

Testowanie

Testy funkcjonalne

Dla każdego wymagania funkcjonalnego F1. □□– F10. (znajdującego się w dokumencie „Wymaganie funkcjonalne”, a opisanego w „Przypadkach użycia”) powinien zostać przygotowany test realizowany z poziomu interfejsu użytkownika obu klientów: webowego i desktopowego (poziom: *end-to-end*). Testy powinny być automatyczne i wykonywane w środowisku złożonym ze wszystkich elementów systemu, aczkolwiek sztucznym (z testową bazą danych przywracaną do wyjściowego stanu dla każdego testu). Testy nie powinny zależeć od kodu systemu (z wyjątkiem przygotowania testowej bazy danych) ani sprawdzać jego wewnętrznego stanu (muszą bazować jedynie na informacji dostępnej z poziomu klienta). Testy powinny być przygotowane jako program testujący, uruchamiający, w ramach każdego testu, serwer z wyjściową testową bazą danych.

Klient desktopowy

Zgodnie z wymaganiem N3. testy powinny być przeprowadzane w systemie Linux ze środowiskiem graficznym obsługującym Qt5. Program testowy powinien używać narzędzia **Linux Mouse and Keyboard Automation Software Tools (murguu.com)**, pozwalającego na symulowanie zaplanowanych interakcji użytkownika (kliknięć myszą, wprowadzania danych z klawiatury itp.). Odczytywanie danych z aplikacji powinno odbywać się przez pobieranie w odpowiednich chwilach zrzutu zawartości okna aplikacji przez program testujący i porównywanie elementów pobranego obrazu z przygotowanymi ręcznie, graficznymi wzorami. Minusem tego podejścia jest konieczność testowania w jednym środowisku graficznym (stała rozdzielczość itp.) oraz ręczne przygotowywanie graficznych wzorów do porównywania po zmianie wyglądu aplikacji. Ponadto przy zmianach związanych z czasem wyświetlania poszczególnych elementów aplikacji, konieczne jest poprawienie konfiguracji opóźnień symulacji interakcji użytkownika i odczytywania danych z okna aplikacji.

Klient webowy

Zgodnie z wymaganiem N4. testy powinny być przeprowadzane w przeglądarce obsługującej JavaScript w wersji 6. Testy powinny być przygotowane za pomocą narzędzia **Selenium** pozwalającego na naprzemienne symulowanie interakcji użytkownika oraz pobieranie danych z elementów HTML znajdujących się na stronie aplikacji wyświetlanej w przeglądarce. Symulowanie interakcji i odczytywanie danych powinno uwzględniać dopuszczalne rozbieżności w czasie (mieć zdefiniowane limity czasowe na poszczególne kroki testu). Testy powinny być uruchamiane z konsoli SBT.

Minusem tego rozwiązania jest konieczność aktualizacji testów po zmianie treści lub hierarchii wyświetlanych komponentów HTML.

Testy wydajności

Testy powinny być przeprowadzone na przygotowanej testowej bazie danych jako rozproszony system programów testujących, z których jeden uruchamia serwer, a pozostałe symulują klientów łączących się z serwerem i wykonujących standardowe akcje klienta. Liczba symulowanych klientów, częstotliwość interakcji i ich rodzaj powinny być nieznacznie bardziej wymagające niż maksymalne wartości obciążenia, zdefiniowane w dokumencie „Wymagania niefunkcjonalne”, z którym musi pracować serwer.

Testy bezpieczeństwa

Testy z poziomu klienta webowego

Test sprawdzający czy system jest odporny na Cross-Site Request Forgery (CSRF).

Testy serwera

- Testy polegające na interakcji serwera z fałszywymi klientami, próbującymi uzyskać dostęp do części systemu, do których nie posiadają uprawnień.
- Testy polegające na wysyłaniu danych do serwera mogących spowodować jego niepoprawne funkcjonowanie.

