Dokumentacja projektu

Gherkin Scenario Generator

1 Ogólna informacja o projekcie

Przedmiotem opisanym w poniższej dokumentacji jest projekt aplikacji desktopowej przeznaczonej na system Windows, mający za zadanie wspierać tworzenie scenariuszy testowych w języku GHERKIN.

1. 1 Opis problemu

W Cyklu życia każdej aplikacji bardzo ważny element stanowi etap testowania oprogramowania. W zależności od sposobu realizacji i zastosowanych metodyk, testowanie może odbywać się równolegle z etapami tworzenia i planowania aplikacji lub tuż po zakończeniu etapu implementacji. W każdym z wyżej wymienionych przypadków, zespół testerski do swojej pracy potrzebuje tzw. Historyjek Użytkownika, które opisują funkcjonalności pokryte w aplikacji. Stanowi to jednocześnie listę rzeczy do przetestowania.

Są różne podejścia do przygotowywania dokumentacji testowej. Dawniej, gdy metodyki zwinne dopiero zyskiwały na popularności tworzono plany testowe oraz liczną dokumentację tj. Scenariusze, czy przypadki testowe wykorzystując narzędzia takie jak arkusz kalkulacyjny, czy edytor do tworzenia dokumentów tekstowych.
Plan testów jest odpowiedzialny za opisywanie metody testowania, zakres testów do przeprowadzenia oraz spisywał czynności wymagane do przeprowadzenia podczas etapu testów.
Scenariusze testowe skupiały się na przetestowania konkretnej funkcjonalności zaimplementowanej w testowanej aplikacji.
Na najniższym poziomie znajdowały się Przypadki Testowe, które znajdowały się pod odpowiednim Scenariuszem testowym i opisywały możliwie najwięcej sposobów przetestowania konkretnej funkcjonalności.

Zakładając, że funkcjonalnością do przetestowania jest realizacja zamówienia,
Plan testów przewidywał by metodykę testowania danego fragmentu aplikacji,
Scenariusz testowy ogólnie opisywałby przeznaczenie danego elementu, wymagania co do niego, oraz gromadziłby przypadki testowe, które zostaną sprawdzone.

Może to być np.

- Próba realizacji zamówienia dla jednego produktu
- Próba realizacji zamówienia dla wielu produktów
- Próba wykorzystania różnych metod płatności
- Próba wykorzystania różnych metod dostawy
- Tworzenia zamówienia dla zalogowanego użytkownika
- Tworzenie zamówienie dla gościa itd.

Następnie wyniki testów zostałyby zapisane w scenariuszu testowym wraz z podsumowanie, czy rezultat jest zgodny z wymaganiami oraz datą realizacji testów.
Przy kolejnych zmianach wymagane jest utworzenie nowego dokumentu, który opisuje testy dla aktualnej wersji aplikacji. Takich dokumentów może być bardzo dużo podczas trwania projektu, co jasno pokazuje jak duży wysiłek związany jest z manualnym testowaniem oprogramowania.

1.2 Możliwe rozwiązanie problemu

Naprzeciw problemowi manualnego testowania regresji (elementów aplikacji, nad którymi praca została zakończona), co jest wymagane, aby zapewnić, że praca w innych częściach aplikacji lub wprowadzanie poprawek nie uszkodziło istniejących elementów, wyszła metodyka TDD, a później BDD.
Metodyki te skupiają się na bliskiej współpracy całego zespołu technicznego oraz klienta oraz testowaniu aplikacji na bieżąco, w małych porcjach, ale regularnie.
Dodatkowo zaczęto używać języka GHERKIN jako sposobu realizacji scenariuszy testowych.

Język GHERKIN ma wiele zalet. Po pierwsze ma składnie zrozumiałą dla człowieka, co ułatwia zrozumienie funkcjonalności osobom niezwiązanych ze światem technicznym, Dodatkowo język GHERKIN daje się interpretować przy użyciu różnych silników do rozpoznawania języka naturalnego tj. cucumber, behat, dzięki czemu możliwe jest zaimplementowanie takich scenariuszy i na podstawie wygenerowanych kroków stworzenie w dość prosty sposób skryptów testujących.
Tworzenie skryptów testujących pozwala zaoszczędzić dużo czasu oraz środków i zapewnić większą wiedzę na temat stanu aplikacji. Nie potrzebujemy w takim przypadku przeprowadzać testów regresyjnych manualnie, możemy wykorzystać do tego celu skrypty. Dzięki czemu zaoszczędzimy ogromne ilości czasu zespołowi testerskiemu oraz pozwolimy programistom mieć większą kontrolę nad błędami, które popełniają.

