Menšou nevýhodou je, že štruktúra programu je do určitej miery daná vývojovým prostredím.

2.3.7 Mockup

V preklade napodobenina. Vo všeobecnosti môžeme chápať ako prototyp cieľového produktu. Pri tvorbe webových stránok a aplikácií sa mockupy používajú na návrh užívateľského rozhrania. Ich hlavnou výhodou je, že máme ucelený pohľad na stránku, bez toho aby sme ju museli klasickým spôsobom vytvoriť. Po ich vytvorení sa rozhodujeme či začneme s vyvíjaním reálnej verzie alebo sa opäť pustíme do tvorby nového mockupu či modifikácie. Ich hlavné výhody sú: rýchla tvorba, ľahká modifikovateľnosť a cena. Mockupy je možné vytvárať mnohými spôsobmi a nekladú žiadne obmedzenia. Základný návrh môže prebiehať na aj na papieri či tabuly. Ak však chceme získať reálnejší pohľad, je dobré využiť niektorý

z voľných či platených produktov akým je napríklad: mockups.com . Tu si môžeme poskladať rozhranie z vopred vytvorených komponentov. Na mnohých stránkach je možné nájsť predlohy mockupov. Za tie lepšie je nutné si priplatiť.



Obrázek 2.4: Príklad tvorby mockupu pomocou mockups.com.

2.4 Testovanie užívateľského rozhrania

Testovanie užívateľského rozhrania je technika testovania, používaná na identifikáciu prítomnosti chýb v rozhraní daného produktu/softwareu. Patrí medzi dôležité časti vývoja každého softwarového produktu. Úlohou testov je odhaliť chyby, anomálie, zlyhania a zistiť ako vplýva rozhranie na užívateľa. Pri včasnom odhalení chýb môžeme ušetriť mnoho času a financií. Existuje mnoho metód ako otestovať užívateľské rozhranie. Testovanie sa delí na dve základné skupiny.

2.4.1 Testovanie funkčnosti

Tvoria ho testy, kde je predmetom skúmania funkčnosť rozhrania. Hlavným cieľom je odhaliť chyby, ktoré sú zapríčinené chybou programátora, nekonzistenciou údajov, zlým návrhom štruktúry a iné. Testy môžeme na základe úrovne rozdeliť na tri skupiny:

- Testy modulov: Sú vykonávané programátormi individuálneho modulu, pred tým ako je integrovaný do systému.
- Integrované testy: Vykonáva ich skupina, ktorá moduly do systému integruje
- Množina QA testov :Vykonáva ich skupina sledujúca kvalitu kompletne integrovaného systému

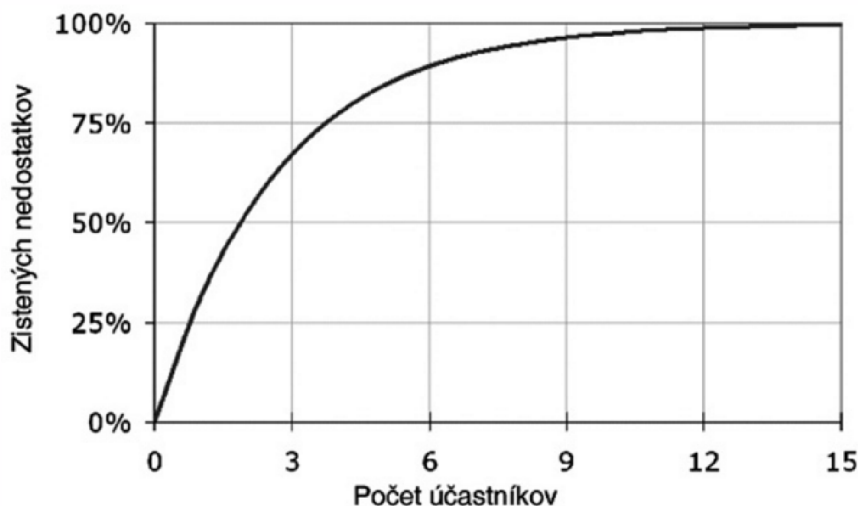
Tieto testy môžu byť vykonávané samotnými programátormi alebo testerami. Preskúvanie všetkých možností a stavov rozhrania môže byť niekedy, hlavne pri veľkých projektoch veľmi zdĺhavé. Preto boli vyvinuté prostriedky na automatické testovanie. Na trhu existuje

niekoľko dostupných nástrojov, ktoré sa líšia svojou podporou programovacích jazykov, množinou možností aj obmedzení. Príkladmi nástrojov sú: Selenium, WebTest, Twill a iné.

2.4.2 Testovanie na užívateľoch

V angličtine Usability testing. Patrí medzi najdôležitejšie časti pri vývoji webových aplikácií. Často krát rozhoduje o úspechu a neúspechu aplikácie. Dá sa zisťovať viacerými metódami a postupmi. Cielom je pochopiť správanie používateľov a nájsť problémy produktu, ktoré je možné eliminovať. Výsledkom tejto snahy je zvýšenie spokojnosti užívateľa. Testovanie je optimálne vykonávať ešte vtedy, pokiaľ sú úpravy možné a nestoja toľko ako v neskoršej fáze. Pri testovaní je dôležitý správny návrh testov a výber testovanej skupiny ľudí. Optimálna skupina ľudí je skupina, ktorá aktuálne používa podobnú technológiu, alebo vzorka z potencionálnych cieľových užívateľov. Mali by byť nezainteresovaní, najlepšie takí, ktorí nemajú nič spoločné z autormi. Je potrebné rozlišovať kvalitatívne a kvantitatívne testovanie[3].

Rozhodnutie o spôsobe testovania má vplyv na celkové náklady aj náročnosť. Pri kvalitatívnom testovaní, testy absolvuje menej účastníkov (max 5-8). Týmto dosiahneme, že za krátky čas odhalíme až 90% problémov. Kvantitatívne testovanie si vyžaduje viac ako 20 účastníkov. Využíva sa pri veľkých projektoch, kde je potrebné podrobné štatistické vyhodnotenie a využitie metrík.



Obrázek 2.5: Vzťah medzi počtom účastníkov a percentom zistených problémov použiteľnosti[4].

Pred samotným testovaním je dobré účastníkov oboznámiť s tým, že nie sú testovaní oni, ale produkt. Znižuje to nervozitu a tým zvyšujeme presnejší výsledok testu. Mnoho odborníkov ako Jakob Nielse alebo Kuniavsky, sú zástancami nechať používateľov myslieť nahlas počas vykonávania testov. Spätná väzba od užívateľov môže byť podaná formou dotazníkov, formulárov, rozhovoru, video nahrávky a pod. Dôležitým krokom je analýza týchto informácií. Každé rozhranie má svoje špecifiká a zameriava sa na iné parametre. Napríklad pri e-shope je dôležité sledovať, kedy sa užívateľ rozhodne vložiť alebo odstrániť produkt zo svojho virtuálneho košíka. Taktiež je tu napríklad dôležité sledovať, prečo si užívateľ vložil do svojho košíka tovar ale rozhodol sa nakoniec, že ho nekúpi. Nemohlo to byť preto že priebeh platby bol komplikovaný? Alebo stratil prehľad pri platbe a zo strachu

o stratu svojich peňazí zavrel prehliadač a vypol počítač ?Možno áno a možno nie. A presne na tieto otázky je možné nájsť odpoveď v spätnej väzbe od užívateľov.

2.5 Implementačné nástroje

Na trhu je veľké množstvo technológií, ktoré umožňujú a uľahčujú tvorbu webových aplikácií. Vybrať však najvhodnejší nástroj nie je vždy také jednoduché, vzhľadom na odlišné možnosti, ktoré ponúkajú. Presvedčil som sa o tom aj ja sám pri implementácii tejto práce. V nasledujúcej časti v skratke predstavím niektoré technológie s ktorými som pracoval alebo uvažoval o použití pri implementácii webovej aplikácie.