Testy instalacji

Testy instalacji klientów: desktopowego i webowego powinny być przeprowadzane ręcznie za każdym razem, gdy ma być wydana nowa wersja danego klienta ze zmienionym którymkolwiek z poniższych elementów:

- zależności środowiskowe/biblioteczne,
- konfiguracja uruchomienia aplikacji klienta.

Klient desktopowy

Zgodnie z wymaganiem N3. test polega na uruchomieniu pliku wykonywalnego klienta desktopowego w systemie Linux ze środowiskiem graficznym obsługującym Qt5:

- uruchomienie 64-bitowej wersji klienta na 64-bitowej wersji systemu operacyjnego
- uruchomienie 32-bitowej wersji klienta na obu wersjach systemu operacyjnego (32- i 64-bitowej)

Klient webowy

Zgodnie z wymaganiem N34. test polega na uruchomieniu klienta webowego w przeglądarkach: Firefox, Google Chrome obsługujących JavaScript w wersji 6.

Serwer

Przed każdą instalacją nowej wersji (aktualizacji) serwera na maszynie obsługującej system produkcyjny, powinien zostać przeprowadzony test polegający na aktualizacji serwera na testowej maszynie, na której znajduje się stara wersja oprogramowania (identyczna z tą na maszynie produkcyjnej, ale posiadająca testową bazę danych. Testowa baza danych powinna być reprezentatywną próbką produkcyjnej bazy danych). Test powinien sprawdzać poprawność migracji danych systemu do nowej wersji serwera oraz działanie uruchomionej, nowej wersji systemu. Testowa migracja danych (tak samo jak produkcyjny odpowiednik) powinna mieć charakter transakcyjny.

Testy integracyjne

Klient desktopowy

Test powinien polegać na uruchomieniu programu symulującego serwer (atrapy) oraz oryginalnego klienta desktopowego, który łączy się z atrapą serwera. Celem testu jest zweryfikowanie poprawności komunikacji klienta desktopowego.

Klient webowy

Test powinien polegać na uruchomieniu programu symulującego serwer (atrapy) oraz oryginalnego klienta webowego, który łączy się z atrapą serwera. Celem testu jest zweryfikowanie poprawności komunikacji klienta webowego.

Serwer (jako całość)

Test powinien polegać na uruchomieniu serwera w środowisku testowym oraz programów symulujących klientów łączących się z serwerem. Celem testu jest zweryfikowanie poprawności komunikacji serwera.

Aktorzy serwera (wewnętrzni modułu *server* oraz *GamePlayActor* z modułu *game* i *BotActor* z modułu *bot*)

Testy przygotowane za pomocą *ScalaTest* i modułu testowego frameworku *Akka* uruchamiane z konsoli *SBT*. Każdy test ma za zadanie weryfikować poprawność komunikacji jednego aktora.

Test rozwiązania problemu niekompatybilności wersji klientów i serwera

Test polegający na uruchomieniu starszych wersji klientów desktopowych, niekompatybilnych z równolegle uruchomioną najnowszą wersją serwera. Celem testu jest sprawdzenie obsługi takiego przypadku zarówno przez serwer jak i klientów.

Analogiczny test w przypadku klienta webowego.

Testy jednostkowe

W każdy module systemu, każdy znaczący kawałek kodu powinien posiadać zautomatyzowane testy jednostkowe sprawdzające zgodność jego działania z opisem dokumentacji.

Testy regresyjne

Dla każdego wykrycia błędu w systemie powinien powstać odpowiedni test, którego scenariusz daje możliwość zaobserwowania danego błędu. Taki test powinien mieć odpowiedni poziom (jednostkowy, integracyjny, end-to-end), na którym da się zaobserwować dany błąd. Z uwagi na konieczność automatyzacji testowania regresyjnego, o ile to możliwe, preferowane są testy niższego poziomu. Każdy taki test powinien być dodany do puli regularnie wykonywanych testów.