

Wprowadzenie do zadania projektowego

Projekt: Kapu\$ta

SZCZEPAN SZCZĘCH

Quality Assurance Engineer

I. Cel:

1. Przetestowanie programu Kapusta.
2. Napisanie przypadków testowych/scenariuszy testowych.
3. Zaraportowanie defektów ze wszystkich przypadków testowych i scenariuszy testowych, które dały efekt niezgodny z oczekiwanym.

II. Dostępne zasoby:

- A. Specyfikacja.
- B. Makiety.
- C. Strona URL z opublikowanym programem.

III. Planowane działania:

1. Przeczytaj specyfikację i zanotuj uwagi i pytania.
2. Zapoznaj się z makietami i zanotuj pytania i uwagi, jeśli jakieś nie zostały zawarte w pytaniach do specyfikacji.
3. Wejdź na stronę programu.
4. Opisz co będziesz testować.
5. Stwórz koncepcję kategorii z opisem co chcesz przetestować w każdym module.
6. Napisz test case'y.
7. Wykonaj testy.
8. Zaraportuj defekty.
9. Stwórz listę artefaktów testowych powstałych podczas wykonywania zadania.

ad.1. Specyfikacja z pytaniami i uwagami – zał. 1.

ad. 2. Link do makiet:

<https://www.figma.com/file/m2DLMAhjLHuQOPuIdwnGAI/Kapusta?node-id=19401%3A155&t=dn4Wh7stnmkk3e2t-0>

ad. 3. URL: <https://kapusta-qa.netlify.app/>

ad. 4. Opis programu:

Projekt Kapu\$ta.

Kapu\$ta jest programem do zarządzania budżetem domowym. Kapu\$ta umożliwia wprowadzenie wydatków i wpływów z opisem i w odpowiedniej kategorii oraz ze szczególną datą. Użytkownik może zobaczyć listę swoich transakcji w danym dniu i w podsumowaniu miesiąca. W dziale raporty może zobaczyć podsumowanie transakcji,

a także wykres prezentujący wszystkie transakcje w tej kategorii w danym miesiącu. Dzięki opcji logowania użytkownik może zapisać transakcje na swoim indywidualnym koncie, a w razie pomyłki także je usunąć.

ad. 5. Kategorie do przetestowania:

1. Aspekt wizualny:	a. Czytelność i przejrzystość strony. b. Symetryczność i spójność. c. „Przyjemny dla oka” odbiór strony. d. Bardziej ogół niż szczegół.
2. Autoryzacja Google:	a. Możliwość prowadzenia autoryzacji Google na etapie rejestracji i logowania.
3. Rejestracja:	a. Sposoby rejestracji. b. Zgodność danych użytych w rejestracji ze specyfikacją. c. Wymagalność rejestracji. d. Komunikaty systemowe ich sens i spójność z faktami.
4. Logowanie:	a. Sposoby logowania. b. Zgodność danych do logowania z danymi z rejestracji. c. Zgodność danych do logowania ze specyfikacją. d. Komunikaty systemowe ich sens i spójność z faktami.
5. Nagłówek:	a. Widoczność wszystkich niezbędnych elementów nagłówka. b. Funkcjonalność elementów nagłówka.
6. Strona główna*:	a. Zachowanie strony przy pierwszym użyciu i kolejnych. b. Widoczność wszystkich elementów strony głównej i możliwość ich użycia. c. Dodawanie, usuwanie i modyfikowanie transakcji oraz wpływy tych działań na podsumowania. d. Warunki dodawania transakcji i ich zgodność ze specyfikacją. e. Funkcjonalność kalendarza. f. Tłumaczenia. g. Tryb jasny i ciemny. h. Prezentacja danych.
7. Raporty:	a. Funkcjonalność i wiarygodność sposobu prezentacji danych zawartych w raportach. b. Zgodność z innymi danymi. c. Zakres prezentowanych danych. d. Spójność ze specyfikacją. e. Możliwość edycji.

* Ze względu na zakres funkcji na stronie głównej precyzyjny podział powstanie na etapie pisania test casów.

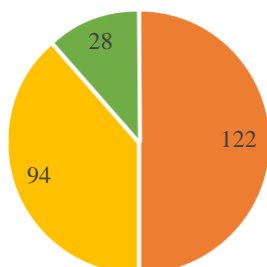
ad. 6. Test casy – załącznik nr 2.

ad. 7. Wykonanie testów.

ad. 8. Lista defektów – załącznik nr 3.

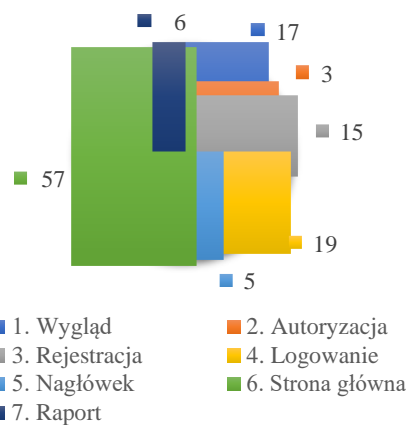
ad. 9. Lista artefaktów testowych:

- a. Opis projektu z podsumowaniem.
- b. Plan pracy.
- c. Test suity.
- d. Test casy.
- e. Zestaw danych testowych – załącznik nr 4.
- f. Raport defektów.