2.5.1 HTML

HTML(Hypertext Markup Language) Slúži na tvorbu hypertextových dokumentov v sieti WWW. V súčasnosti sa používa hlavne na definíciu obsahu štruktúry jednotlivých dokumentov. Dokumenty v jazyku HTML[7] sú prosté textové súbory označené príponou .html alebo .htm. Obsah tohto dokumentu tvoria: vlastný text dokumentu, vložené značky definované špecifikáciou HTML, znakové entity. Použité značky, ich možné atribúty a znakové entity sú definované príslušným DTD a ich množina je pevne daná. To znamená, že tvorca dokumentu nemôže vytvárať nové značky ani entity. Tým, že dokument HTML je tvorený elementmi, ktoré môžu byť vzájomne vnorené, pričom existuje jeden koreňový element <html>, môžeme celý dokument popísať ako strom elementov. Elementy sa delia na dva typy: riadkové a blokové. Dodnes nebol plne nahradený žiadnou novou technológiou. Neustále však prebieha vylepšovanie jazyka, kde sú pridávané nové rysy, rešpektujúce vývoj a naopak, tie ktoré sú nedostatočné sú postupne odstraňované.

2.5.2 CSS

Skratka znamená Cascading Style Sheet[6]. Slúži na vizuálne formátovanie internetových dokumentov. Hlavnou výhodou je možnosť oddeliť vzhľad od štruktúry stránky. Vďaka tomu je možné dosiahnuť väčšiu prehľadnosť a jednoduchosť kódu. Ďalšou výhodou je, že definícia vzhľadu môže byť zdieľaná viacerými HTML dokumentmi. Špecifikácia CSS definuje obsiahlu množinu vlastností, ktoré HTML môže mať. Niektoré vlastnosti sa týkajú všetkých elementov, iné platia pre vybranú množinu elementov. Problémom môže byť, že nezaručuje rovnaké vykresľovanie vo všetkých prehliadačoch. Základnou jednotkou definície štýlu dokumentu je štýlový predpis (style sheet). Ide o textový súbor obsahujúci postupnosť pravidiel určujúcich vzhľad jednotlivých HTML elementov. Každé pravidlo sa skladá z dvoch častí: zo selektoru a definície pravidiel určujúcich vzhľad elementov. Selektor určuje množinu elementov, na ktorú sa pravidlo aplikuje. Existujú tri spôsoby ako je možné pripojiť štýlový predpis k dokumentu HTML:

- Vložíme štýlový predpis priamo do hlavičky HTML dokumentu pomocou špeciálnych znakov
- Umiestnime štýlový predpis do zvláštného súboru a do hlavičky HTML dokumentu umiestnime odkaz na tento súbor
- Špecifikujeme štýl priamo pre konkrétny element v tele dokumentu pomocou zvláštného atribútu

2.5.3 PHP

PHP(Personal Home Page)⁵ je open source skriptovací jazyk, ktorý sa používa najmä na vývoj klient-server aplikácií a dynamických webových stránok. Pracuje na strane servera. PHP bolo inšpirované jazykmi podporujúcimi procedurálne programovanie. V neskorších verziách objektovo orientovaný. Hlavnou výhodou je, že dokáže spolupracovať s relačnými databázami ako MySQL, Oracle, IBM DB2 a iné. Nevýhodou PHP je špecializovaný prístup k jednotlivým druhom SRBD. Znamená to, že ak chceme preniesť projekt napríklad z MySQL na ORACLE, musíme prepísať skripty. Všetko PHP kód je vykonávaný pomocou PHP runtime, aby dynamicky tvoril obsah na webovej stránke.

2.5.4 Javascript

Hlavným vývojárom je Brendan Eich zo spoločnosti Netscape Communication pod názvom Mocha. Neskôr bol premenovaný na LiveScript. Pred uvedením na trh bol premenovaný na JavaScript. Skriptovací programovací jazyk, radí sa medzi interpretované jazyky, čo znamená že program je vykonávaný na klientskej strane, teda obvykle vo webovom prehliadači, bez nutnosti prekladu do strojového jazyka. Je Case sensitive a objektovo orientovaný. Syntakticky vychádza z jazykov Java a C/C++. Jeho úloha v aplikáciách je väčšinou doplnková. Najčastejšie sa používa na zvýšenie interaktivity webovej stránky. Umožňuje reagovať na užívateľské akcie a na ich základe meniť vzhľad alebo obsah bez toho aby bola nutná komunikácia so serverom. Pracuje s elementmi DOM (Document Object Model) a dovoľuje ich klientovi aktualizovať a modifikovať. Neumožňuje pracovať so súborami, z dôvodu bezpečnosti

⁵<https://secure.php.net/>

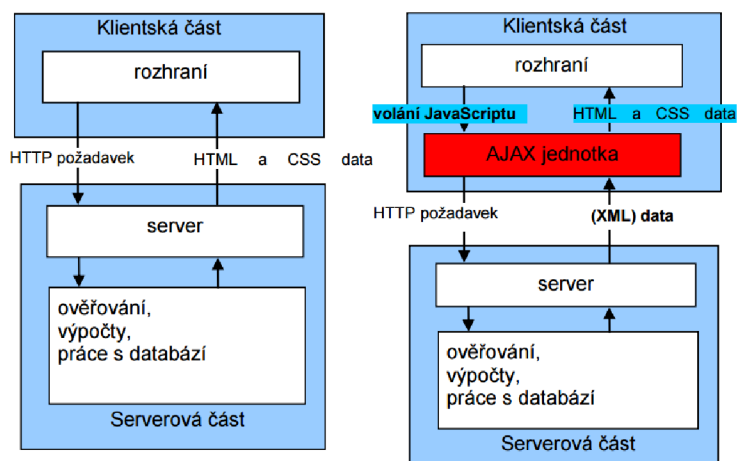
2.5.5 Ajax

Je to skratka slov Asynchronous Javascript And XML⁶

Je zlúčením technológií :

- HTML, XHTML, CSS
- DOM model
- XML pre výmenu dát
- Objekt XMLHttpRequest pre asynchrónnu komunikáciu zo serverom
- Javascript

Rieši problém, ktorý vzniká keď je potreba zmeniť len časť užívateľského rozhrania a bez toho, aby sa nenarušila súbežná práca užívateľa s rozhraním. Ajax nežiada o celú novú internetovú stránku, ale len vytvorí dátový kanál medzi klientom a serverom, cez ktorý server pošle potrebné dáta. Pri tomto využíva Ajax XMLHttpRequest. Najčastejšie formát pre predávanie dát je XML. Je dôležité vedieť, kedy použiť túto technológiu. Niekedy stačí použiť JavaScript na zmenu DOM.



Obrázek 2.6: Rozdiely modelu klasickej komunikácie (vľavo) s modelom využívajúci prístup AJAXu.

2.5.6 Nette

Nette⁷ je open source framework pre tvorbu webových aplikácií v PHP 5. Je to ucelený súbor do seba zapadajúcich knižníc. Staví sa na MVC (model view controller) architektúre. Zameriava sa na elimináciu bezpečnostných rizík. Podporuje AJAX, DRY, KISS, MVC a znovupoužitelnosť kódu. Využíva udalosťami riadené programovanie a z veľkej časti je založený na použití komponentov.