1.2.1 Przykładowy scenariusz w języku GHERKIN

```
Given Użytkownik jest na stronie głównej
And Użytkownik jest zalogowany
And Użytkownik dodał produkt do koszyka
When Użytkownik kliknie przycisk stwórz zamówienie
Then Użytkownik zostanie przekierowany do koszyka
```

1.2.2 Przykładowa interpretacja kroków ze scenariusza w języku Java

```
@Given("^Użytkownik jest na stronie głównej$"){
    throw new PendingException();
}

@Given("^Użytkownik jest zalogowany$"){
    throw new PendingException();
}

@Given("^Użytkownik dodał produkt do koszyka$"){
    throw new PendingException();
}

@When("^Użytkownik kliknie przycisk stwórz zamówienie$"){
    throw new PendingException();
}
@Then("^ Użytkownik zostanie przekierowany do koszyka$"){
    throw new PendingException();
}
```

Tworzenie scenariuszy w języku GHERKIN podobnie jak tworzenie scenariuszy z wykorzystaniem arkusza kalkulacyjnego, czy edytora dokumentów tekstowych wymaga manualnej pracy.
Jest to czasochłonne zajęcie obarczone dużym ryzykiem, ponieważ przy generowaniu szablonów do testów automatycznych na podstawie scenariuszy w języku GHERKIN każda różnica w kroku jest interpretowana jako zupełnie oddzielny i niezależny krok.
Jeżeli osoba przygotowująca scenariusz popełni błąd i w kroku pojawi się np. Literówka lub przez nieuwagę zostanie stworzona druga wersja kroku opisującego tę samą funkcjonalność, silnik interpretujący język naturalny wygeneruje kilka zupełnie niezależnych kroków, co prowadzi do niespójności i wprowadza nieporządek.

Np. kroki

- Użytkownik jest zalogowany
- użytkownik jest zalogowany
- Użytkownik zalogował się

Opisują tę samą funkcjonalność, ale przez generator zostaną zinterpretowane jako zupełnie różne. Jest to spowodowane różnicami w składni kroku.

1.3 Wymagania funkcjonalne

Celem aplikacji opisanej w poniższej dokumentacji jest wspieranie zespołu testerskiego w procesie przygotowywania scenariuszy testowych, które mogą zostać wykorzystane zarówno w testach manualnych jaki i automatycznych.

Program z wykorzystaniem interfejsu graficznego w sposób prosty i zrozumiały wesprze użytkownika w procesie tworzenia scenariuszy oraz pomoże mu wygenerować gotowy plik z rozszerzeniem .feature (jest to natywny format języka GHERKIN).

W dalszych etapach rozwoju aplikacji, program będzie w stanie podpowiadać użytkownikowi istniejące kroki, aby uniknąć powtórzeń, oraz będzie w stanie wesprzeć testera w procesie tworzenia tzw. Scenario Outline tj. Scenariuszy testowych zawierających parametry - dane testowe, które muszą być wykorzystane podczas przeprowadzania testów

