

Slovenská technická univerzita v Bratislave

Fakulta informatiky a informačných technológií

Projekt 1:

Implementácia techník čiernej skrinky pri návrhu
testovacích prípadov

Obsah

1. Popis projektu	3
2. Cieľ projektu.....	3
3. Testovanie čiernej skrinky.....	3
4. Využitie techník čiernej skrinky pri tvorbe testovacích prípadov.	4
4.1 State Transition Testing.....	5
4.2 Decision Table Testing.....	6
4.3 Error Guessing	7
5. Zhodnotenie	8
Zoznam zdrojov	10

1. Popis projektu

Projekt je moja diplomová práca ktorá sa zaoberá manažmentom datasetov, anotácii a následnou anotáciou stĺpcov datasetu podľa dostupných anotácii a exportu týchto anotácii. Anotácie sú v stromovej hierarchickej štruktúre a pri ich listovaní môžeme vidieť ich štruktúru od začiatku až po koniec to znamená po najvyšší prvok. Toto môžeme vidieť pri ich listovaní na stránke alebo pri exporte.

Táto stránka je dostupná momentálne u mňa na localhoste ako developerské prostredie. Využíva Neo4j databázu avšak mám aj PostgreSQL databázu. Ktorú databázu stránka použije sa vyberá pomocou .env súboru. K dispozícii sú tri typy používateľov a to guest to znamená neprihlásený používateľ, user to znamená prihlásený používateľ a v neposlednom rade admin.

Manažment anotácii dovoľuje anotácie pridávať, mazať a spájať ich. Manažment datasetov dovoľuje dataset pridať, premenovať, pridať mu popis a anotovať jeho stĺpce dostupnými anotáciami týmto stĺpcom je tiež možné pridať popis. Taktiež dovoľuje exportnúť JSON formát týchto anotácii a CSV formát datasetu.

Admin rozhranie nám dovoľuje meniť roly používateľov. Máme možnosť aj pridať CSV dataset. Máme taktiež možnosť aj uploadnúť JSON a CSV súbor a tým naseedovať našu databázu. Mimo tohto sú aj dostupné seedery ktoré nám vytvoria oanotované datasety na vyskúšanie si stránky.

2. Cieľ projektu

Cieľom projektu je využiť vybrané techniky čiernej skrinky pri návrhu testovacích prípadov. Cieľom projektu je ukázať na mojej diplomovej práci ako tieto techniky pomáhajú pri tvorbe testovacích prípadov

V mojom projekte aplikujem techniky čiernej skrinky na tvorbu testovacích prípadov pre prihláseného používateľa a konkrétne pre manažment anotácii a manažment datasetov. Využijem tri techniky a tými technikami sú: state transition testing, decision table testing a error guessing.

3. Testovanie čiernej skrinky

Testovanie čiernej skrinky je metodika testovania softvéru, ktorá sa zameriava na analýzu funkčnosti aplikácie bez toho, aby sa ponárala do jej vnútorného kódu alebo štruktúry [1]. Testéri vykonávajú testovanie čiernej skrinky skúmaním vonkajšieho správania softvéru a hodnotením jeho funkčnosti na základe vopred definovaných požiadaviek a špecifikácií. Tento prístup testovania zabezpečuje, že softvér spĺňa očakávané správanie a poskytuje

požadované výstupy pre rôzne vstupy bez toho, aby sa vyžadovala znalosť vnútorného fungovania systému [1].

Testovanie čiernej skrinky je známe aj ako funkčné testovanie, testovanie uzavretej skrinky alebo testovanie nepriehľadnej skrinky [1]. Zaoberá sa predovšetkým vstupno-výstupným vzťahom aplikácie a tester neberie do úvahy základné správanie aplikácie [1]. Táto metodika testovania je vhodná pre testovacie činnosti vyššej úrovne vrátane akceptačného a systémového testovania [4]. Testovací tím nepotrebuje na vykonanie tejto metodiky testovania hlboké znalosti programovania alebo implementácie [4].

Testovanie čiernej skrinky možno použiť na rôznych úrovniach testovania softvéru vrátane unit integration, system a acceptance testingu [3]. Je to základná metodika na zabezpečenie toho, aby softvér spĺňal požiadavky a špecifikácie z pohľadu používateľa [4].

Medzi kľúčové vlastnosti testovania čiernej skrinky patria:

- Skúma funkčnosť softvéru bez zohľadnenia jeho vnútorného fungovania alebo štruktúry.
- Je to neinvazívna metodika testovania, ktorá nevyžaduje znalosť vnútorného kódu alebo konštrukcie.
- Zameriava sa na vstupno-výstupný vzťah aplikácie.
- Používa sa na testovanie softvéru na rôznych úrovniach cyklu vývoja softvéru.
- Je vhodná na testovacie činnosti vyššej úrovne vrátane akceptačného a systémového testovania.
- Nevyžaduje si hlboké znalosti programovania alebo implementácie.
- Dá sa automatizovať, ale nie vždy je to jednoduché.