⁶<http://api.jquery.com/jquery.ajax/>

⁷<https://nette.org/>

2.5.7 Bootstrap

Bootstrap⁸, povodne Twitter Bootstrap bol vyvinutý vo firme Twitter ako framework pre dosiahnutie konzistencie medzi internými aplikáciami. Je open-source, front-end knižnica určená pre tvorbu web stránok a webových aplikácií. Pozostáva z HTML, CSS a ponúka JavaScript rozšírenia. Je kompatibilná z najnovšími verziami prehliadačov Google Chrome, Internet Explorer, Opera, Safari. Od verzie 2.0 podporuje responzívny design. Medzi hlavné výhody patrí: Úspora času, responzívny a konzistentný design, jednoduché použitie, kompatibilita s prehliadačmi, je voľný.

2.5.8 SQL

Structured query language. V preklade štruktúrovaný vyhľadávací jazyk. Je určený na manipuláciu DML(výber, vkladanie, úprava a mazanie) a definíciu údajov (DDL). V súčasnosti je najpoužívanejší jazyk, tohto druhu v relačných systémoch riadenia báz dát. Medzi základné syntaktické konštrukcie jazyka patrí:

- Manipulácia dát: SELECT,INSERT,DELETE,UPDATE
- Definícia dát: CREATE, ALTER, DROP
- Riadenie dát: GRANT, REVOKE

2.5.9 phpMyAdmin

Je voľný softwarový nástroj napísaný v PHP, určený k ovládaniu MySQL prostredníctvom webového prehliadača. phpMyAdmin podporuje širokú škálu operácií nad MySQL a MariaDB. Medzi najčastejšie úkony patria vytváranie tabuliek, definovanie vzťahov, indexov, práv a podobne. Tieto operácie môžu byť vykonávané cez grafické užívateľské rozhranie alebo pomocou SQL príkazov.

2.5.10 Strih videí

Na trhu je dnes mnoho nástrojov určených na úpravu a strih videí. Problémom je, že väčšina je platená. Tie ktoré sú zadarmo neponúkajú taú širokú škálu možností. Pre môj účel stačil aj základný program ako je Avidemux.

Avidemux⁹ je voľný program, navrhnutý na úpravu videí. Je ho možné používať na toperačných systémoch : Windows, Linux, BSD, Mac OS X. Základné funkcie sú strihanie videa, používanie filtrov a kódovanie do potrebných formátov. Podporované formáty sú: AVI, DVD, MPEG,MP4, ASF.

2.5.11 Vývojové prostredie

Anglicky veľmi často používaná skratka IDE (Integrated development environment). Je software, ktorý uľahčuje prácu programátorom. Dnes už mnohé IDE majú podporu pre viac programovacích jazykov. Základom toho software je editor kódu. Ten zvyrazňuje syntax daného programovacieho jazyka, tak aby bol kód čo najviac čitateľnejší a prehľadný. Ďalej mnoho vývojových prostredí ponúka automatické dopĺňanie príkazov podľa programovacieho jazyka. Ďalej môže obsahovať kompilátor, prípadne interpret a debugger. Niektoré

⁸<http://getbootstrap.com/>

⁹<http://fixounet.free.fr/avidemux/>

obsahujú systém pre rýchly vývoj aplikácií (RAD), ktorý slúži pre vizuálny návrh grafického užívateľského rozhrania. Ak sa jedná o objektovo orientované programovanie, môže obsahovať „object broeser“.

Kapitola 3

Analýza cieľového užívateľa

Určiť cieľovú skupinu rozhrania pres správu digitálnych prednášok bolo v tomto prípade celkom jednoznačné. Sú ňou predovšetkým študenti. Keďže som študentom aj ja a využívam možnosť digitálnych prednášok, bolo pre mňa ľahšie zosumarizovať všetky potrebné požiadavky dôležité pre toto rozhranie.

Študentov, ktorí budú využívať túto webovú aplikáciu, môžem považovať za technicky zdatných, preto v prípade potreby môže byť rozhranie komplexnejšie. Pri návrhu som zohľadnil aj to ako sú užívatelia zvyknutí sa orientovať v digitálnych prednáškach a aké prvky ponechať. Pri skúmaní užívateľov som sa najprv zamyslel nad tým, aká situácia bude viesť študenta k tomu, aby využil služby mojej webovej stránky. Našiel som tieto tri situácie.

- situácia 1: Študent vynechal prednášku a chce si ju pozrieť z domu. V tomto prípade je dôležité pre užívateľa to, aby si vedel vyhľadať prednášku podľa presného dátumu alebo názvu témy a mohol si pozrieť všetko čo zameškal.
- situácia 2: Študent má pred semestrálnou písomkou alebo štátnicami a chce si vyhľadať len tie témy, ktoré budú na teste. V tomto prípade je dôležité brať ohľad na to, že užívateľ bude chcieť vyhľadávať podľa názvu alebo konkrétneho príkladu. Cieľom je rýchlo vyhľadať danú tému a za čo možno najkratší čas pochopiť danú problematiku.
- situácia 3: Študent je aktívny a chcel by sa vo svojom voľnom čase zdokonaľiť v oblasti, ktorá ho zaujíma. Je dôležité si uvedomiť, že pri takomto type študenta je potrebné ponúknuť čo najväčšie množstvo videí a informácií. Užitočné je, ak aplikácia umožňuje užívateľovi kontinuálne prechádzať jednotlivé video prednášky, ktoré na seba tematicky nadväzujú. Toto motivuje užívateľa k aktívnejšiemu štúdiu.

Tieto tri prípady boli základným kameňom pri návrhu finálnej štruktúry a užívateľského rozhrania aplikácie. Ďalej som zaznamenal konkrétne situácie, ktoré nastávajú, alebo môžu nastať.

- študenti pri pozeraní prednášky využívajú prezentácie alebo oporu.
- študent si často robí poznámky.
- študenta častokrát pri pozeraní niečo vyruší a rád by si presunul prednášku na neskôr.
- študent si vyberie a označí prednášky, ktoré ho zaujali a ktoré si chce pozrieť neskôr.
- študenta prednáška nudí a zaspí pri nej.

- študent si chce pozrieť prednášku offline.
- študent chce odporučiť kamarátom konkrétne prednášky a časti v nich, ktoré sú podstatné na pol semestrálnu písomku, skúšku či na vyriešenie projektu

Tieto poznatky o správaní užívateľov a ich problémy sú dôležité a preto som si ich riešenia stanovil ako hlavný cieľ pri návrhu. Cieľom každej webovej stránky či aplikácie by malo byť riešenie nejakého spoločenského problému.

Túto analýzu cieľového užívateľa som nevykonal hneď na začiatku, čo bola veľká chyba. Vykonával som ju až po niekoľkých neúspešných návrhoch. Po tomto kroku sa mi pracovalo oveľa lepšie a mal som jasný cieľ a predstavu ako má aplikácia vyzerať.

Kapitola 4

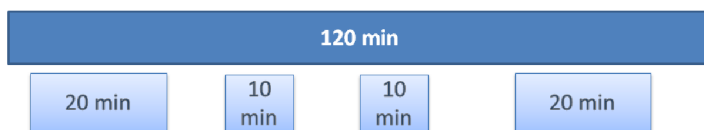
Návrh riešenia

Cieľom tejto kapitoly je navrhnúť riešenia na problémy a nedostatky, ktoré som zozbieral v časti „Analýza cieľového užívateľa“. V tejto kapitole bližšie predstavím celkovú štruktúru stránky a jej jednotlivé časti.

4.1 Forma digitálnych prednášok a vzťahy medzi nimi

Forma video prednášok je veľmi dôležitým rozhodnutím a bude určovať celkovú logiku a zmysel aplikácie. Video prednášky môžu mať dve formy. Prvou z nich je, že prednášku ponechám celú. Väčšinou je to dvoj- až štvrt hodinové video na jednu alebo viac obsiahnejších tém z daného predmetu v konkrétny deň. Toto riešenie má výhodu v tom, že študent si môže presne vyhľadať prednášku, na ktorej nebol a pozrieť si ju celú. Nevýhodou tohto spôsobu je, že pri popise videa by musel byť dlhý zoznam kľúčových slov, aby si užívateľ vedel vyhľadať či sa v danej prednáške nachádza potrebná časť. Ak ma študent záujem si pozrieť len časť prednášky, ktorá sa zaoberá jeho problémom, musí prejsť celú prednášku, aby sa dostal k požadovanej časti.

