Posudek oponenta bakalářské práce

Autor/autorka práce: **Hana Hrkalová**

Název práce: **2D vizualizace výsledků hierarchicky strukturovaných testů**

Obsah práce

Text práce je logicky strukturovaný jen při letmém prvním pohledu. Při bližším ohledání jsou kapitoly a menší celky místy nelogické jak pořadím, tak dělením. Například první částí hned po Úvodu je Analýza knihoven pro práci s formátem JSON, bez širšího kontextu, ze kterého práce zjevně vychází. Popis TbUIS a SquashTM přicházejí až v podkapitole 2.3 a popis dosavadní implementace zamýšlené funkčnosti dokonce až v kapitole 3, která se podle názvu má zabývat implementací (čekal bych, že výhradně navrženého řešení, ne analýzou stávajícího stavu). Na začátku nevíme, proč se zabýváme JSON formátem, proč jenom jím a proč pak JSON na téměř 10 stránek opustíme a mezitím čteme z ničeho nic o CSV exportech. To je bohužel jen jeden příklad za všechny a tento úkaz se objevuje na všech úrovních od kapitol, až po dělení jednotlivých odstavců a vět. Testování je zmíněno letmo ve dvou, hluboko zanořených podkapitolách a návrhy na budoucí rozšíření jsou letmo zmíněné. Přitom oboje by si zasloužily vlastní ucelené sekce.

Po obsahové stránce jsou tu další nedostatky: popis implementace z velké části působí spíše jako návod k použití (ten je v příloze A a s kapitolou Implementace nemálo textu sdílí), není zmíněn rozsah (počet testerů nebo jednotkových testů), pokrytí ani výsledky testování, chybí instalační návod, celková architektura řešení, vysvětlení (zjevně netriviálního) toku dat z několika zdrojů skrze aplikace až do následného generování výsledků. Na druhé straně text zabíhá místy do zbytečných detailů jako popisy jednotlivých proměnných a změn barev na GUI oproti výchozí verzi.

Ač implementačně se jednalo o dvě aplikace, jejich složitost na realizaci a rozhodně na popis se nezdá tak významná, aby to opodstatnilo takto nesystematický text. Rozsah je 46 stran od Úvodu po Závěr, ale nutno dodat, že mnoho místa zabírají četné kódové výpisy (mnohdy zbytečné).

Kvalita řešení a dosažených výsledků

Nejlepším aspektem výstupů jsou přiložené vygenerované HTML soubory, které představují element v mnoha aplikacích využitelný a zatím chybějící. I v nich by se ale našly drobné vady, jako nekonzistence číselného formátu („1“ vs. „1.0“).

Z aplikací se mi podařilo spustit pouze jednu (oks_app) a to v GUI formě. Ta nebyla nijak přesvědčivá (obsahuje překlepy), intuitivní (nestandardní použití menu), ani přívětivá (výběr souborů začíná vždy z kořenového adresáře stroje), čemuž nepomohl ani uživatelský manuál v příloze práce, který mluví více o použití vytvořené knihovny (opět nesystematicky a těžko pochopitelně) a spuštění aplikací, ale postrádá klasický návod k použití opatřený screenshoty.

Kód je sice dobře strukturovaný (ve smyslu délky tříd a metod a čitelnosti díky odsazování), ale při nejlepším velmi nekonzistentně komentovaný dokumentačními komentáři (ve smyslu jen někde, do různé míry a podle nejasných konvencí) a vůbec ne doplňujícími řádkovými komentáři. Vygenerovaný JavaDoc v odevzdání chybí úplně. Navíc kód používá mnoho přímých textových (i jiných) hodnot/proměnných, místo konstant nebo konfiguračních souborů (u textů pro GUI nebo chyby je to do očí bijící), a jiné špatné programátorské praktiky.

Absence jednotkových testů u general_app je omluvitelná, jelikož jde jen o ukázkovou aplikaci, používající jeden z hlavních výstupů, TableGenerator knihovnu. Ale i u této knihovny a oks_app je testů podezřele malé množství.

Formální úroveň

Na úrovni odevzdaného archivu, general_app a knihovna používají na sestavení Gradle, kdežto oks_app Maven, objevují se složky, soubory a dokonce třídy s „temp“ v názvu, target složky s očividně staršími sestavenými verzemi aplikací, opakující se třídy v různých balících jedné aplikace, atd. Obecně to působí jako export neudržovaného workspace, více než na čisto připravená adresářová struktura k odevzdání.