Testovanie čiernej skrinky je základnou metodikou testovania softvéru a používa sa v rôznych scenároch testovania vrátane funkčného testovania, testovania bezpečnosti a testovania výkonnosti [2]. Často sa používa v spojení s inými metodikami testovania, ako je napríklad testovanie bielej skrinky, aby sa zabezpečilo, že softvér je dôkladne otestovaný a spĺňa požadované normy kvality.

4. Využitie techník čiernej skrinky pri tvorbe testovacích prípadov.

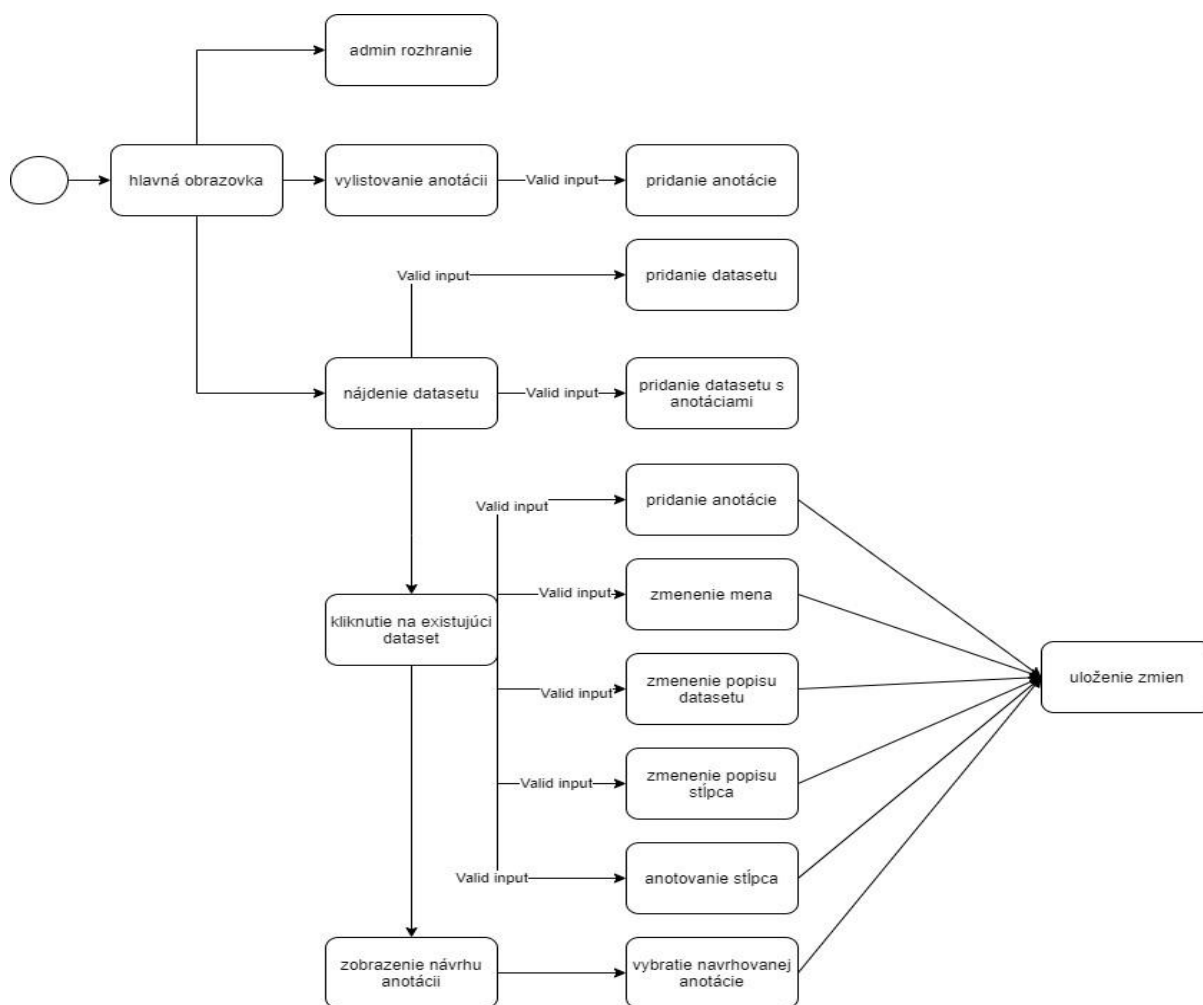
V tejto sekcii si prejdeme mnou vybrané techniky čiernej skrinky ktoré aplikujem na moju diplomovú prácu. Boli vybraté tri techniky ktoré som vyhodnotil ako najvhodnejšie techniky pre môj projekt Výsledné testovacie prípady sú k dispozícii v súbore test-cases.docx. Testovacie prípady boli navrhnuté pre prihláseného používateľa teda typ používateľa user. Dokopy bolo vytvorených 7 testovacích prípadov, 2 použitím techniky state transition testing, 4 použitím techniky decision table testing a 1 použitím techniky error guessing.