Druhou možnosťou je rozdelenie prednášky na viacero kratších videí, ktoré zaznamenávajú vysvetlenie len jedného konkrétneho problému. Toto riešenie má nevýhodu v tom, že vkladanie videí bude komplikovanejšie. Výhoda sa nám naskytuje logicky v tom, že si užívateľ vyhladá priamo video, ktoré potrebuje a nemusí preskakovať obsah, ktorý ho nezaujíma.

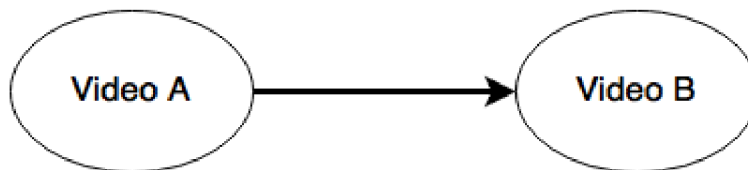


Obrázek 4.1: Rozdelenie prednášky na menšie časti.

Rozhodol som sa, že pre moju webovú stránku použijem druhú možnosť. Dôvodom môjho rozhodnutia je fakt, že prvé riešenie je štandardné, zaužívané a neponúka nič nové. Druhý spôsob je oveľa zaujímavejší a ponúka väčšie množstvo možností ako štrukturovať a prepájať jednotlivé videá. Videá sa budú môcť prehrávať priamo cez webové rozhranie. Nie je potreba ich sťahovať.

Jednotlivé videá som sa rozhodol usporiadať do sieťovej štruktúry. Tvoria jednu veľkú sieť. Vzťahy medzi videami sú orientované, čo znamená, že dané video sa odkazuje na ďalšie

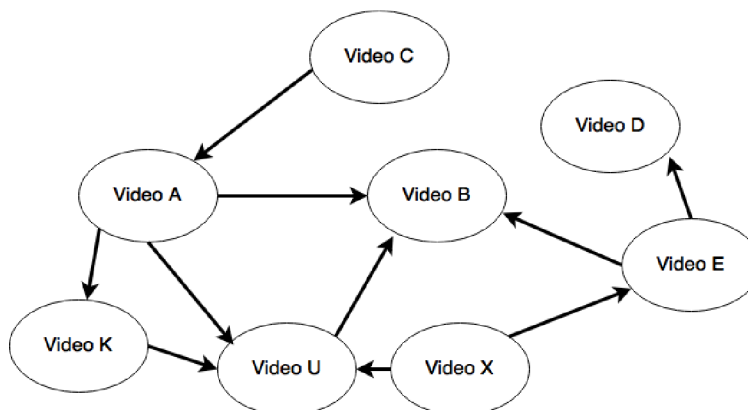
videá.



Obrázek 4.2: Vzťah prerekvizity a videa.

Tento vzťah značí dve veci

- Prvá vec: video A je prerekvizitou videa B. To znamená, že pred pozretím videa je dobré mať znalosť z videa A.
- Druhá vec: video A sa odkazuje na video B. To znamená, že po pozretí videa A sa naskytuje možnosť pozrieť video B, ktoré rozširuje znalosť problematiky vo videu A.



Obrázek 4.3: Štruktúra videí.

4.2 Návrh funkcionality aplikácie

Samotnému návrhu užívateľského rozhrania predchádzal návrh štruktúry a funkcionality. Ten definuje, čo bude v aplikácii možné vykonávať a čo nie. Pri tomto návrhu už je potreba mať v hlave predstavu o užívateľskom rozhraní. Pri návrhu funkcionality som sa riadil analýzou cieľového užívateľa. V tejto časti riešim problémy, ktoré sa užívateľovi pri prehliadaní prednášok môžu alebo nemusia naskytnúť. Pre túto aplikáciu som sa rozhodol implementovať nasledujúcu funkcionality:

- **Vyhľadávanie videí** Je základným a dôležitým prvkom. Pomocou vyhľadávania užívateľa prístupujú k daným videám. Má veľký vplyv na rýchlosť vyhľadania potrebnej informácie. Vyhľadávanie funguje na základe kľúčových slov, ktoré sa vzťahujú k danému videu.
- **Prechádzanie videí** Keďže som sa rozhodol videá usporiadať do sieťovej štruktúry, je potrebné mať možnosť touto štruktúrou prechádzať. To znamená, že pri danom

videu je možnosť prejsť na prerekvizitu alebo odporučené video. Táto možnosť ponúka užívateľom kvalitnejšie a širšie vzdelávanie. Tým, že sa študentovi zobrazia pred daným videom prerekvizity môže veľmi rýchlo uvážiť či má dostačujúce vedomosti pre dané video. Ak nie, tak jednoducho zmení prerekvizitu na hlavné video a môže začať študovať. Podobný princíp platí pre doporučené videá. Ak študent dopozera dané video a má záujem rozšíriť svoje nadobudnuté vedomosti, vyberie si niektoré z doporučených videí a môže pokračovať. Výhodou tohto je, že sa študent učí postupne.

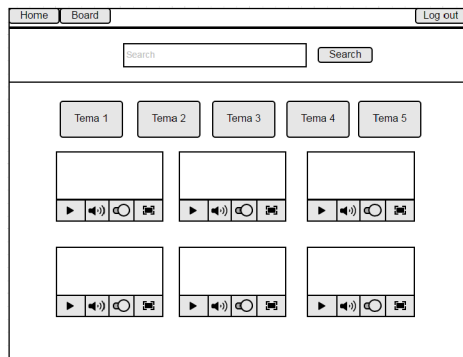
- **Triedenie videí podľa kategórií.** Služi pre užívateľov, ktorých zaujíma len určitá oblasť a chcú sa v nej zdokonaľovať.
- **Prehrávanie videa.** Možnosť prehrať video cez webový prehliadač je dôležitou funkčnosťou. Video nie je potreba sťahovať, šetrí pamäť a je k dispozícii okamžite. Je tu však nevýhoda, že užívateľ musí mať prístup k internetovej sieti.
- **Sťahovanie videa.** Sťahovanie videí je užitočné vtedy keď si chce užívateľ pozrieť video neskôr a vie že bude offline. Napríklad vo vlaku, v parku a podobne.
- **Doprovodné materiály.** Tie rozširujú informácie uvedené v prednáške. K danému videu je možné pripojiť ľubovoľný súbor. Najčastejšie sú to samotné prednášky v pdf formáte. Ak sa jedná o cvičenie je možnosť pripojiť obrázky či výpočty.
- **Poznámky.** Táto funkčnosť služi k tomu, aby si užívateľ mohol priebežne poznamenávať informácie o prednáške. Môže si tu poznačiť kde v prednáške skončil, časy dôležitých častí, iné poznámky a podobne. Týmto odstránime nutnosť písať si poznámky na papier. Nenahradzuje to však počítanie príkladov na papier alebo robenie si obsiahnejších poznámok. Výhodou je že sa poznámky nestratia.
- **Like.** Možnosť dať „Like“. Služi čiastočne ako spätná väzba pre autorov prednášok a čiastočne pre užívateľov, aby vedeli odhadnúť kvalitu prednášky.
- **Seen.** Možnosť označiť za „videné“. Služi na to, aby si študent mohol označiť videá, ktoré už videl.
- **Nástenka.** Ide o plochu, kde je možné si vytvárať boxy a do nich vkladať prednášky a fyzicky si ich presúvať. Zámerom je, aby si študent mohol vytvárať svoje štruktúry z prednášok pri príprave na skúšky a podobne. Možné využitie je aj pri videách, ktoré si chce študent pozrieť neskôr.

