

Dokumentacja testowa

Spis treści

I Wstęp	3
1. Tworzenie nazw plików:	3
2. Wprowadzenie	3
3. Cele	3
4. Podejście do testów	3
II Opis testów	4
1. Testowany obiekt	4
2. Narzędzia	4
3. Funkcjonalność nietestowana	4
4. Plan testów	4
5. Wstępny plan testów (zaaprobowany)	5
6. Dokumenty dodatkowe do planu wykonanych testów	7
III Zarządzanie projektem testowym i metryki	8
1. Pokrycie wymagań	8
2. Pokrycie błędów	8
3. System śledzenia defektów	8
4. Definiowanie ważności defektów	8
5. Raporty defektów	9
6. Kryterium zakończenia testów	9
7. Końcowy raport wydania	9
8. Potwierdzanie zgodności końcowej wydanego produktu	10

I Wstęp

1. Tworzenie nazw plików:

Solucja testowa składa się z 1 plikiem zawierające wszystkie przypadki testowe które zostały testowane manualnie.

2. Wprowadzenie

Ankieter jest aplikacją do tworzenia, wypełniania i analizowania ankiet. Zawiera wiele funkcjonalności, które mają na celu tworzenie łatwych, przyjaznych i szybkich ankiet dla użytkownika.

3. Cele

Chcemy osiągnąć działającą, nie zawierającą błędów uniemożliwiających pracę, aplikację, która będzie możliwa do stosowania przez każdy kto chce utworzyć lub wypełnić ankietę. Aplikacja ma ułatwiać tworzenie ankiet, wypełnianie i przesyłanie ankiet, możliwość usuwania i blokowania ankiet, jak i również wgląd do wyników ankiet.

4. Podejście do testów

Testowanie projektu ma na celu sprawdzenie:

1. czy spełnia on wszystkie postawione na początku wymagania,
2. czy wszystkie funkcjonalności działają poprawnie,
3. czy projekt jest odporny na niestandardowe (skrajne) dane lub zachowania użytkownika,
4. czy nie wkraść się żaden błąd ludzki,
5. czy nie istnieją elementy, które użytkownik mógłby uznać za niejasne lub niedopuszczalne podczas użytkowania,
6. czy w każdej sytuacji aplikacja zachowuje się, tak jak tego można byłoby się w danej chwili spodziewać.

II Opis testów

1. Testowany obiekt

Obiektem testowym jest aplikacja do zrobienia i wypełnienia ankiet Ankieter napisana w języku Java.

2. Narzędzia

Wszystkie testy zostały sporawdzone manualnie.

Wszystkie testy zostały zgłoszone do odpowiedzialnego członka zespołu poprzez raporty z testów przygotowywane w pliku pdf „RAPORTY Z TESTÓW”.

3. Funkcjonalność nietestowana

Połączenie internetowe – niemal niezawodne działanie aplikacji, jest gwarantowane jedynie przy stałym i dość szybkim połączeniu internetowym, gdyż aplikacja opiera się o ciągłą komunikację z bazą danych

Zasoby bazy danych – zakłada się, że zasoby bazy danych nie są ograniczone, jednakże aplikacja zarządza zasobami najlepiej jak to możliwe

4. Plan testów

Wszystkie testy są manualne. Przede wszystkim zostaną sprawdzone testy do wymagań funkcjonalnych. Następnie będzie wykonane testowanie statyczne w celu znalezienia innych błędów. Równocześnie będą sprawdzane wymagania нефunkcjonalne i wymagania związane z interfejsem. Przypadki użycia zostaną wykorzystane do uzyskania podstawowego pomysłu na testy.

Przypadki testowe powinny zawierać: typowy scenariusz użycia, błędny, nietypowy scenariusz użycia, wszystkie scenariusze użycia, gdzie istnieje konieczność sprawdzenia zachodzenia danych warunków, które w określonych okolicznościach mogą prowadzić do zgłoszenia przez system komunikatu błędu.

Najważniejszą częścią są testy akceptacyjne, gwarantujące zgodność produktu z wymaganiami klienta. Następnie przejście przez podstawowe i dodatkowe funkcjonalności systemu, niezbędne do prawidłowego jego funkcjonowania oraz upewnienie się, iż żadna funkcjonalność, nie jest w żadnym miejscu nadużyciem. Kolejnym krokiem jest próba niestandardowego oraz błędnego zachowania w celu upewnienia się, iż nie są możliwe żadne nieprzewidziane zachowania. Gdy testy manualne nie wykazują już więcej defektów i miejsc w aplikacji, które wymagają poprawy, w celu poprawnego jej funkcjonowania, aplikację uznaję, jako gotową do weryfikacji i testowania dla użytkowników docelowych. Za testowanie aplikacji odpowiedzialni jest tester projektu : Blagoja Mladenov, jak i również developerzy, którzy po zaimplementowaniu danej funkcjonalności powinni wstępnie zweryfikować jej poprawność.

