



WYŻSZA SZKOŁA BANKOWA
w Poznaniu

Maciej Nikonowicz

„Hungry Capybara”
Web Application

PROJEKT PODYPLOMOWY

Promotor: Łukasz Rybak, Bartosz Kaczmarek

Podyplomowe Studia Testowanie Oprogramowania

Poznań, 2018/2019

I. Wstep

1. Wprowadzenie

Niniejszy dokument jest planem testów dla aplikacji webowej „Hungry Capybara”, stworzonej przez niezależny zespół programistów, której grupą docelową są użytkownicy z potrzebą większej kontroli nad produktami w swojej kuchni, chcący wprowadzić w swoim życiu politykę „zero waste”. Główną funkcjonalnością aplikacji jest sugerowanie użytkownikom przepisów z zewnętrznego API na podstawie wskazanych produktów. Aplikacja dostępna jest produkcyjnie w podstawowej wersji i wymaga przetestowania przed nowymi wydaniem.

2. Cele

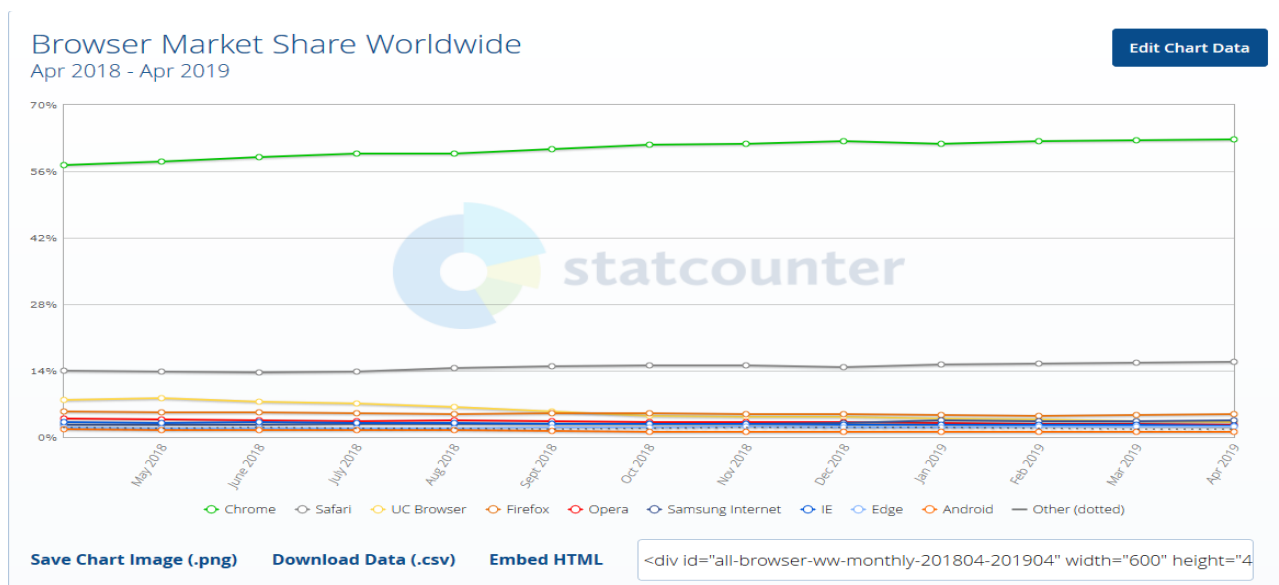
Głównym celem testów jest sprawdzenie, czy poprawnie działają główne funkcjonalności aplikacji: logowanie i rejestracja, komunikacja z zewnętrznym API, generowanie przepisów, dodawanie/usuwanie/edycja produktów oraz składników. Wymagane jest także sprawdzenie jak aplikacja reaguje na różne obciążenie systemu. Warunkiem dalszego rozwoju aplikacji są poprawnie działające wszystkie główne funkcjonalności. Poprawność ich działania została opisana w punkcie „Kryteria zaliczenia/niezaliczenia testów”

3. Oczekiwania

Faza testowania ma dowieść, że wykonany System spełnia wymagania funkcjonalne i niefunkcjonalne, przedłożone przez Zamawiającego (Głównego programistę) i nadaje się do dalszego rozwoju.

4. Podejście

Z racji tego, iż System nie posiada dokumentacji przyjęte zostało podejście testowania eksploracyjnego. Testy manualne wykonane zostaną przez osobny, zewnętrzny zespół testerski. Zostaną wykonane zatem testy automatyczne, wydajności, funkcjonalne(end-to-end). Z powodu ograniczonego budżetu oraz czasu, testy zostaną przeprowadzone wyłącznie na przeglądarce Chrome kompatybilną do wersji 74, która stanowi blisko 65% użycia w sieci.