4.3 Uživatelské rozhranie

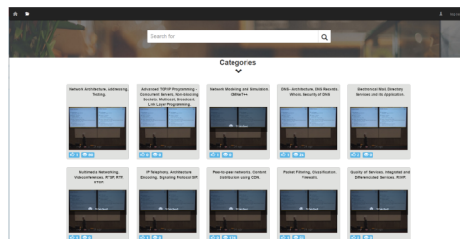
Uživatelské rozhranie definuje akým spôsobom bude možné vykonať úkony a pristupovať k informáciám. Nadväzuje na časť - analýza cieľového užívateľa 3 a návrh funkcionality 4.2. V tejto časti vytváram rozhranie medzi navrhnutou funkcionalitou a užívateľom. Užívateľovi sa pri práci budú naskytovať štyri možné úrovne stránky. Každá z nich má svoj osobitný účel a poskytuje iné možnosti. Tvoria základ a sú kostrou celej stránky.

4.3.1 Úvodná stránka

Po prihlásení sa užívateľovi naskytne náhľad na úvodnú stránku. Na úvodnej stránke sa nachádza viacero komponentov.



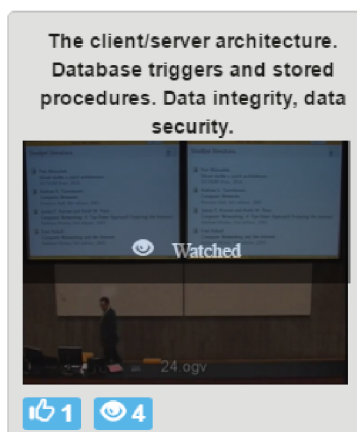
(a) Návrh.



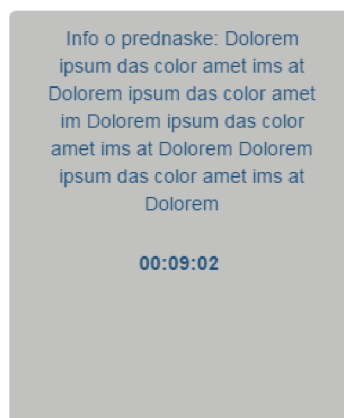
(b) Realizácia.

Obrázek 4.4: Úvodná stránka

Karta s videom. Jednotlivé prednášky sú reprezentované kartou. Na tejto karte je potreba uchovať toľko informácií, aby stačili užívateľovi na pochopenie o čom prednáška je a čo sa po jej pozretí dozvie. Rozhodol som sa, že na karte budú tieto informácie: názov videa; vystihujúci obrázok alebo snímok z videa; počet videní a like-ov; obsah prednášky a dĺžka videa. To je však mnoho informácií na to, aby sa proporciálne zmestili na jednu kartu. Preto som navrhol riešenie. Je založené na tom, že sa karta automaticky otočí ak po nej prejdeme kurzorom. Tým sa voľná plocha karty zväčší a tak nie je problém na ňu umiestniť všetok vyššie uvedený obsah. Na karte je informácia o tom, či užívateľ dané video už videl. Ak áno, zistí to jediným pohľadom takže nemusí nad kartou rozmýšľať a môže zamerať svoju pozornosť na ďalšie videá. Počet like-ov užívateľovi môže pomôcť pri výbere videa. Ak si všimne, že video má veľa like-ov, nepriamo ho bude lákať si dané video pozrieť. Čas na zadnej strane karty slúži k tomu, aby získal užívateľ prehľad o dĺžke videa. Obrázok na karte zachytáva snímok z videa.



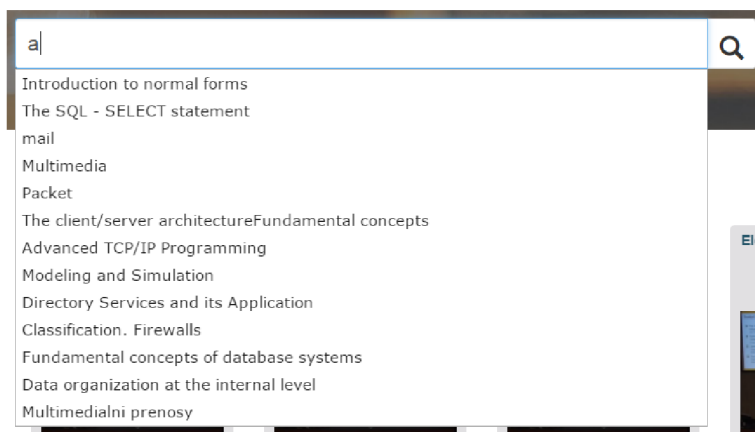
(a) Predná strana.



(b) Zadná strana.

Obrázek 4.5: Karta videa.

Vyhľadávanie. Ďalším dôležitým prvkom na úvodnej stránke je vyhľadávací panel. Tu je možné zadať výraz, ktorý chceme hľadať. Vyhľadávanie automaticky ponúka možnosti v momente ako začne užívateľ písať. Uľahčuje užívateľovi vyhľadávanie a šetrí čas tak, že nemusí dopísať hľadaný výraz, len šípkami (hore a dole) alebo kurzorom prejde na výraz o ktorý sa zaujíma. Panel je situovaný v strednej hornej časti plochy. Je pomerne veľký, aby užívateľovi bol stále na očiach a tým poskytol možnosť rýchleho vyhľadávania. Predstavuje hlavný komponent úvodnej stránky a umožňuje užívateľovi prístup ku konkrétnym videám.



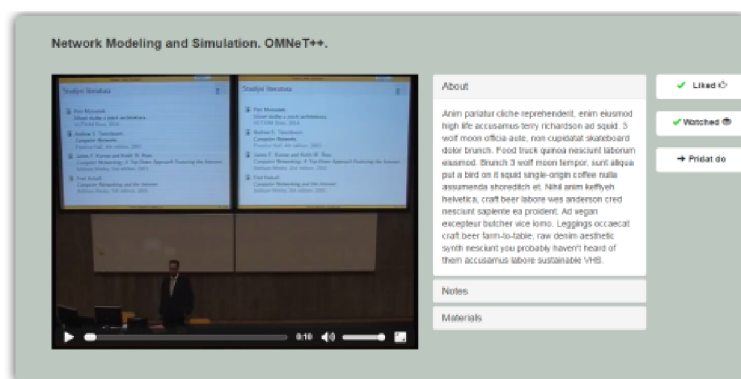
Obrázek 4.6: Vyhľadávací panel.

Kategórie. Nad zoznamom videí je tlačidlo s kategóriami. Po jeho rozkliknutí sa zobrazia jednotlivé kategórie a užívateľ si má možnosť vybrať, ktoré kategórie si chce vyfiltrovať. Uľahčuje to vyhľadávanie pri pasívnom prehlíadaní prednášok, v prípade ak sa užívateľ zaujíma len o konkrétnu tému.

4.3.2 Stránka s videom

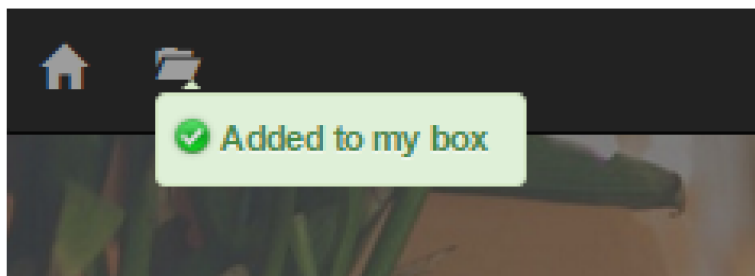
Stránka obsahuje vyhľadávací panel, kartu s hlavným videom, panel s prerekvizitami a doporučenými videami.

