

Vysoké Učení Technické v Brně



Projekt ITU, testování struktury menu, 2016Z

Číslo projektu: 40

Číslo a název týmu: 132, xstude21

Autor: Jan Pawlus (xpawlu00)

Další členové týmu: Zdeněk Studený (xstude21), David Kočnar (xkocna01)

Termín řešení: 19. 9. - 12. 12. 2016

1 Abstrakt

Cílem projektu je vytvoření několika typů menu pro určitou aplikaci a následné testování, která varianta je pro uživatele nejvýhodnější. Důležitým faktorem zde je čas, který s jednotlivými variantami uživatelé stráví - každé menu může totiž být jinak efektivní po určitém čase používání, nejde zde tedy jen o prvotní intuitivitu. Dalším nezanedbatelným faktorem se stal samotný obsah menu a dobře pojmenované položky.

2 Cílové požadavky na aplikaci a její rozhraní

Pro naše účely jsme si vybrali jako téma aplikace program pro správu personalistiky a mezd ve firmě. Samotná aplikace v tomto projektu ale v podstatě nehraje roli, tu hraje pouze rozhraní. Pointa tkví v tvorbě různých typů menu, nejlépe tak, aby se od sebe varianty co nejvíce lišily - ne jen v grafickém designu, ale hlavně ve struktuře. Dá se říct, že požadavek na aplikaci je v podstatě sada testů, která vyhodnocuje využívání jednotlivých variant menu. Spokojeného uživatele následně poznáme na základě výsledků testů - varianta, kde je uživatel nejefektivnější (rychlá orientace v menu a nejméně chyb), je očividně ta nejsprávnější pro typ menu, které náš tým vytvoří. Výsledky se také mohou lišit podle skupiny uživatelů, která je testována.

3 Studium cílové skupiny a případy užití

Cílová skupina je zde velmi široká - program pro personalistiku a mzdy může používat takřka kdokoli - od vývojářů této aplikace přes manažery a zaměstnance v dané firmě, kde je aplikace nasazená, až po mzdové účetní. Obsah menu je ale variabilní, na tomto faktoru tedy nesejde. Pokud se budeme bavit o testování, tam je cílová skupina stejná - jde totiž o to otestovat všechny možné skupiny lidí, co budou jakoukoliv aplikaci používat. Případy užití zahrnují situace, kdy vývojář potřebuje zjistit, jaký typ menu má pro vyvíjenou aplikaci zvolit.

4 Existující řešení

Existující řešení zároveň existuje a neexistuje - všechny typy menu, které jsou implementovány, již existují, a každý typ má své plus a mínus (HotKeys menu - po naučení rychlé na používání, ale ze začátku neintuitivní apod.). Na druhou stranu aplikace, která testuje pouze menu, dle našeho průzkumu neexistuje, místo toho existují rozsáhlejší programy, které testují celkovou strukturu aplikace, nejen menu (Hotjar, UXpin).

5 Návrh GUI

Aplikace je rozdělena na čtyři části - první budou úvodní informace k aplikaci, návod k použití, formuláře na feedback a vysvětlení všeho, na co by se mohl uživatel zeptat. Druhou je samotné menu, zatímco třetí symbolizuje instrukce uživateli o tom, co má v menu najít. K dispozici jsou tři různé formy menu, pokaždé ale budou uživateli určitou formou přístupné všechny položky. Varianty GUI se tedy liší na základě aktuálně probíhajícího testu, který je určen strukturu a rozvržením menu.

Charakteristiky a výhody jednotlivých variant můžeme předpokládat, přesné výsledky ale dostaneme pouze kvalitním testováním. Základní typy menu, které byly pro tento projekt zvoleny, jsou vertikální menu, horizontální menu a menu s klávesovými zkratkami.

Poslední částí GUI jsou grafické statistiky týkající se výsledků testů.

6 Návrh a implementace back-endu

Náš tým se rozhodl pro variabilitu a proto lze definovat jednotlivé struktury menu v JSON formátu mimo samotný kód. Jedna z funkcí aplikace je načtení tohoto JSONu a na základě obsahu vytvoření menu. Jako další je sada několika testů, kde pointa každého testu tkví v měření času mezi zobrazením instrukce uživateli a úspěšnou navigací na zadanou položku menu. Pomyslný back-end tvoří Javascript, který se stará o veškerou funkčnost (střídání menu, samotné testování, ukládání výsledku do JSON formátu a tvorba statistik).