4.1 State Transition Testing

Testovanie prechodov medzi stavmi je významnou technikou v oblasti testovania čiernej skrinky, ktorá podrobne skúma dynamiku systému a jeho schopnosť riadne reagovať na rôzne udalosti a situácie. Táto metóda sa zaoberá sledovaním prechodov medzi rôznymi stavmi systému a overuje, či tieto prechody sú vykonávané správne a spoľahlivo. Jednou z hlavných výhod tejto techniky je jej schopnosť zabezpečiť spoľahlivosť softvéru testovaním zmien stavu vyvolaných udalosťami, čo je kľúčové pre správne fungovanie systému v reálnych podmienkach [5].

Pri testovaní prechodov medzi stavmi sa využíva modelovanie správania systému pomocou konceptov ako stavy, udalosti, akcie a prechody medzi nimi. Tento prístup je najefektívnejší pre malé a konečné systémy, kde je možné jednoducho definovať všetky možné stavy a ich prechody. Avšak, pre veľké a zložité systémy môže byť tento prístup obmedzený, pretože modelovanie všetkých možných stavov a prechodov môže byť časovo náročné [6].

Pomocou programu draw.io som vytvoril nasledujúci diagram so stavmi a prechodmi pre používateľa.



Na základe tohto diagramu vidíme cesty ktorými môže používateľ prejsť. Identifikovali sme možné stavy a prechody. Vieme vytvoriť testovacie prípady na overenie či program vykonáva platné prechody medzi stavmi. Vytvoril som testovací prípad (ID: 1 v súbore test-cases.docx) na otestovanie pridania anotácie a testovací prípad (ID: 2 v súbore test-cases.docx) na otestovanie zmeny názvu datasetu.

4.2 Decision Table Testing

Testovanie rozhodovacích tabuliek je jednou z technik testovania čiernej skrinky. Využíva tabuľkový formát na prezentáciu rôznych kombinácií vstupov a ich príslušných reakcií zo strany systému. Táto metóda umožňuje dôkladné pokrytie všetkých možných scenárov, čo vedie k optimalizácii testovacích prípadov a zlepšeniu sledovateľnosti testovacieho procesu. Vďaka identifikácii podmienok a akcií môžu tester systematicky vytvárať rozhodovacie tabuľky, ktoré generujú testovacie prípady podľa stanovených kritérií. Tento prístup sa osvedčuje najmä pri testovaní komplexných obchodných pravidiel, kde zabezpečuje rozsiahle pokrytie testov a efektívne odhaľuje možné chyby a nedostatky. Takto aplikovaná metóda testovania rozhodovacích tabuliek predstavuje cenný nástroj pre zabezpečenie kvality softvérových systémov a ich bezproblémový chod v reálnom prostredí [7].

V tabuľke nižšie môžeme vidieť podmienky, pravidla a očakávané výstupy. Podľa tejto tabuľky som zostavil 4 use casey ktoré slúžia na celkové otestovanie funkcionality nahrávania súborov v mojej diplomovej práci. Tieto use casey sú otestované testovacími prípadmi (ID: 3,4,5,6 v súbore test-cases.docx)

Podmienky	Pravidlo 1	Pravidlo 2	Pravidlo 3	Pravidlo 4
Je prvý súbor v CSV formáte	nepravda	nepravda	pravda	pravda
Je druhý súbor v JSON formáte	nepravda	pravda	nepravda	pravda
Výstupy				
Boli súbory nahrané ?	nie	nie	nie	áno

Na obrázku nižšie môžeme vidieť ako vyzerá pop up pridania datasetu a jeho anotácii ktorí sme testovali pomocou rozhodovacej tabuľky vyššie.

Upload dataset and annotations

Choose CSV file

Choose File No file chosen

Choose JSON file

Choose File No file chosen

Upload dataset and its annotations

4.3 Error Guessing

Odhadovanie chýb je jednou z techník testovania čiernej skrinky. Opiera sa o skúsenosti testera a jeho schopnosť intuitívne odhadnúť miesta, kde sa v softvéri môžu vyskytnúť chyby. Táto metóda sa osvedčuje najmä v prípadoch, keď formálne metódy testovania nedokážu poskytnúť dostatočné pokrytie alebo keď je potrebné identifikovať chyby, ktoré by inak mohli zostať nepovšimnuté. Hoci odhadovanie chýb môže poskytnúť cenné informácie a odhaliť skryté problémy, nie je bezpodmienečne zaručené, že zaistí úplné pokrytie alebo dosiahne najvyššie kritériá kvality. Preto sa odporúča kombinovať túto techniku s ďalšími metódami testovania a analyzovať výsledky s opatrnosťou a kritickým pohľadom [8].