Karta s videom. Je situovaná v centre stránky, obsahuje všetky informácie a doprovdné materiály týkajúce sa danej prednášky.



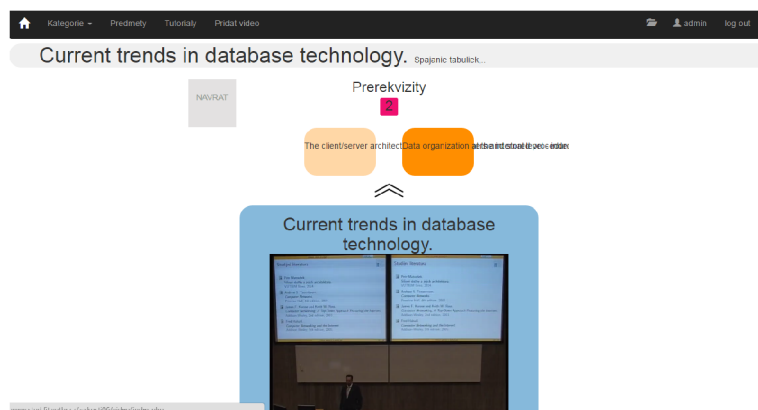
Obrázek 4.7: Hlavná karta s videom.

Hlavnú časť tvorí samotné video, ktoré je najdôležitejšie. Napravo sú doprovodné materiály a informácie. Prednáška obsahuje mnoho informácií a preto som si vytvoril menu s políčkami medzi ktorými je možné dynamicky prepínať. Prvé políčko obsahuje základné informácie o videu. Tu sa môže užívateľ dočítať o čom video je a čo sa po pozretí dozvie. Ďalej obsahuje políčko s možnosťou ukladania si vlastných poznámok. Poznámky sú umiestnené vedľa videa, takže je možné si počas pozerania priebežne robiť poznámky, bez nutnosti prerušiť sledovanie. Design karty napodobňuje reálny poznámkový blok. V menu sú aj doprovodné materiály slúžiace k rozšíreniu informácií o videu. Užívateľ nemusí prehadzovať školský informačný systém, všetko má po ruke a na jednom mieste. Posledným prvkom v menu je možnosť si dané video stiahnuť, v prípade, že by si ho chcel užívateľ pozrieť neskôr v režime offline. Karta s videom má aj funkcie ako: dať „Like“; označiť za videné alebo možnosť pridať do svojho albumu. Pri výbere albumov sa zobrazuje informácia o tom, či sa už dané video nachádza v kategórii alebo nie. Ak užívateľ pridať video do albumu, informuje ho o tom oznámenie, ktoré sa objaví pri odkaze, cez ktorý sa môže dostať na nástenku. Týmto sa snažím užívateľa informovať o jeho úkonoch.



Obrázek 4.8: Informovanie užívateľa.

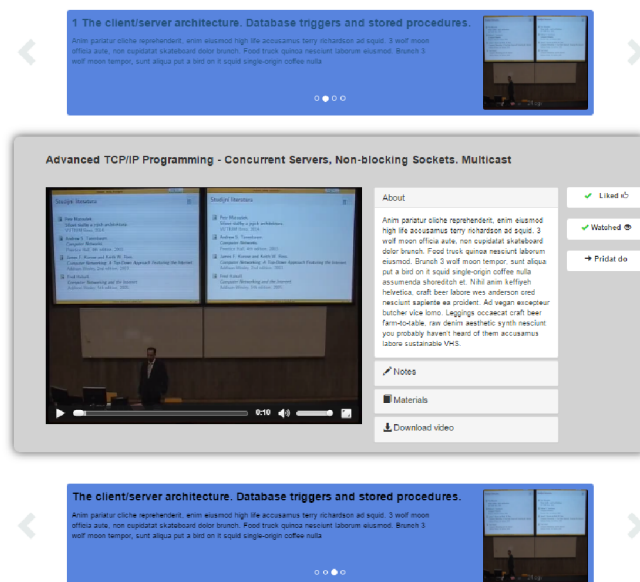
Prerekvizity a doporučené videá. Prvý návrh bol, že prerekvizity budú umiestnené nad hlavným videom, kde budú pod rolovacím tlačidlom. Videá boli reprezentované ako malé boxy, ktoré niesli informáciu o názve videa. Tu však vznikol problém, keď bol názov príliš dlhý, nezmestil sa a prečnieval mimo box. Tento problém môžeme vidieť na obrázku pod textom. Ďalší problém vznikol pri väčšom množstve videí, ktoré sa nezmestili do panelu.



Obrázek 4.9: Prvý návrh.

Druhý návrh rieši problém zobrazenia tak, že sa zobrazuje len jedna prerekvizita a je možné medzi nimi prepínať pomocou šípkok. Na karte je možné zobraziť mnoho viac informácií

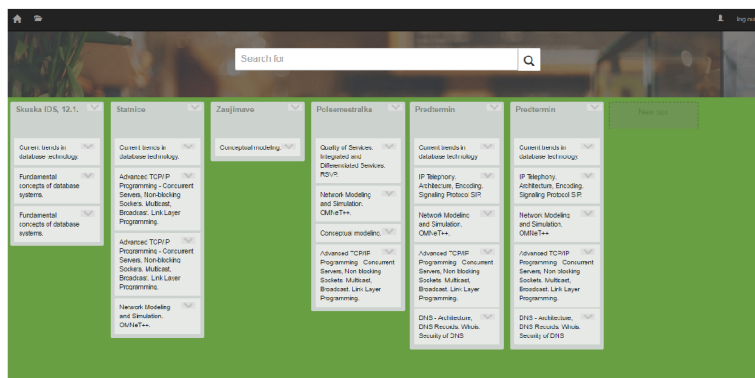
bez toho, aby užívateľ musel na kartu kliknúť. Prepínanie je nastavené tak, aby sa po určitom čase prepli videá automaticky. Príkladom nám môže byť situácia kedy si chce študent pozrieť konkrétne video. Nie každý si poctivo prejde všetky prerekvizity. Automatické prepínanie preto upozorňuje študenta na videá, ktoré možno nevidel. Bodky dole označujú počet prerekvizít a je možné nimi prechádzať medzi videami. Pre doporučené videá platí to isté. Rozdielom je, že sú umiestnené pod hlavným videom.



Obrázek 4.10: Druhý návrh.

4.3.3 Nástenka

Umožňuje užívateľovi si vytvárať vlastné kategórie a pridávať do nich videá. Spôsob tvorby kategórií ako reálnej nástenky som si vybral z toho dôvodu, aby si užívateľ nepamätal len podľa názvu videa ale aj podľa jeho fyzického umiestnenia na ploche. Pre umocnenie pocitu reálnosti nástenky je možné karty s kategóriami premiestňovať pomocou kurzora. Tak isto je možné v rámci kategórie premiestňovať jednotlivé videá. Pri každom videu je taktiež možnosť vymazať video.



Obrázek 4.11: Nástenka.