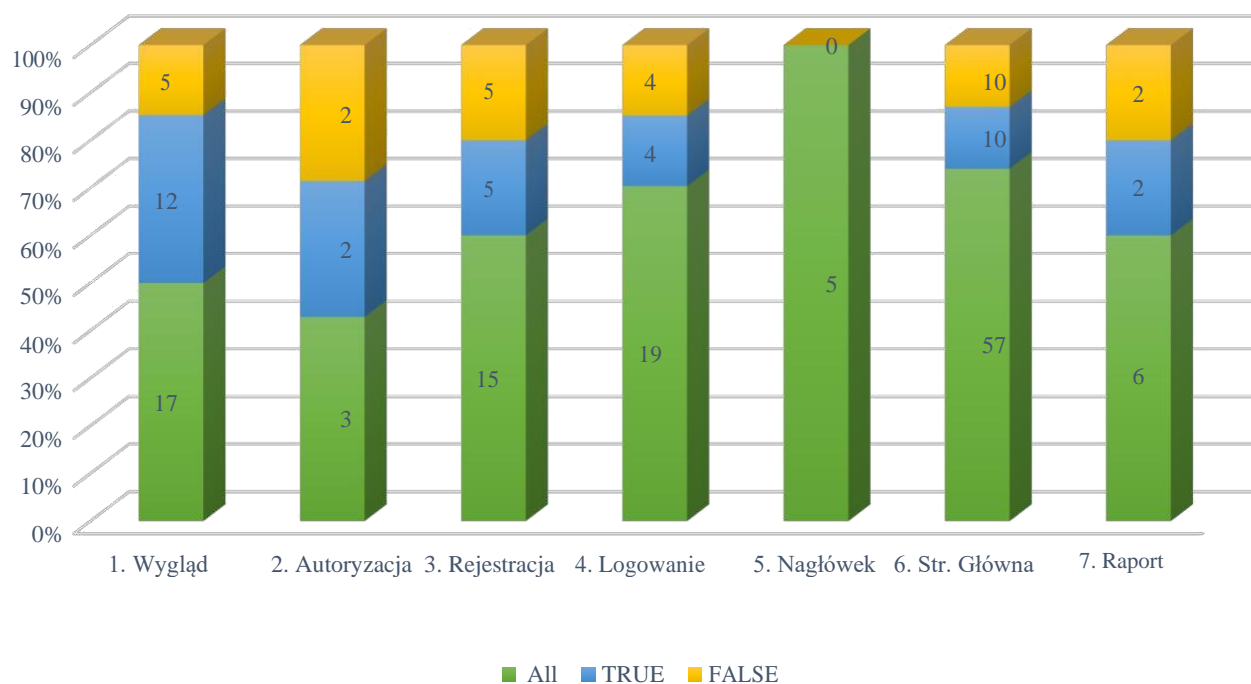
10. Podsumowanie:**Test casy.**

- 1. Wykonane.
- 2. Wynik zgodny z oczekiwaniem.
- 3. Wynik niezgodny z oczekiwaniem.

WYKRES 1. Ilość test casów.

Podział testów na kategorie.

WYKRES 2. Ilość test casów z podziałem na kategorie.

Ilość testów z podziałem na wynik.

WYKRES 3. Ilość testów wykonanych, z wynikiem oczekiwanym i wynikiem odmiennym do oczekiwanego w każdej kategorii.

W podejściu ogólnym program działa i spełnia swoją funkcję. Dla poprawy funkcjonalności i odbioru programu przez użytkownika proponuję wprowadzenie pewnych poprawek, które upłynnią działanie programu i zwiększą jego funkcjonalność.

Poniżej lista obszarów, na które należy zwrócić szczególną uwagę:

1. Akceptowanie przez system adresów mailowych i haseł niezgodnych ze specyfikacją i wymaganiami/podpowiedziami na stronie – proponuję ponowne przyjrzenie się specyfikacji i wprowadzenie w niej zmian lub zwiększenie zabezpieczeń dotyczących loginu i hasła. Wymagania prezentowane użytkownikowi oraz faktyczne dane przyjmowane przez system powinny być spójne.
2. Brak możliwości odzyskania loginu i hasła – konieczne jest wprowadzenie odpowiedniej funkcjonalności, bez tego program traci użyteczność dla użytkownika, który stracił/zapomniał hasło lub login. Użytkownik, który utracił/ zapominał login lub hasło traci całkowicie dostęp do programu.
3. Wyskakujące okno tam, gdzie nie powinno go być przypominające o konieczności zmiany sald przy pierwszym logowaniu (okno startowe pojawiające się przy powrocie do strony głównej klikając w Logo) wprowadzające użytkownika w błąd – warto zwrócić na to uwagę, jest denerwujące.
4. Suma miesięczna tylko dla bieżącego miesiąca – nie ma możliwości zmiany widoku sumy na wcześniejszy lub poprzedni miesiąc, co jest mocno ograniczające dla funkcjonalności programu i nie pozwala na pełną analizę wydatków i wpływów.
5. Raporty – raporty nie są kompletne, prezentują tylko część możliwości i część danych.
6. Aspekt wizualny – szczegóły do dopracowania.

Dokument przygotowany i opracowany na zaliczenie kursu QA Engineer przez Szczepan Szczęch.