Na základe môjho odhadu som čakal, že keď zvolím rovnaké anotácie a skúsim medzi nimi vytvoriť vzťah nastane chyba. Pre toto bol vytvorení testovací prípad (ID: 7 v súbore test-cases.docx). Na obrázku nižšie môžeme vidieť pop up pridania anotácie v ktorom sú rôzne možnosti ako, pridanie anotácie, zmazanie anotácie a spojenie anotácie. Tento test case sa zaoberal spojením rovnakých anotácií pričom výsledok môžeme vidieť v notifikácii nižšie to znamená test case bol úspešný a bolo otestované, že nemôžem vytvoriť vzťah medzi rovnakou anotáciou.

Add annotation

Name

Enter name of annotation

Shortcut

Enter the shortcut of an annotation

Description

Enter the description of an annotation

Specify the related annotations for this annotation (parent).

Select annotation

Delete annotation

Select annotation to view it

Add annotation

Specify the related annotations for this annotation (child).

Select annotation

Connect annotation

Select annotation to view it

Annotations are the same

You can not connect the same annotation to itself

5. Zhodnotenie

Po dôkladnom preskúmaní výsledkov nášho projektu sme spokojní s tým, že sme úspešne dosiahli ciele, ktoré sme si stanovili. Implementovali sme rôzne techniky čiernej skrinky pri návrhu testovacích prípadov, čím sme zabezpečili komplexné pokrytie funkcionality našej aplikácie. Konkrétne sme využili techniky ako state transition testing, decision table testing a error guessing, ktoré nám poskytli rôzne pohľady na testovanie a pomohli nám identifikovať kritické oblasti, ktoré je potrebné otestovať.

Technika State transition testing nám umožnila efektívne reprezentovať stavy a prechody medzi nimi v našej aplikácii, čo bolo kľúčové pri vytváraní detailných testovacích prípadov.

Technika decision table testing nám poslúžila na systematické vytvorenie tabuľky rozhodovania, ktorá nám pomohla identifikovať všetky potrebné stavy a ich kombinácie, ktoré je potrebné otestovať.

Technika error guessing nám poskytla flexibilitu a umožnila nám hlbšie sa zamyslieť nad možnými problémovými oblasťami a vytvoriť pre ne adekvátne testovacie prípady.

Je však dôležité poznamenať, že aj keď techniky čiernej skrinky majú svoje výhody, existujú aj určité nevýhody. Jednou z hlavných nevýhod je obmedzenie na externú funkčnosť aplikácie a nedostatok pohľadu na vnútornú štruktúru a logiku programu. Preto je vhodné kombinovať rôzne techniky testovania, ako je black box testing a white box testing. Kombinácia týchto prístupov umožňuje získať komplexnejší obraz o kvalite a fungovaní softvéru, pričom zohľadňuje nielen vonkajšie aspekty, ale aj interné procesy a logiku aplikácie. Takto zabezpečujeme efektívne pokrytie testov a odhaľovanie rôznych typov chýb a nedostatkov.

Zoznam zdrojov

- [1] Advantages and disadvantages of black-box testing - javatpoint. www.javatpoint.com. (n.d.-a). <https://www.javatpoint.com/advantages-and-disadvantages-of-black-box-testing>
- [2] Koundinya. (2010, June 25). Black box testing, its advantages and disadvantages. CodeProject. <https://www.codeproject.com/Articles/5579/Black-Box-Testing-Its-Advantages-and-Disadvantages>
- [3] GfG. (2024, March 14). Black box testing - software engineering. GeeksforGeeks. <https://www.geeksforgeeks.org/software-engineering-black-box-testing/>
- [4] Phanivedala. (2023, August 29). Black Box Testing : Exploring types and benefits. Learning Center. <https://www.extnoc.com/learn/security/black-box-testing>
- [5] Thomas Hamilton, & Hamilton, T. (2024, April 3). State transition testing – diagram & technique (example). Guru99. <https://www.guru99.com/state-transition-testing.html>
- [6] GfG. (2024a, January 9). State transition testing. GeeksforGeeks. <https://www.geeksforgeeks.org/state-transition-testing/>
- [7] Sharma, L. (2024, April 7). Decision table testing - A black box testing technique. TOOLSQA. <https://toolsqa.com/software-testing/istqb/decision-table-testing/>
- [8] What is error guessing technique?. Software Testing Help - FREE IT Courses and Business Software/Service Reviews. (2024, March 9). <https://www.softwaretestinghelp.com/error-guessing-technique/>