5. Wstępny plan testów (zaaprobowany)

Na podstawie wymagań oraz znajomości głównych funkcjonalności aplikacji, zostaje sporządzona lista obszarów, które koniecznie powinny zostać pokryte testami oraz przypadki testowe, o których nie należy zapomnieć. Testowanie zostanie prowadzone podczas całego czasu trwania implementacji projektu (po zakończeniu każdej funkcjonalności) . Kolejne przypadki testowe będą dopisywane w miarę powstawania kolejnych funkcjonalności. Zostaną zakończone po zakończeniu implementacji, w celu upewnienia się, iż aplikacja spełnia wszystkie wymagania i nadaje się do użytku przez klientów końcowych. Wszelkie awarie, defekty, błędy, incydenty, niezgodności i wyjątki, będą na bieżąco zgłaszane do zespołu implementującego aplikację, a następnie weryfikowane w celu zakwalifikowania danego problemu, jako naprawiony, a aplikacji w pełni gotowej do oddania w ręce użytkowników. Po zakończeniu każdej funkcji było zrobione testowanie statyczne. Następne były przeprowadzone testy manualne do sprawdzania poprawności wszystkie wymagania funkcjonalne i niefunkcjonalne. Hierarchia kolorów była testowana statycznie już na samego początku tworzenia aplikacji.

Obszary do pokrycia:

1. Rejestrowanie/ Logowanie do systemu:
 - a. poprawne
 - b. niepoprawne
 - c. puste
 - d. brak – funkcja Gościa
 - e. logowania nieistniejącego użytkownika
 - f. rejestrowanie istniejącego użytkownika
2. Przeglądanie listy publicznych ankiet
 - a. Gość – możliwość przeglądania listę publicznych ankiet
 - b. Użytkownik – możliwość przeglądania listę publicznych ankiet
3. Wypełnianie i przesyłanie ankiet
 - a. czy gość może wypełnić i przesać ankietę publiczną
 - b. czy gość ma dostęp do ankiety prywatne bez kod dostępu
 - c. czy użytkownik może wypełnić i przesać ankietę publiczną
 - d. czy gość może wypełnić i przesać ankietę prywatną do którą ma kod
 - e. czy użytkownik może wypełnić i przesać ankietę do którą ma kod dostępu
4. Tworzenie ankiet
 - a. Ankieta bez pytaniami:
 - b. Ankieta zawierająca więcej niż 100 pytań
 - c. Czy można dodać pytanie mające 0 odpowiedzi
 - d. Czy można dodać ankietę bez opisem
 - e. Czy można dodać pytanie mające więcej niż 10 odpowiedzi
 - f. Czy można dodać ankietę bez nazwy