7 Návrh uživatelských testů

Nejlepší způsob pro testování naší aplikace bude vlastně samotná funkčnost - zobrazování různých struktur menu a na každé této struktuře určitá sada úkolů, které má tester co nejrychleji splnit. Podle rychlosti jeho orientace v menu zjistíme adekvátní výsledky, se kterými jde poté pracovat a porovnávat s jinými strukturami. Na konci testů má také tester možnost napsání feedbacku pro nás, díky čemuž jsme zjistili chyby v našem návrhu uživatelského rozhraní. Další forma testování naší aplikace je celková statistika správných a špatných kliknutí jednotlivých položek menu - tím jsme zjistili, zda jsme neudělali chybu v sadě úkolů ke každé položce menu (nejednoznačné zadání atd.). Testeři budou v podstatě simulovat uživatele, a na základě výsledků lze zjistit, zda jsou naše úkoly srozumitelné a splnitelné. Feedback od testerů zohledňuje samozřejmě také vzhled a intuitivitu uživatelského rozhraní.

8 Nástroje pro tvorbu GUI

Pro naši aplikaci se nejvíce hodí webové technologie. Použili jsme tedy HTML, CSS (s podporou Materialize CSS) a Javascript, resp. JQuery a AJAX. Výhoda je velká variabilita stylování, kterou nám ostatní technologie neposkytují. V CSS můžeme udělat opravdu jakékoliv menu a dynamicky měnit jeho vzhled, což je velmi důležité.

9 Programování GUI

Samotné prvky GUI byly naprogramovány pomocí HTML a CSS, kde se využívaly prvky Material designu. GUI je tedy složeno z jednotlivých HTML stránek, kde má každá určité CSS styly, stejně tak jako každá struktura menu. Uživatelské rozhraní je složeno z textových informací o nástroji, z instrukcí pro uživatele, formulářů pro informace o uživateli o feedbacku, z grafických znázornění výsledků ve formě grafů a ze samotného menu (klávesové zkratky byly implementovány HTML atributem `accesskey`).

Ve zkratce je tedy aplikace naprogramována takto: **AJAX** na začátku načte **JSON** soubor, ze kterého vytvoří **HTML** menu, a kde jsou obsaženy i sady úkolů pro uživatele (ty jsou poté uživateli náhodně vybrány). Na základě struktury, která se bude zrovna testovat, se aplikují styly **CSS** a celý průběh testování je vyřešen **Javascriptem** (záznam času času, snímání klikání a kláves, zobrazování instrukcí, střídání menu a vyhodnocování).

Mým úkolem bylo již zmíněné načtení **JSONu** (plus návrh položek menu a úkolů) do **HTML** menu, práce s instrukcemi v globálním poli (náhodný výběr ke každé položce), veškerá práce s klávesovými zkratkami a ukládání výsledků zpět do **JSONu**.

10 Uživatelské testy

10.1 Funkčnost aplikace

Jelikož funkčnost aplikace je testování, je nutné rozdělit uživatelské testy na dvě části. Aplikace testuje orientaci uživatele v různých strukturách menu a také vhodnost názvů jednotlivých položek menu. Uživateli jsou tedy střídány tři struktury menu, kde každá položka má určitou sadu úkolů, nad kterou se musí uživatel zamyslet a vybrat správnou položku.

10.2 Testování návrhu aplikace

Testování proběhlo nahráním aplikace na webový server a zaslání odkazu lidem (bylo osloveno co nejvíc lidí s co největší diverzitou věku a profesního zaměření), ale bez většího vysvětlování, co mají lidé vlastně dělat - jediné tak jsme mohli zjistit, zda jsme uživatelské rozhraní dobře navrhli, jelikož by měl uživatel pochopit z informací na webu, co že se to po něm chce, a měl by být schopen aplikaci ovládat sám. Byl také zprovozněn feedback, kde po ukončení testů měli lidé vyplnit, co si o testování myslí, popřípadě co se jim nelíbilo a s čím si nevěděli rady, což byl důležitý zdroj informací, na základě kterých jsme aplikaci dále upravovali (stejně tak jako výsledky testů, kdy jsme zjistili, zdali jsme některé úkoly nenavrhli nejednoznačně či zmatečně). Testování tedy, jak již bylo zmíněno výše, probíhalo prostou funkčností aplikace, kdy lidé dostali 45 úkolů při měnícím se menu.