II. Opis testów

1. Testowany obiekt – opis aplikacji

Obiektem testowanym jest aplikacja webowa znajdująca się pod adresem <http://vol4-capybara.herokuapp.com/>, dostępna wyłącznie w języku angielskim. Po zarejestrowaniu/zalogowaniu użytkownik ma dostęp do strony ze składnikami, które może dodawać/edytować oraz z których generowane są później przepisy z zewnętrznego API

My ingredients

Add a new ingredient

What would you like to eat?

Breakfast

Vegan

#	Name	Expiration date	Quantity	Unit	Checkbox
1	almonds	2019-01-01	2	kg	<input type="checkbox"/>
2	onions	2019-03-02	1	kg	<input type="checkbox"/>
3	milk	2019-01-01	1	litre	<input type="checkbox"/>
4	garlic	2019-01-01	1	kg	<input type="checkbox"/>
5	turkey	2019-03-05	10	kg	<input type="checkbox"/>
6	carrot	2019-01-01	1	kg	<input type="checkbox"/>
7	beans	2019-01-01	1	kg	<input type="checkbox"/>
8	salt	2019-07-18	1	kg	<input type="checkbox"/>

Generate a recipe

- *My ingredients* – tabela ze składnikami widoczna na zrzucie
- *Add a new ingredient* – strona odpowiedzialna za dodanie nowego składnika
- *Breakfast/Vegan* – inputy służące za wybór rodzaju diety i typu posiłku
- *Checkbox* – wybór składników, z których chcemy wygenerować przepisy
- *Generate a recipe* – przycisk do generowania przepisów z wybranych składników

Użytkownik „Admin” ma możliwość dodawania produktów, z których później użytkownicy mogą kompletować listę swoich składników.

All products

Add new product

List of Products:

1. aaa	2. almonds	3. apple
4. avocado	5. bacon	6. banana
7. barbecue	8. basil	9. beans
10. beef	11. bell pepper	12. berries

2. Funkcjonalności do przetestowania

Tabela priorytetów testowanych elementów:

1	Niski priorytet, drobnostka
2	Normalny priorytet
3	Wysoki priorytet, kluczowe

- Rejestracja (2)
- Logowanie (2)
- Dodawanie nowego produktu (tylko użytkownik „admin”) (2)
- Dodawanie/edycja/usuwanie składników (1)
- Generowanie przepisów z wybranych składników (3)
- Poprawność API (3)
- Wydajność (1)

3. Funkcjonalności nietestowane

- Edycja danych użytkownika
- Logowanie za pomocą Facebook
- Zmiana avatara
- Wylogowanie
- Ilość generowanych przepisów
- Negatywne scenariusze, np. błędne dane logowania, produkty z niedozwoloną nazwą itd.
- Funkcja połączenia z aplikacją Instagram

Z racji braku dokumentacji, a także minimalnego budżetu i ograniczeń czasowych zostaną wykonane testy integracyjne sprawdzające tylko główne funkcjonalności systemu niezbędne do jego funkcjonowania i dalszego rozwoju. Nie jest istotnym „co” jest generowane, a „czy” jest generowane, dlatego pominięte zostaną testy negatywne oraz funkcjonalności nie wpływające na całościowe działanie Systemu.

4. Techniki i narzędzia

- a) **Testowanie API** – w celu przetestowania zewnętrznego API użyte zostanie narzędzie SoapUI w wersji 5.5.0. Stworzone zostaną przypadki testowe sprawdzające, czy API zwraca prawidłowe wyniki z odpowiednim statusem HTTP oraz w odpowiednim czasie, zawierające odpowiednie słowa kluczowe.
- b) **Testy automatyczne** – zostaną napisane testy automatyczne przy pomocy Selenium oraz języka Python. Testy automatyczne będą sprawdzać, czy prawidłowo działają główne funkcje Systemu, tj. logowanie, rejestracja, generowanie przepisów – sprawdzenie zostanie wykonane za pomocą testów automatycznych, aby mieć podstawę do późniejszych testów regresyjnych przy kolejnych wydaniach systemu. Kluczowe wyniki zostaną zapisane w formie zrzutów ekranu. Kluczowym jest sprawdzenie, czy podstawowe funkcjonalności

działają poprawnie, czyli zwracają dane i nie generują błędów. Nie sprawdzamy co dokładnie zostało dodane.