5. Ustawienia publiczności ankiety
 - a. Sprawdzanie czy ankieta publiczna jest widoczna dla wszystkich
 - b. Czy ankieta publiczna może być wypełniona przez wszystkich
 - c. Czy ankieta prywatna jest widoczna
 - d. Czy kod dostępu działa
 - e. Czy można dodać własny kod dostępu
 - f. Czy można wypełnić i przesłać ankietę prywatną
 - g. Czy ankieta prywatna jest widoczna dla właściciela
6. Usuwanie ankiet
 - a. Czy użytkownik może usunąć ankietę stworzoną przez jego
7. Ukrywanie wyników ankiety
 - b. Czy wyniki takiej ankiety są zablokowane
 - c. Czy można odblokować
 - d. Czy właściciel ma dostęp do tych wyników
8. Zablokowanie możliwości wypełniania ankiety
 - a. Czy można zablokować ankiety
 - b. Czy inni użytkownicy mogą wypełnić i przesłać ankietę
 - c. Czy można odblokować ankietę
 - d. Czy zablokowana ankieta jest i dalej widoczna dla wszystkich o ile jest ona publiczna
9. Wgląd do wyników ankiety
 - a. Czy gość może zobaczyć wyniki ankiety
 - b. Czy gość może zobaczyć ankietę o ile jej wyniki zostali ukryte
 - c. Czy autor ankiety ma dostęp do wyników o ile zostały one ukryte
 - d. Czy autor ankiety ma dostęp do wyników ankiety publicznie
 - e. Czy autor ankiety ma dostęp do wyników ankiety prywatnej
 - f. Czy użytkownik może zobaczyć wyniki ankiety
10. Generowanie linku/kodu dostępu do ankiety
 - a. Czy można wygenerować kod dostępu do ankiety
 - b. Czy kod działa
11. Eksportowanie wyników ankiety
 - a. Czy można eksportować poprawnie do txt
 - b. Czy można eksportować poprawnie do csv
12. Pytanie wielokrotnego wyboru
 - a. Czy są dostępne pytanie pojedynczego wyboru
 - b. Czy są dostępne pytania wielokrotnego wyboru
 - c. Czy pytanie wielokrotnego wyboru działają i odpowiedzi są poprawnie zapisane
13. Hierarchia kolorów (była sprawdzana statycznie na początku samego projektu)
14. Pola obowiązkowe
 - a. Czy jest możliwość pola obowiązkowego dla pytanie pojedynczego wyboru
 - b. Czy jest możliwość pola obowiązkowego dla pytanie wielokrotnego wyboru
 - c. Czy można odznaczyć pole obowiązkowe
 - d. Czy wyświetla się komunikat o błędzie jeśli takie pytanie nie zostało odpowiedziane
 - e. Co się dzieje jeśli zaznaczymy i odznaczmy takie pytanie

6. Dokumenty dodatkowe do planu wykonanych testów

Każdy przypadek testowy składać się ma z części przygotowującej dane, części wykonującej odpowiednie metody oraz części porównującej otrzymane wyniki i wynikami oczekiwanymi. Przypadki testowe powinny sprawdzać możliwie jak najwięcej przykładowych danych wejściowych, prowadzących do uzyskania oczekiwanych rezultatów oraz niepoprawnych i oczekiwanych rezultatów błędnych oraz wyjątków. Przygotowywane dane wejściowe do każdej testowanej metody, powinny być danymi możliwymi do wprowadzenia za pośrednictwem interfejsu użytkownika oraz przetestowane w taki sposób, jak będą używane w aplikacji. Wszelkie dane wejściowe muszą być przygotowane przed wykonaniem każdego testu oraz usunięte po wykonaniu wszystkich testów. Każdy przypadek testowy powinien testować jedną funkcjonalność danej metody oraz w razie niepowodzenia informować jasnym i zrozumiałym raportem, zawierającym dane prowadzące do zakończenia testu niepowodzeniem. Test powinien zakończyć się powodzeniem w sytuacji otrzymania danych identycznych z oczekiwanymi.

7. Dokumenty dodatkowe do planu finalnych testów

Testowanie rozpoczęło się i zakończyło zgodnie z planowanym czasem. Efekty napisanych testów są zadowalające, gdyż umożliwiły wykrycie nawet niewielkich błędów developerów, już na najbardziej wczesnym poziomie. Wszelkie błędy zostały poprawione na bieżąco, co pozwalało na dalsze testowanie danej funkcjonalności również pod przy innymi danych wejściowych, co pozwalało na określenie iż dany błąd został naprawiony i nie powoduje zagrożenia dla poprawnego działania systemu.

III Zarządzanie projektem testowym i metryki

1. Pokrycie wymagań

$$RC = \frac{\text{liczba przetestowanych wymagań}}{\text{liczba wymagań zapisanych w specyfikacji}} = \frac{21}{21}$$

2. Pokrycie błędów

$$EC = \frac{\text{liczba błędów wykrytych w fazie testów akceptacyjnych}}{\text{liczba wszystkich błędów wykrytych zgodnie ze specyfikacją}} = \frac{1}{3}$$

3. System śledzenia defektów

Defekty znalezione zostały podczas prowadzonej manualnej weryfikacji. Testy już napisane również dobrze weryfikowały, czy żadna z dotychczas napisanych części systemu nie została popsuta, nowo wprowadzoną zmianą.