11 Výsledky testů

11.1 Testování s pozorováním

Všichni testovaní dokončili test, aplikace tedy je použitelná, během testování se však objevily nežádané situace, které dále rozebereme. U dvou testovaných se stalo, že si nepřečetli instrukce potřebné k plnění úkolů (užívání aplikace). Po rozhovoru s testovanými jsme se dozvěděli, že nechtějí číst dlouhý úvodní text, nýbrž rovnou začít testovat a naučit se aplikaci používat za běhu. Řešením takového problému by mohlo být přetvoření bodového úvodu do samotného testu. První by tedy uživatel prošel neměřený test, jehož úkoly by jej seznámili s podmínkami testování. Zároveň by měl možnost vyzkoušet si rozhraní před ostrým měřením testu.

11.2 Osvojení si úkolů

Další připomínky směřovaly k jistému osvojení si prováděných úkolů, což způsobovalo, že uživatelé v druhé polovině věděli, kde hledat řešení, aniž by nad ním museli přemýšlet.

„U třetího menu jsem už věděl na co klikat a nemusel nad úkoly přemýšlet, řekl bych, že je zvýhodněné.“

Tento problém jsme vyřešili náhodným střídání pořadí menu, každý tester tedy testuje menu v jiném pořadí. Sice k osvojení úkolů dochází i nadále, avšak s dostatečným vzorkem testerů se tato nesrovnalost ve statistikách zprůměruje. Dalším možným řešením by mohlo být přidání více úkolů, aby se neopakovaly.

11.3 Klávesové zkratky

Zde jsme narazili na úskalí webových technologií. Na různých operačních systémech a v různých prohlížečích se může atribut `accesskey` chovat podstatně jinak - rozdíl je například v operačních systémech (MAC vs. Windows, kde MAC klávesové zkratky aktivuje stisknutím **Control** tlačítka a klávesy, na Windows je to **Alt**) a prohlížečích (Chrome vs. Firefox - **Alt** vs. **Shift + Alt**). Bylo tedy nutné v instrukcích pro uživatele uvést, pro jaké kombinace OS a prohlížeče funguje která kombinace kláves.

Pro naše testování jsme u zkratkového ovládání nechali i možnost na položky v menu kliknout.

„Na menu se s klávesovými zkratkami jsem klikal.“

Zde jsme cítili jistou nevýhodu orientace pomocí klávesových zkratek, ty je totiž třeba si osvojit, aby se začaly časově vyplácet. Řešením by mohlo být přidání více typů menu a přidání čistě zkratkového menu na závěr, kde by byla možnost klikání deaktivována.

11.4 Beta testing

Při beta testingu bylo cílem zjistit, zda data získaná z naší aplikace jsou opravdu použitelné, podporují analýzu a další možnost zlepšení menu. Nasbírané data odhalila rozdíly mezi jednotlivými menu. Data také ukázala problémovou položku menu, kde by bylo vhodné dále prozkoumat úkoly a správný název položky menu.

11.5 Možná vylepšení

Počet sbíraných dat by se mohl rozšířit o zaznamenávání obrazovky a pohybu myši. V propojení s kamerou by tak aplikace dávala designérovi možnost osobně rozebrat testování, aniž by musel být během testování přítomný.

12 Závěr

Tento projekt bych shrnul tak, že na základě odezvy od testerů byl vytvořen poměrně kvalitní nástroj pro testování struktur menu, který je díky své variabilitě jednoduché upravit pro menu jakékoliv aplikace. Zjistili jsme, že není důležité pouze to, jak menu vypadá, ale také jeho obsah - položky v menu je nutné navrhnout tak, aby uživatel přesně věděl, kam chce kliknout. Z výsledků nelze zcela jasně určit, zda je lepší vertikální nebo horizontální menu, dle zpětné vazby od testů je ale důležité to, aby se muselo co nejméně klikat - víceúrovňové menu je tedy vhodné co nejvíce otevřít, rozklikávání jednotlivých úrovní není žádoucí. Klávesové zkratky jsou pak bonusem, který lze uplatnit například při každodenním užívání dané aplikace poté, co si uživatel na menu zvykne.