- c) **Testy wydajności** – do testów wydajności zostanie użyte narzędzie JMeter. Zostaną stworzone przypadki sprawdzające jak zachowują się główne strony aplikacji przy dużym obciążeniu użytkowników. Wyniki zostaną zalogowane w odpowiednich formatach.

5. Kryteria zaliczenia/niezaliczenia testu

- a) **Logowanie** – kryterium zaliczenia jest poprawne zalogowanie się do Systemu za pomocą dostarczonych danych (dla zwykłego użytkownika oraz użytkownika „admin”), wyświetlenie przycisku „Check your fridge”
- b) **Rejestracja** – kryterium zaliczenia jest poprawna rejestracja nowego użytkownika za pomocą dostarczonych danych (+ poprawny upload pliku do avatara), możliwość zalogowania na utworzonego użytkownika, wyświetlenie przycisku „Check your fridge”
- c) **Dodawanie nowego produktu** – kryterium zaliczenia jest poprawne dodanie nowego produktu przez użytkownika „admin” (wyświetlenie na liście produktów oraz możliwość jego dodania jako składnika – wybór z listy na stronie dodawania)
- d) **Dodawanie/edycja/usuwanie składników** – kryterium zaliczenia jest poprawne dodanie składnika do listy, a następnie możliwość jego edycji (zmiana nazwy/daty/ilości) oraz usunięcia
- e) **Generowanie przepisów** – kryterium zaliczenia jest wyświetlona lista przepisów z zewnętrznego API po kliknięciu przycisku (standardowo 10 pozycji)
- f) **Poprawność API** – kryterium zaliczenia jest pozytywne przejście wszystkich asercji do każdego przypadku testowego (zawartość odpowiedzi, statusy nagłówek, czas odpowiedzi)
- g) **Wydajność** – kryterium zaliczenia jest pozytywne przejście wszystkich asercji do każdego przypadku testowego

Każdy test automatyczny generować będzie zrzut ekranu potwierdzający, czy funkcjonalność spełnia założenia.

Przewidywane utrudnienia, które mogą wystąpić podczas testowania:

- brak dostępu do internetu / spowolnione łącze
- brak dostępu do zewnętrznego API (przykładowo aktualizacje po stronie dostawcy)
- brak dostępu do serwera, na którym działa aplikacja (Heroku)

6. Dokumentacja testu

- a) Plan testu - „test_plan_HC.odt” - niniejszy dokument
- b) Scenariusze testowe – „scenariusze_testowe.ods” - rozpisane przypadki testowe dostarczone przez osobny zespół odpowiedzialny za testy manualne
- c) Dokumentacja do testów automatycznych - „testy_automatyczne_spec.odt”
- d) Opis testów API - „testy_API_spec.odt”
- e) Opis testów wydajnościowych - „testy_wydajności_spec.odt”
- f) Raport podsumowujący - „raport_finalny.odt”
- g) Wszystkie pliki, zarówno dokumenty jak skrypty, dostępne na stronie Github

7. Środowisko testowe

- Komputer z dostępem do sieci
- System operacyjny Windows (od wersji 7 w górę)
- Zainstalowany Python w wersji 3.x
- Zainstalowane Selenium w wersji 3.14
- Zainstalowane SoapUI w wersji 5.5.0
- Zainstalowany JMeter w wersji 5.1.1
- Zainstalowane środowisko Java(TM) SE Runtime Environment
- Przeglądarka Chrome w wersji 74
- Dowolny edytor kodu (preferowany PyCharm 2018.2.4)

8. Osoby odpowiedzialne

Za wyżej wymienioną część testów odpowiedzialna jest jedna osoba, specjalnie do tego wyznaczona, która ma zaplanować, przygotować, wdrożyć i zaraportować wszystkie testy, które zostały opisane we wcześniejszych punktach. Jej zadaniem jest dostarczenie Zamawiającemu informacji, czy główne funkcje Systemu działają i nadają się do dalszego rozwoju aplikacji.

9. Harmonogram

- 08.04 – 20.04 Plan testów
- 20.04 – 05.05 Testy API
- 05.05 – 12.05 Testy wydajnościowe
- 12.05 – 20.05 Testy automatyczne, integracyjne
- 20.05 – 30.05 Czynności zamykające, raportowanie
- 01.06 – dostarczenie dokumentacji do Zamawiającego

10. Historia zmian

Data zmiany	Osoba aktualizująca	Opis zmiany
11.03.2019	MN	Wersja 1.0
20.04.2019	MN	Wersja 1.1
08.05.2019	MN	Wersja 1.2
21.05.2019	MN	Wersja 1.3
28.05.2019	MN	Wersja 1.4
31.05.2019	MN	Wersja 1.5