4. Definiowanie ważności defektów

Ocena ważności defektu jest subiektywną metodą używaną do zaraportowania stopnia ważności każdego zaraportowanego defektu. Defekty oceniane będą na podstawie następujących wytycznych:

1. Defekt **WYSOKIEGO PRIORYTETU** (ważność 1)
 - a. Występuje, gdy brakuje podstawowej funkcjonalności, użyteczności lub wydajności produktu podczas normalnych operacji i nie ma możliwości stworzenia „obejścia”
 - b. Defekty niezwiązane z funkcjonalnością, ale będące jasnymi pomyłkami
2. Defekt **ŚREDNIEGO PRIORYTETU** (ważność 2)
 - a. Dla najważniejszych funkcjonalności, które nie działają prawidłowo w określonych warunkach lub funkcjonalności drugiego rzędu, które w ogóle nie działają lub słaba jakość funkcjonalności lub wydajności podczas normalnych operacji, trudności z użyciem podstawowych funkcjonalności etc.
 - b. Defekty powinny zostać naprawione w cyklu tworzenia oprogramowania. Jednak w ostatecznej fazie testowej mogą zostać zaakceptowane (znaczy nienaprawione).
3. Defekt **NISKIEGO PRIORYTETU** (ważność 3) – (nie ma w RAPORTY Z TESTÓW)
 - a. Defekty, które wprowadzają pewne problemy w obsłudze, nie zdarzające się zbyt często, m. in. pomniejsze problemy w wyświetlaniu, błędy językowe, pomniejsze problemy z działaniem oraz podobne.
 - b. Defekty powinny być naprawione w międzyczasie implementacji innych funkcjonalności przez programistów
 - c. Należy pamiętać, iż wiele małych defektów może się przełożyć na ogólną niską ocenę jakości produktu, stąd konieczna jest ich poprawa na bieżąco

5. Raporty defektów

Lista podsumowująca znalezione defekty:

1. błędy w procedurach ilości charakterów podczas logowania/rejestrowania
2. brak wymagane pytania wielokrotnego wyboru
3. niepoprawne dane dla blokowanej ankiety
4. niepoprawne dane dla blokowane wyniki ankiety
5. brak komunikati o błędu
6. mało ograniczenie dla pytań
7. brak możliwości wypełnienia ankiety prywatnie
8. brak możliwości odznaczanie wymaganego pytania

Znane ograniczenia:

1. nie można odblokować ankiety
2. nie można się wylogować
3. nie można odblokować wyniki ankiety
4. nie można dodać własne hasło do ankiety
5. nie można się reregistrować/wylogować z innym nickiem lub hasło, konieczne jest otwieranie nowej

kartce

6. Kryterium zakończenia testów

Podstawowym kryterium zakończenia testów jest:

1. zweryfikowanie poprawności i zgodności z opisem wymagań aplikacji, które zostały postawione przez klienta aplikacji,
2. poprawne manualne przejście przez wszystkie możliwe funkcje systemu zakończone powodzeniem, a co za tym idzie nie znalezieniem żadnych rażących błędów oraz defektów systemu, które mogłyby spowodować nieprawidłowe jego funkcjonowanie – klasyfikacja błędów odbywa się przez co najmniej jednego developera, dwóch testerów i Team Leadera
3. uznanie iż nie istnieją elementy projektu, których poprawa mogłaby wnieść jakąkolwiek wartość do projektu

7. Końcowy raport wydania

Testy manualne również pozwoliły na upewnienie się, iż użytkownik końcowy otrzymał w pełni działający system, który ułatwi mu codzienną pracę. Zrobione testy systemowe znacząco skróciły czas lokalizacji błędów, a co za tym idzie, tworzenia całej aplikacji. Wcześniej przygotowane plany testów oraz przygotowana architektura i schematy, znacząco przyspieszyły proces robienia testów i zapewniły, że wszystkie obszary aplikacji zostały przetestowane. Można stwierdzić, iż przeprowadzenie procesu testowania w taki sposób, byłoby bardzo dobrą decyzją.

8. Potwierdzanie zgodności końcowej wydanego produktu

Produkt po zakończeniu procesu implementacji oraz testowania, spełnia wszystkie określone przez klienta wymagania. Aplikacja została przetestowana w 85%, zatem istnieje niezerowa szansa na wystąpienie drobnych defektów ważności 2 i 3, jednak doświadczony sztab członków zespołu przez długi czas pracował nad prawidłowym funkcjonowaniem aplikacji, jego optymalnym i intuicyjnym działaniem, interfacem przyjaznym dla użytkownika, zrozumiałą, szczegółową dokumentacją oraz nad minimalizacją wszelkich niedociągnięć, co czyni aplikację w pełni używalną, odporną na krytyczne defekty oraz skazaną na satysfakcję przyszłych użytkowników.