Potenciální rozšíření aplikace by mohl být online editor, kdy by si mohl uživatel vytvořit jednotlivé položky a podpoložky svého menu (tyto údaje by byly přetransformovány do JSON souboru, ze kterého aplikace menu čte) a získat odkaz na samotný test, kde by bylo jeho menu obsaženo.

13 Práce v týmu

Spolupráce probíhala mezi třemi studenty, kdy kde sdílení kódu byl použit nástroj **Github**, ke komunikaci web **Facebook.com** (či osobní setkání) a k rozdělení úloh pak sloužilo **Trello**. Celkově nebyl ve spolupráci až na výjimky větší problém a projekt tak byl dotažen do kýženého konce jak z funkčního hlediska, tak z hlediska testovacího.

Pokud tým funguje, je práce v něm určitě přínos, jelikož si lze rozdělit úkoly, kontrolovat po sobě již implementované věci, vymýšlet další postup, sdílet znalosti s ostatními lidmi z týmu a testovat. Náš tým naštěstí fungoval tak, jak má, práce v něm tedy byla přínosem. Další plus je například dohlížení na stanovené termíny.

14 Reference

Vzhledem k tomu, že podobná aplikace pravděpodobně neexistuje, jsme veškeré informace čerpali z vědomostí získaných na přednáškách předmětu ITU a ze zpětné vazby od testerů, kdy se na základě toho aplikace upravovala.

15 Příloha - testovací protokol

15.1 Testovací prostředí

Testovaná aplikace by neměla být omezená prostředím. Testovaná totiž může jak aplikace pro korporát, kde je vhodné testovat v kancelářském prostředí, tak i menu webových služeb a stránek - v takovém případě se variabilita prostředí velice cení.

15.2 Testované osoby

Spektrum testovaných osob by mělo být co nejširší, avšak primární je cílová skupina, která pracuje s počítačem na denní bázi. Za tu se dají označit pracující lidé od 18 do 50 let. Zastoupení obou pohlaví je zde také vhodné.

15.3 Příprava testovaných osob

Celý koncept testování je zde pro testované osoby trochu zmatečný, testují totiž aplikaci na testování. Je proto třeba jim dobře vysvětlit, co je cílem a na co se mají zaměřit, na druhou stranu není žádoucí vysvětlit testerům, jak se má aplikace používat - je nutné, aby toto pochopili ze samotné aplikace, v opačném případě to znamená, že uživatelské rozhraní aplikace není správně navrženo. Po ukončení testu by s testovanými subjekty měl proběhnout rozhovor o jednotlivých krocích a nejasnostech v testování.

15.4 Testování s pozorováním

U tohoto testování je třeba být s testovaným v místnosti a nebo mít aktivní videohovor. Je důležité, aby tester byl uvolněný, a aby upřímně sděloval své aktuální myšlenky zapisovateli. Zapisoval by neměl do testování nijak zasahovat, pouze zaznamenávat poznámky testera a výrazné emoce při testování.

Tento způsob je časově náročnější a také se hůře zpracovávají získaná data. Ukáže však myšlenkové procesy uživatelů, kteří přišli do kontaktu s aplikací poprvé.

15.5 Beta testing

Při testování aplikace na testování je třeba také zjistit, jestli její provoz nese kýžené výsledky. Při tomto testování je důležité rozeslat test testerům a nechat je interagovat pouze s aplikací. Výsledkem testování by měly být poznatky nad uživatelským rozhraním a funkčností aplikace.

15.6 Čas potřebný k testování

Pro testování s pozorováním je třeba počítat s časem na provádění testu, tedy 5-10 minut + čas na diskuzi o testování, tedy dalších 10 min. Tento čas potřebují mít jak testovaný, tak dohlížející.

Pro beta testing je délka jednoho testu 5-10 minut a testeři jej vyplňují nezávisle na někom dalším. Jde spíše o masovou záležitost. Pro hodnotitele je poté třeba vyhradit si hodinu až dvě času, kdy prozkoumá výsledky a dojde k závěrům. Čas potřebný k vyhodnocení je také silně ovlivněn počtem nasbíraných výsledků.