

Проектна Задача по предметот Софтверски Квалитет и Тестирање

Selenium и Мутациско тестирање на Spring Web Апликација

Виктор Мегленовски 191001

GitHub репозиториум од проектот: <https://github.com/viktor-meglenovski/skit-proekt-1911001>

1. Вовед

Целта на овој проект е да се тестира интерфејсот на веб апликација со помош на Selenium, како и да се тестира сервисната логика на апликацијата со помош на мутациско тестирање. Апликацијата која што се тестира е „ПрваПомош++“ која што самостојно ја изработив во рамки на предметот Дизајн на интеракцијата човек-компјутер.

ПрваПомош++ претставува Spring MVC веб апликација изработена со помош на: HTML, CSS, Bootstrap, JavaScript, AJAX, jQuery и BootBox.

Целиот изворен код од апликацијата е достапен на: <https://github.com/viktor-meglenovski/dnick-hifi-prototype>.

Апликацијата е хостирана на Heroku и е достапна на следниот линк: <https://prva-pomosh-plus.herokuapp.com>.

2. Краток опис на апликацијата

ПрваПомош++ претставува веб базиран систем кој ќе му овозможува на корисниците на интуитивен начин да ги научат сите потребни детали и техники потребни за давање на прва помош. Давањето на прва помош е вештина која секој човек е потребно да ја има но за жал во нашето општество свесноста и заинтересираноста за оваа тема се на ниско ниво.

Системот ќе биде користен од хетерогена група на различни типови на луѓе меѓу кои: млади тинејџери, возрасни лица и постари лица кои без потешкотии можат да ракуваат со паметен телефон или компјутер. Специфична подгрупа на корисници се лицата кои се спремаат да полагаат испит за прва помош кој им е потребен за различни цели како: полагање за возачка дозвола, добивање на сертификат за безбедност на работно место и слично.

Системот има два главни дела: дел за учење, преку кои корисниците преку видео и текст материјали можат да ги научат најважните работи поврзани со темата која ја одбрале, и дел за тестирање, каде корисниците можат да го проверат своето знаење преку решавање на тест од прашања кои што се паѓаат на самите испити за прва помош.

За секој корисник се зачувуваат неговите достигнувања и со тек на време системот според перформансите кои ги покажал корисникот му доделува сертификати во форма на беџови кои се прикажуваат на профилот на корисникот.

3. Мутациско тестирање на сервисната логика

Апликацијата содржи вкупно 6 сервиси и сите тие беа покриени со мутациско тестирање. При процесот, искористени беа: JUnit 5, Mockito и Pitest.

Самите тестови беа пишувани со помош на JUnit, додека пак потребните зависности кон репозиториумите (како и кон сервиси) беа инјектирани како мок објекти со помош на Mockito. На крај, тестовите се извршуваа со соодветна Pitest конфигурација, при што како резултат се добива извештај во кој имаме информации за тоа колку линии код сме покриле, колку мутанти сме убиле, кои мутанти сме ги убиле и слично.

На крајот од овој процес, се достигна 100% покривање на кодот (131 линија код) и 100% убивање на мутантите (72 мутанти) од сите 6 сервиси. Вкупно во овој дел беа напишани 34 тестови.

Pit Test Coverage Report

Package Summary

prototype.hifi.dnick.service.impl

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
6	100% 131/131	100% 72/72	100% 72/72

Breakdown by Class

Name	Line Coverage	Mutation Coverage	Test Strength
AuthServiceImpl.java	100% 7/7	100% 5/5	100% 5/5
BadgeServiceImpl.java	100% 42/42	100% 31/31	100% 31/31
TestResultServiceImpl.java	100% 11/11	100% 11/11	100% 11/11
TestServiceImpl.java	100% 25/25	100% 9/9	100% 9/9
TopicServiceImpl.java	100% 27/27	100% 6/6	100% 6/6
UserServiceImpl.java	100% 19/19	100% 10/10	100% 10/10

Слика 1. Финалниот извештај од Pitest

Во продолжение следува краток опис за секој од сервисите.