Úroveň textu je také velmi špatná. Krom nelogického členění se objevují velké problémy s interpunkcí, překlepy a skloňováním. Dále nekonzistentní (ne)používání odrážkových seznamů, některé zkratky nejsou nikdy vysvětleny (ani v jejich seznamu; např. API, CSS, HMTL, KIV), některé skočí z českého pojmu rovnou na anglickou zkratku, která se chvíli používá, a pak se přejde opět k plnému názvu v češtině (např. TC, UC, RQM). Na druhé straně se v seznamu zkratek objevují „TableGenerator“ a „selectOneMenu“, pojmy hodné vysvětlení, ale nikoli zkratky. Nekonzistence je i ve formátování. Někdy je návěští odrážek seznamu formátováno, někdy ne, někde ani odrážky. „Kódový“ font se také používá nekonzistentně při zmínkách v textu.

Jazykově je text místy zároveň triviální a náročný. Často se ve stejné větě objevují stejná slova nebo slova o stejném základu dvakrát a více. Často také věta začíná (téměř) stejně, jako předchozí skončila. To vede na instance, kde je třeba v jedné větě třikrát použito slovo „hodnoty“, vždy referuje k něčemu jinému a následující věta začíná slovy „Tyto hodnoty…“, což je nad míru matoucí. Dále jsou některé věty zkrácené tak, že nemají podnět nebo přísudek a jiné zase pokrývají několik myšlenek, ač obsahují přirozená místa, kde měly být rozděleny.

Obrázky (pokud vůbec existují) jsou z většiny nečitelné. To by nevadilo tolik u převzatých, ale i ty mohly být alespoň roztaženy na šíři textu. Navíc nejsou odkazovány v textu a ledabyle umístěny (platí i pro tabulky). Ještě větší problém je ale absence některých obrázků. Architektura, diagram balíků a tříd, datové modely, toky dat apod., to všechno by pomohlo srozumitelnosti popisu implementace, který je takto velmi těžko uchopitelný, i díky již dříve zmiňovaným faktorům.

Nedostatků je mnohem více na všech úrovních, zmiňuji jen ty nejzávažnější. Jak text, tak odevzdaný archiv celkově působí jako narychlo daný dohromady, ač práce na pozadí pravděpodobně bylo nemalé množství. Výstupy to bohužel nereflektují.

Práce s literaturou

Všechny citované zdroje jsou online.

Splnění zadání

Splněno s výhradami. U bodu 3 se nedá posoudit „odpovídající pokrytí“ testy, jelikož to není nikde kvantifikováno.

Doplňující informace k práci

Nezdařené spuštění/použití aplikací mohlo být mojí chybou, nicméně absence, respektive nízká kvalita a úroveň detailu materiálů k tomu použitelných této situaci rozhodně ani v nejmenším nepředchází.

Dotazy k práci

1) Můžete nakreslit schéma (ne nutně ve formální notaci) základních částí obou aplikací (rozdíl je, předpokládám, nevelký) včetně zdrojů všech vstupních dat, jejich zdrojů, průchodu aplikacemi, až po generování výstupů?
2) Jaké je pokrytí oks_app a TableGenerator jednotkovými testy? Testovali aplikaci jiní uživatelé za hranici vás a vedoucího práce? Byly k tomu nějaké scénáře? Jaké byly výsledky testů?
3) Objemu kroků, které někdo musí postoupit pro použití knihovny TableGenerator (připojení knihovny, seznámení s anotacemi, jejich správná aplikace, příprava JSON, CSV a TXT dat ve správném formátu) je nezanedbatelný pracností i expertízou. Nebylo by snazší mít definovanou strukturu JSON souborů, které by obsahovaly vše podstatné (tzn. kompletní hierarchickou strukturu, popisky a tooltipy na všech úrovních, hodnoty, odkazy, atd.) a aplikaci, která by je jen zpracovala do HTML a JS? Tvorba JSON by pak byla na uživateli, lhostejno jak. Ručně by je pak dokázali sepsat i neprogramátoři, základna potenciálních uživatelů by se zvětšila a pracnost užití by zůstala (zdá de) maximálně stejná.
4) K jak velkým změnám by vedlo a jak náročné by bylo doimplementovat do knihovny více možností agregace (tzn. nejen sumarizaci, ale např. i součin, logické operace, atd.) v „zabalených“ buňkách tak, aby si uživatel knihovny mohl při generování HTML vždy vybrat, která operace se kde použije?

Navrhuji hodnocení známkou **dobře** a práci doporučuji k obhajobě.

V Plzni 30.5.2022 Ing. Petr Pícha