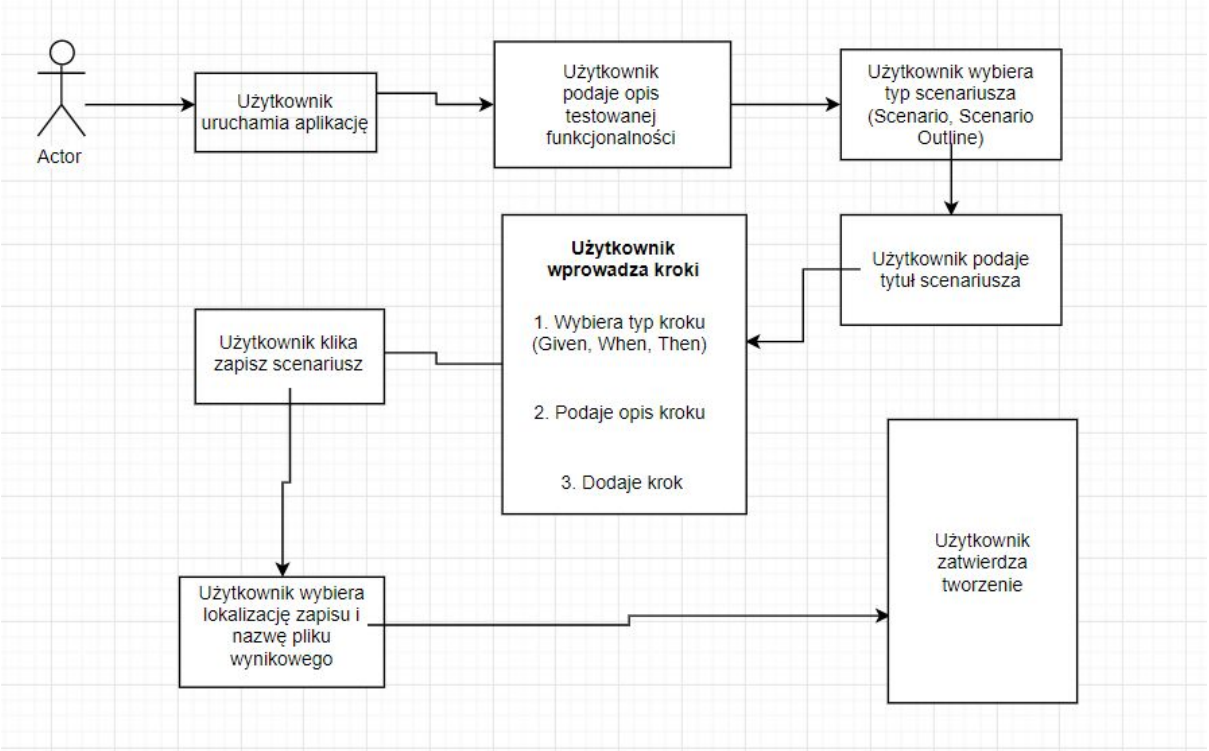
1.4 Wymagania niefunkcjonalne

Ze względu na niską złożoność aplikacji, wymagania sprzętowe nie są wygórowane. Aplikacja powinna być w stanie pracować na standardowych komputerach użytku domowego jak i profesjonalnym sprzęcie programistycznym.
Jedynym ograniczeniem jest kompatybilność z systemem Windows, ponieważ na taki system aplikacja jest tworzona.

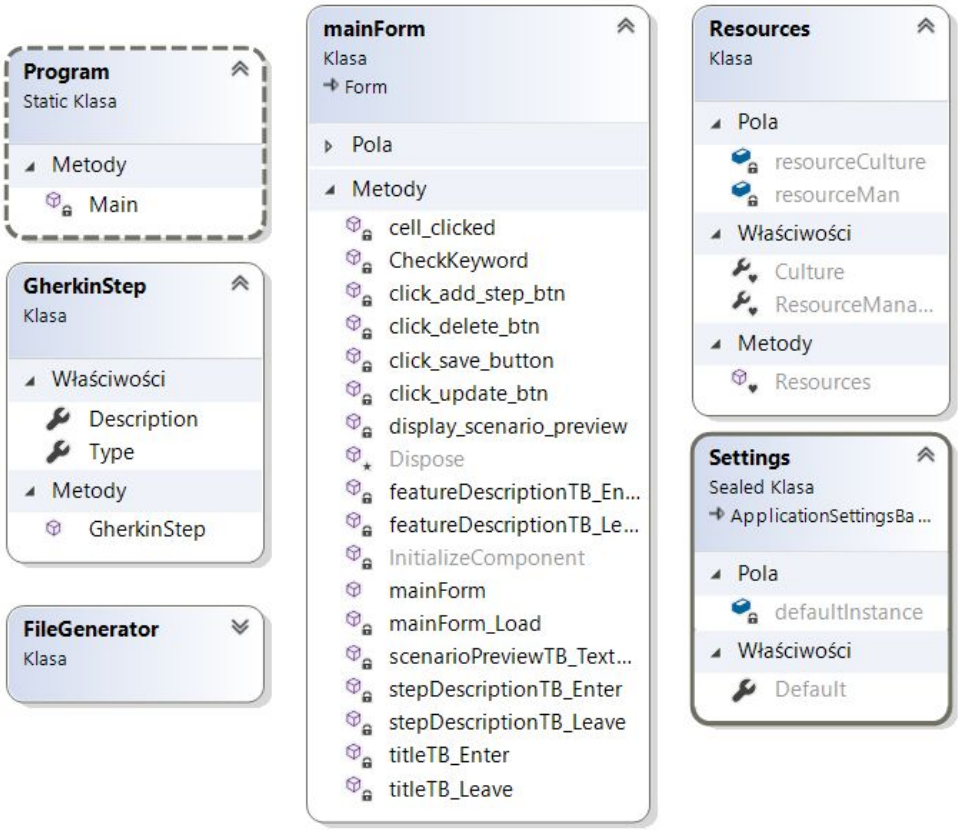
1.5 Opis słowny

1. Użytkownik uruchamia aplikację
2. Użytkownik podaje opis testowanej funkcjonalności (Feature)
3. Użytkownik wybiera typ scenariusza (Scenario, Scenario Outline)
4. Użytkownik podaje tytuł scenariusz
5. Użytkownik wprowadza kroki
    a. Wybiera typ kroku (Given, When, Then)
    b. Podaje opis kroku
    c. Dodaje krok
6. Użytkownik klika zapisz scenariusz
7. Użytkownik wybiera lokalizację zapisu i nazwę pliku wynikowego
8. Użytkownik zatwierdza tworzenie

1.6 Opis graficzny



1.7 Diagram Klas



1.8 Wygląd programu GUI