3.1. UserServiceImpl

Овој сервис е задолжен за регистрирање на нови корисници и пребарување на корисници според нивното корисничко име. Содржи 3 функции, една за регистрација и 2 за вчитување на корисник според корисничко име (притоа едната функција враќа User а другата UserDetails). Има четири зависности: UserRepository, PasswordEncoder, TopicService, TestResultService.

3.2. AuthServiceImpl

Овој сервис е задолжен за најава на корисници и содржи само еден единствен метод login, кој пробува да најави корисник преку соодветното корисничко име и пасворд што се испраќаат како параметри. Има зависност само кон UserRepository.

3.3. TopicServiceImpl

Овој сервис е задолжен за менаџирање со темите за учење во апликацијата. Содржи методи за креирање на нови теми, пребарување на теми според ИД, земање на сите теми, добивање на информации за сите теми во однос на даден корисник, маркирање на тема како завршена/незавршена во однос на даден корисник и слично. Има зависности кон TopicRepository и TopicUserRepository.

3.4. TestServiceImpl

Овој сервис е задолжен за генерирање на тест прашања и проверка на поени на даден одговорен тест. Има зависности кон: QuestionRepository, TestResultService, BadgeService.

3.5. TestResultServiceImpl

Овој сервис е задолжен за зачувување на нови тест резултати, за проверка на број на тестови, иницијализација на првични поени како и земање на трите најдобри тест резултати во однос на даден корисник. Има зависност само кон TestResultRepository.

3.6. BadgeServiceImpl

Овој сервис ја содржи најкомплексната логика во целата апликација и е наменет за менаџирање со добивање на беџови за одредени достигнувања на корисникот. Во системот има вкупно 5 вакви предефинирани беџови кои се засновани врз бројот на направени тестови и бројот на освоени поени.

Овој сервис содржи пет методи за добивање на сите пет различни беџови, еден метод за добивање на сите пет беџови во една листа, пет методи за проверка дали даден тест резултат го задоволува условот за добивање на секој беџ поединечно, и еден метод кој ги повикува сите овие пет методи за проверка. Содржи зависности кон: UserRepository, BadgeRepository, TestResultService.

4. Тестирање на интерфејсот на апликацијата со Selenium

Целиот интерфејс на апликацијата беше истестиран со помош на Selenium 4.0.0 и ChromeDriver. Опфатени беа сите можни типови на линкови, пренасочувања, форми и повици. Опфатени беа и таканаречените „среќни патишта“, но и патиштата што се проблематични.

За секоја страница беа напишани тестови за секој линк, секоја акција посебно (добри и лоши патишта). Па така, за страницата за најава, покрај тест за успешна најава, вклучени беа и тестови каде: не се внесуваат аргументи, каде се внесуваат невалидни аргументи, како и тест преку кој се пренасочуваме на страницата за регистрација. Истиот овој принцип беше искористен низ целиот процес на тестирање.

Опфатени беа сите страници (вкупно 8): Home, Login, Register, My Profile, Learn, Topic, Test, Results and Badges. Дополнително беше истестирано и едно целосно сценарио од регистрација на нов корисник, па најава, исполнување на задачи кои ги нуди системот, правење на тестови, проверка на резултати па се до одјава на корисникот. Вкупно во овој дел беа напишани 35 тестови.

5. Заклучок

Процесот на тестирање на оваа апликација се покажа како многу успешен и корисен бидејќи помогна да се отстранат одредени слабости и нелогичности во самиот дизајн на апликацијата, како и да се направат одредени ситни измени со цел да се зголеми доверливоста и квалитетот на самата апликација.

Иако не беше истестирана целата апликација (domain моделите, репозиториумите и самите контролери не беа опфатени), сепак апликацијата сега има значително подобар квалитет од претходно, а воедно и се покажа дека и самиот интерфејс на апликацијата е соодветен и нема значителни пропусти.

Мутациското тестирање помогна да се осигура квалитетот на бизнис логиката на самата апликација преку тестирање на најразлични можности (мутанти) и покажување дека ниту една од тие верзии не е валидна, додека пак тестирањето со Selenium овозможи детално да се провери целиот интерфејс на апликацијата и да се најдат одредени пропусти кои би требало да се променат.