Kapitola 5

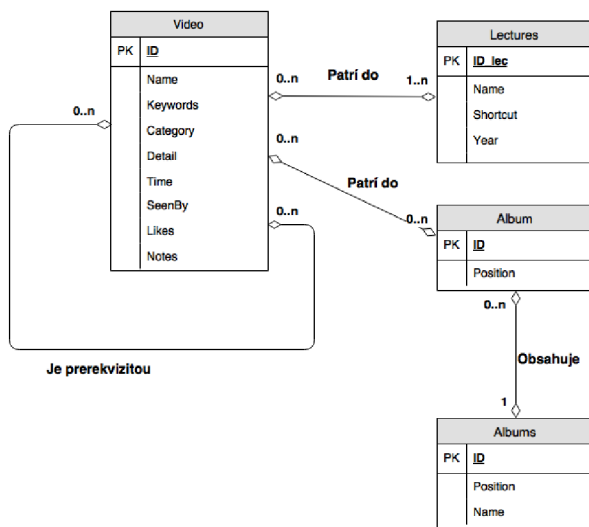
Implementácia

5.1 Výber nástrojov

Ako prvú technológiu na implementáciu som sa rozhodol použiť Nette framework. Táto technológia ponúka veľkú škálu možností a nástrojov pre tvorbu webových stránok a aplikácií. Avšak s touto technológiou som pracoval po prvý raz a narážal som na mnoho problémov. Preto som sa rozhodol, že si všetko naprogramujem ručne pomocou technológií: PHP, javascript, JQuery, Ajax, SQL, Bootstrap. Ako vývojové prostredie som použil codeLobster. Je to z dôvodu, že umožňuje pripojenie na server, má podporu pre všetky programovacie jazyky, ktoré som použil a je zdarma. Pre uloženie dát používam databázu na školskom serveri a prístupujem k nej pomocou phpMyAdmin.

5.2 Dátová štruktúra

Hlavným prvkom databázy je tabuľka „Video“. Sú v nej uložené všetky informácie o videách. Na obrázku 5.1 môžeme vidieť že videá majú navzájom medzi sebou vzťah. Tento vzťah reprezentuje skutočnosť, že jedno video je prerekvizitou druhého a druhé je doporučené k vzhliadnutiu po prvom.



Obrázek 5.1: ER diagram.

Pre prácu s databázou využívam SQL. Pomocou SQL som schopný ukladať a spätne získavať informácie o videách.

```
$sql2="SELECT video.Name, video.ID, video.Time
      FROM video INNER JOIN prerequisites
      ON video.id=prerequisites.ID_pre
      AND $video = prerequisites.ID_video";
```

Obrázek 5.2: Načítanie prerekvizít.

Pre prenos dát medzi serverom a klientom používam Ajax.

```
var xmlhttp = new XMLHttpRequest();
xmlhttp.onreadystatechange = function() {
    if (xmlhttp.readyState == 4 && xmlhttp.status == 200) {
        document.getElementById('main_big_frm').innerHTML = xmlhttp.responseText
    }
};
xmlhttp.open('GET', 'change_tops.php?vx='+vx, true);
xmlhttp.send();
```

Obrázek 5.3: Zmena filtru.

Pre dynamické prvky a animácie používam knižnicu jQuery.

```
jQuery(document).ready(function(){
    var i = 1;
    jQuery('#hideshow').on('click', function(event) {
        jQuery('#menu_sort').toggle('show');
        if(i == 0){
            $('#toggle_du').removeClass('glyphicon glyphicon-chevron-up');
            $('#toggle_du').addClass('glyphicon glyphicon-chevron-down');
            i = 1;
        }else{
            $('#toggle_du').removeClass('glyphicon glyphicon-chevron-down');
            $('#toggle_du').addClass('glyphicon glyphicon-chevron-up');
            i = 0;
        }
    });
});
```

Obrázek 5.4: Otáčanie kariet.

Grafické prvky ako ikony a panely sú z knižnice Bootstrap. Vzhľad ostatných prvkov je nadefinovaný podľa HTML 5 a CSS. Zdrojové kódy sú umiestnené na školskom serveri.

Vytvorená aplikácia slúži skôr ako návrh, prototyp, nie však ako plne funkčná verzia určená na zavedenie do užívania. Po funkčnej stránke by bolo potreba doimplementovať ešte mnoho vecí. Nie je funkčná registrácia užívateľov. Je vytvorený len jeden typ účtu, pomocou ktorého je možné pristupovať k rozhraniu. Mazanie karty z nástenky nefunguje úplne tak ako by malo a ani presúvanie kariet medzi albumami nie je možné.

Kapitola 6

Testovanie

Testovanie užívateľského rozhrania prebiehalo počas celého vývoja aplikácie. Tieto priebežné testy boli formou rozhovoru so spolužiakmi, ktorým som dal vyskúšať prototyp. Pri práci som ich pozoroval a všimol si aké úkony robia, čo je pre nich intuitívne a čo nie. Počas testovania som ich poprosil aby rozmýšľali nahlas. Toto mi pomohlo sa lepšie vžiť do role cieľového užívateľa. Aj za pomoci tohto priebežného testovania som vytvoril prvú verziu. Až po vytvorení prvej verzie som navrhol sadu testov.

Test č.1

Úloha: Zobrazit' videá týkajúce sa len oblasti počítačových systémov a sietí .

Cieľ: Zistiť ako rýchlo a či vôbec je užívateľ schopný nájsť prednášky na daný okruh.

Predpoklad: Predpokladom je, že užívateľ si vyfiltruje videá pomocou filtra kategórií na úvodnej stránke.

Výsledok:

Tester	Čas
1	30 s
2	8 s
3	38 s
4	54 s

Užívateľov ako prvé napadlo zadávať kľúčové slová do vyhľadávacieho okna. Výsledok vyhľadávania bol často prázdny a tak skúšali rôzne varianty kľúčových slov. Preto je výsledná doba tak dlhá. Iba jedného testera napadlo hneď použiť filter pre kategórie.

Záver: Z postupu testerov som usúdil, že pre užívateľov je najintuitívnejšie vyhľadávať pomocou kľúčových slov. V čase testovania nebola databáza dostatočne naplnená a preto užívatelia nemohli vyhľadať videá na základe kľúčových slov. Tester boli zmätení ak sa im ako výsledok vyhľadávania zobrazila prázdna stránka. Pre zlepšenie je potreba naplniť databázu a pridať možnosť vyhľadávania aj na základe kategórií.

Test č.2

Úloha: Vytvoriť nový album a do neho vložiť 4 zadané videá.

Cieľ: Otestovať ako rýchlo zvládne nezalý užívateľ riešiť komplexnejšie úkony.

Predpoklad: Predpokladám, že užívateľ bude mať problém nájsť nástenku s albumami.

Výsledok: Predpoklad sa naplnil. Užívateľom trvalo pomerne dlho pokiaľ našli kde sa nástenka nachádza.

Tester	Čas
1	1 min 14 s
2	1 min
3	2 min 7 s
4	1 min 30s

Záver: Z pozorovania úkonov, ktoré vykonávali testovacie osoby, som si potvrdil svoj predpoklad, že ikona ktorá odkazuje na nástenku je pomerne malá a nový užívateľ si ju nemusí hneď všimnúť.

Test č.3

Úloha: Nechať užívateľa samovoľne pracovať s rozhraním. Voľné testovanie všetkých možností. Testovacia osoba už je oboznámená s prostredím z predchádzajúcich testov. Premýšľať nahlas, zamerať sa na kritiku- čo by zmenili.

Cieľ: Zistiť užívateľove dojmy a pocity, čo sa mu páči a čo by zmenil.

Výsledok: Testerov zaujali otáčacie karty na úvodnej stránke. Hodnotili to ako zaujímavé riešenie. Páčilo sa im, že na zistenie všetkých potrebných informácií o videu stačí len pohyb kurzora. Naopak sa im nepáčilo to, že všetky karty obsahovali rovnaký obrázok. Na stránke s videom veľmi kladne hodnotili možnosť si video stiahnuť a písať si vlastné poznámky. Taktiež sa im páčila možnosť si videá triediť do vlastných albumov. Čo však hodnotili ako veľké mínus bolo to, že jednotlivé videá nie je možné presúvať medzi albumami. Pri presúvaní albumov nebolo moc intuitívne uchopovanie karty. Karta sa dala chytiť kurzorom a preniesť len v jednom určitom bode. Testerov to dosť znepokojovalo a považovali to za negatívum.

