PLAN TESTU DO APLIKACJI „PROJECT MASTER”

1. Wstęp

1.1 Cel i zakres
Celem tego dokumentu jest przedstawienie szczegółowego planu testów aplikacji webowej do zarządzania projektami „Project Master”, która umożliwia użytkownikom tworzenie, śledzenie i zarządzanie projektami oraz zadaniami w ramach tych projektów. Plan obejmuje testowanie funkcjonalności aplikacji, integracji między modułami, wydajności systemu oraz bezpieczeństwa.

1.2 Terminologia
UI (User Interface): Interfejs użytkownika.
API (Application Programming Interface): Zestaw protokołów do komunikacji między różnymi częściami aplikacji.
Regression Testing: Testy regresyjne, czyli testy powtarzalne po wprowadzeniu zmian w aplikacji.
Load Testing: Testy obciążeniowe mające na celu sprawdzenie, jak aplikacja działa przy dużej liczbie użytkowników.

1.3 Odwołania
Ten dokument odnosi się do „Test Policy” firmy, która zawiera wytyczne dotyczące standardów testowania oraz zasad zarządzania procesem testowym.

2. Opis projektu

2.1 Opis systemu/testowanego produktu
Aplikacja „Project Master” to aplikacja internetowa , która umożliwia użytkownikom zarządzanie projektami skierowana do małych i średnich firm. System zawiera moduły takie jak zarządzanie projektami, zadaniami oraz profilami użytkowników. System komunikacji między użytkownikami, śledzenie postępów i generowanie raportów, oraz centrum powiadomień. Testy będą obejmować wersję przeglądarkową.

2.2 Cele testowania
Celem testowania jest zapewnienie, że wszystkie funkcje platformy działają poprawnie i zgodnie z oczekiwaniami. Dodatkowo, testowanie ma na celu zidentyfikowanie i usunięcie wszelkich błędów i zagrożeń związanych z bezpieczeństwem.

2.3 Zakres testów
Zakres testów obejmuje:
Testy funkcjonalne: sprawdzenie głównych funkcji aplikacji (zarządzanie projektami i zadaniami, profilami użytkowników, śledzenie postępów , komunikacja między użytkownikami , generowanie raportów, powiadomienia, oraz rejestracja i logowanie użytkowników)
Testy niefunkcjonalne: ocena wydajności, użyteczności i skalowalności, dostępności i zgodności.
Testy bezpieczeństwa: identyfikacja potencjalnych zagrożeń dla danych użytkowników.
Testy regresyjne: powtarzalne testowanie po każdej większej aktualizacji systemu.

2.4 Założenia i ograniczenia
Testowanie będzie odbywać się na symulowanych środowiskach testowych, co może wpływać na dokładność wyników w porównaniu z rzeczywistym środowiskiem

produkcyjnym. Ograniczenia obejmują czas oraz zasoby dostępne do testów.

3. Strategia testowania

3.1 Strategia testów funkcjonalnych
Testy funkcjonalne będą obejmować wszystkie podstawowe funkcje aplikacji „Project Master”, w tym:
Proces rejestracji i logowania użytkownika.
Zarządzanie projektami i zadaniami oraz profilami użytkowników.
Śledzenie postępów.
Komunikacja między użytkownikami.
Generowanie raportów.
Powiadomienia.

3.2 Strategia testów niefunkcjonalnych
W ramach testów niefunkcjonalnych, skupimy się na:
Wydajności: sprawdzenie czasu ładowania stron oraz szybkości działania aplikacji przy wzrastającej liczbie użytkowników.
Bezpieczeństwa: sprawdzenie czy dane użytkowników w tym hasła są przechowywane w sposób bezpieczny.
Skalowalności: ocena aplikacji przy zwiększonym obciążeniu.
Użyteczności: badania użyteczności i intuicyjności interfejsu użytkownika.

3.3 Strategia testów regresyjnych
Automatyzacja testów regresyjnych dla kluczowych funkcji, takich jak logowanie i rejestracja, pozwoli na szybkie wykrywanie błędów w wersjach rozwojowych. Testy regresyjne będą uruchamiane po każdej aktualizacji systemu.

3.4 Strategia testów integracyjnych
Testy integracyjne obejmują współdziałanie modułów takich jak generowanie raportów, komunikacja między użytkownikami, śledzenie postępów oraz centrum powiadomień.

3.5 Strategia testów wydajnościowych
Testy wydajnościowe będą obejmować:
Obciążeniowe: ocena, jak system radzi sobie z przewidywaną liczbą równoczesnych użytkowników.
Stres testy: sprawdzenie działania systemu przy ekstremalnym obciążeniu.
Testy czasu reakcji: analiza szybkości ładowania strony.

3.6 Strategia testów bezpieczeństwa
Główne aspekty testów bezpieczeństwa to:
Testy podatności na ataki typu SQL Injection, XSS.
Testy autoryzacji, aby zapewnić, że dane użytkowników są chronione.
Testy zgodności z ogólnymi wymogami bezpieczeństwa danych osobowych.

3.7 Planowanie i harmonogram testów
Testowanie zostało zaplanowane na dwa miesiące, szczegóły przedstawione są w punkcie 4.1.

3.8 Środowiska testowe i narzędzia
System będzie testowany w środowisku testowym odzwierciedlającym produkcyjne, z wykorzystaniem takich narzędzi jak:

Selenium do automatyzacji testów funkcjonalnych,
K6 do testów wydajnościowych,
Burp Suite do testów bezpieczeństwa.
Jmeter do testów wydajnościowych.
Jira do zarządzanie testami.

4. Planowanie testów

4.1 Harmonogram testów
Poniżej przedstawiono harmonogram dla testów aplikacji „Project Master”:
Etap 1: Testy jednostkowe -1 tydzień
Etap 2: Testy funkcjonalne – 3 tygodnie
Etap 3: Testy wydajnościowe – 1 tydzień
Etap 4: Testy bezpieczeństwa – 1 tydzień
Etap 5: Testy akceptacyjne – 2 tygodnie

5. Specyfikacja przypadków testowych

5.1 Identyfikacja przypadków testowych
Każdy przypadek testowy będzie opisany unikalnym identyfikatorem. Przypadki obejmują główne funkcje, takie jak zarządzanie projektami i zadaniami,profilami użytkowników, śledzenie postępów , komunikacja między użytkownikami , generowanie raportów, powiadomienia, oraz rejestracja i logowanie użytkowników.

5.2 Opis przypadków testowych
Dla każdego przypadku opisane będą: Czynności wykonywane przez użytkownika i oczekiwane rezultaty działania.

6. Wykonanie testów

6.1 Przygotowanie środowiska testowego
Środowisko testowe odwzorowujące produkcyjne zostanie przygotowane przed rozpoczęciem testów. Przygotuje je DevOps developer.
6.2 Wykonanie testów funkcjonalnych
Realizacja testów funkcjonalnych zgodnie z planem opisanym w harmonogramie. Odpowiedzialnym za te testy będzie firma XYZ, która przejmie odpowiedzialność za tą część testów.

7. Zarządzanie incydentami

7.1 Raportowanie incydentów
Każdy wykryty błąd zostanie udokumentowany w systemie raportowania incydentów w Jira.

8. Zakończenie

8.1 Podsumowanie wyników testów i raporty z testów
Po zakończeniu testów powstanie raport z wyników testów oraz rekomendacje dotyczące dalszych działań.