6.1 Úpravy na základe výsledkov testov

Ako prvé som doplnil databázu a vložil viacero kľúčových slov k danej prednáške. Vďaka tomu je možné vyhľadať danú prednášku pomocou viacerých kľúčových slov. Zvýraznil som ikonu odkazujúcu na nástenku, aby užívatelia vedeli, že rozhranie ponúka danú funkcionálitu. Upravil som kurzor pre uchopenie albumu, aby užívateľ vedel kedy je možné preniesť danú kartu a kedy zasa nie. Po zmenách som poprosil testerov, aby spätne vyskúšali rozhranie a vyjadrili sa či som ich pripomienkam vyhovel. Ich pripomienkam som vyhovel a viac som rozhranie neupravoval.

Kapitola 7

Návrhy na pokračovanie

Existuje mnoho spôsobov ako vylepšiť doposiaľ implementované užívateľské rozhranie. Uvediem tie, nad ktorými som počas tvorby uvažoval, ale rozhodol som sa ich neimplementovať, alebo som ich návrh nedotiahol dokonca.

- Možnosť zdieľať albumy medzi užívateľmi. Táto funkcionality napadla môjho vedúceho a príde mi to ako veľmi dobrý nápad. Študent si prichystá album s prednáškami a aby uľahčil prácu ostatným, vygeneruje si link, ktorý im pošle. Spolužiakom sa potom zobrazí daný album. Návrh vychádza z reálnej situácie, kedy si študenti často vytvárajú zdieľané dokumenty, kde si poznamenávajú dôležité časy z video prednášok.
- Rozšíriť aplikáciu pridávaním komentárov k videu. To by umožňovalo študentom pokladať dotazy a poznamenávať postrehy, ktoré by videli všetci a tým diskutovať danú problematiku. Taktiež by to autorom videa pomohlo získať spätnú väzbu o ich kvalite prednášky v tom zmysle, že by zistili, čomu študenti nerozumejú a čo je pre nich ťažké pochopiť.
- Pridať zobrazenie materiálov priamo na stránke, bez potreby ich sťahovať. Tým by sa docielilo, že užívateľ by pristupoval ku všetkým materiálom cez svoj prehliadač a nemusel by nič sťahovať, pokiaľ by o to nemal záujem.

Kapitola 8

Záver

Moja bakalárska práca sa zaoberá problematikou zefektívnenia učenia sa pomocou digitálnych prednášok. Výsledný produkt je určený predovšetkým študentom medzi ktorých patrím aj ja a to je jeden z dôvodov prečo som si vybral tuto bakalársku prácu. Konkrétnym cieľom bolo navrhnuť a implementovať webové užívateľské rozhranie.

Pred samotným návrhom som si musel naštudovať a zhromaždiť informácie týkajúce sa tvorby užívateľských rozhraní. Taktiež bolo nutne zamyslieť sa nad potrebami študentov. Získané informácie boli základom pri tvorbe návrhu.

Môj produkt pozostáva zo štruktúry navzájom prepojených videí. Jednotlivé videá sú vystrihnutými časťami z video prednášky, ktoré zachytávajú konkrétnu a špecifickejšiu tematiku. Vzťah, vďaka ktorému som túto štruktúru vytvoril je založený na obsahovej nadväznosti daných videí. Tato funkcia umožňuje študentovi kontinuálne prechádzať systémom videí a tým rozširovať jeho vedomosti k danej téme. Každé video ponúka aj doprovodné materiály ako napríklad: informácie o tom, čo sa v prednáške dozvieme, možnosť stiahnuť si prednášku a skriptá vo formáte pdf alebo si stiahnuť samotnú prednášku ako video. Ponúka sa tu aj možnosť viesť si vlastné poznámky k jednotlivým videám. Ako experimentálnu funkčnosť som navrhol vytváranie si vlastnej nástenky. Tu je možné si vytvárať vlastne albumy s jednotlivými videami a pracovať s nimi ako na reálnej nástenke. Pri tvorbe som využil programovacie jazyk: HTML, CSS, PHP, SQL, javascript a knižnicu Bootstrap. Stránka je umiestnená na školskom serveri a na ukladanie dát využívam databázu MySQL.

Moju prácu nemožno považovať za produkt určený k zavedeniu do používania, ale predovšetkým ako prvotný projekt, ktorý ponúka náhľad na jeden zo spôsobov podania digitálnych prednášok a pomáha rozvinúť myšlienky pri tvorbe ďalších návrhov.

Literatura

- [1] Ben Shneiderman, Catherine Plaisant: *Designing the User Interface: Strategies for Effective Human-Computer Interaction (4th Edition)*. Pearson Addison Wesley, 2004.
- [2] Encyclopedia: *Graphical User Interface*. 2007, [Online; navštívené 6.5.2016].
URL http://www.encyclopedia.com/topic/Graphical_User_Interface.aspx
- [3] MAKULOVÁ, S.: *Informačná architektúra a jej vplyv na nájditelnosť webových stránok*. 2009, [Online; navštívené 6.5.2016].
URL <http://www.elet.sk/?pouzitelnost-pristupnost&sprava=pouzitelnost-webovych-sidiel-ako-zakladny-predpoklad-ich-uspesnosti>
- [4] NIELSEN, J.: *Why You Only Need to Test With 5 Users*. 2000, [Online; navštívené 6.5.2016].
URL <http://www.useit.com/alertbox/20000319.html>
- [5] Wikipedia: *Command-line interface*. 2016, [Online; navštívené 6.5.2016].
URL https://en.wikipedia.org/wiki/Command-line_interface
- [6] Wikipedia: *CSS*. 2016, [Online; navštívené 6.5.2016].
URL https://en.wikipedia.org/wiki/Cascading_Style_Sheets
- [7] Wikipedia: *HTML*. 2016, [Online; navštívené 6.5.2016].
URL <https://en.wikipedia.org/wiki/HTML>
- [8] Wikipedia: *Text-based user interface*. 2016, [Online; navštívené 6.5.2016].
URL https://en.wikipedia.org/wiki/Text-based_user_interface

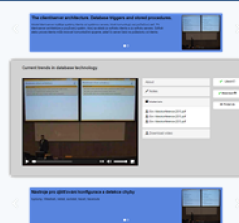
Príloha

Plagát

UŽIVATELSKÉ ROZHRAŇÍ PRO SPRÁVU DIGITÁLNIČ PŘEDNÁŠEK

Bakalárska práca

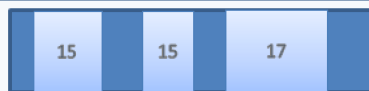
Táto bakalárska práca sa zaoberá užívateľským rozhraním pre správu digitálnych prednášok a ich sprievodných materiálov. Podstatou riešenia je vytvorenie sieťových štruktúr pozostávajúcich z jednotlivých videí, ktoré spolu navzájom súvisia s možnosť touto štruktúrou prechádzať. Aplikácia ďalej umožňuje užívateľovi vytvárať vlastné zoznamy videí a pracovať s nimi ako s reálnymi záložkami na nástenke.



Na obrázku je zobrazené hlavné video v strede a panely prerekvizitami (hore) a doporučenými videami (dole).



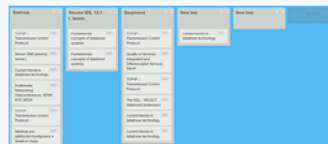
Sieťová štruktúra znázorňuje vzťahy medzi jednotlivými prednáškami



60 min

Jednotlivé videá nepozostávajú z celej prednášky ale len z častí zaoberajúce sa jednou témou.

Nástenka, kde si užívateľ môže vytvárať vlastné zoznamy a vkladať do nich videá. Pre presuny jednotlivých videí je použitá priama manipulácia „Drag and Drop“



Autor: Daniel Košťalik
Vedúci: prof. Ing. Adam Herout, Ph.D.

