

PLAN TESTÓW
Aplikacja MrBuggy 6 Projekt

Wersja 1.0

SPIS TREŚCI

1. Metryka dokumentu	3
2. Historia dokumentu	3
3. Lista załączników	3
4. Cel i zakres dokumentu	3
5. Słownik pojęć i skrótów	4
6. Plan testów	5
6.1. Sposoby organizacji testów	5
6.2. Wymagane wsparcie	5
6.3. Zakres testów	5
6.4. Harmonogram przeprowadzenia testów	6
6.5. Narzędzia wspierające przeprowadzenie testów	6
6.6. Sposoby komunikacji podczas prowadzenia testów	7
6.7. Środowisko testowe i jego ograniczenia	7
6.8. Kryteria <i>błędów</i>	7
6.9. Kryteria <i>wejścia</i>	8
6.10. Kryteria <i>zamknięcia</i>	8
6.11. Kryteria <i>wstrzymania</i>	8
7. Ryzyka	9

1. METRYKA DOKUMENTU

Nazwa produktu	MrBuggy 6 Project
Etap	1

2. HISTORIA DOKUMENTU

Wersja	Utworzyła	Data	Opis zmian
1.0	Anna Stefańska	15.07.2022	Inicjalna wersja dokumentacji

3. LISTA ZAŁĄCZNIKÓW

Nr	Nazwa dokumentu	Wersja dokumentu	Miejsce przechowywania
1	aplikacja_MrBuggy6projekt_wymagania.pdf	1.0	Dysk sieciowy WD.5B29F

4. CEL I ZAKRES DOKUMENTU

Celem głównym dokumentu jest zebranie informacji ogólnych i szczegółowych o sposobach organizacji testów i ich zakresie, potrzebnych do sprawdzenia poprawności działania aplikacji

Mr Buggy 6.

Zakres testów obejmuje weryfikację poprawności działania poszczególnych funkcjonalności aplikacji MrBuggy 6.

Wyłączone z zakresu testowania: wymagania нефункционалне

5. SŁOWNIK POJĘĆ I SKRÓTÓW

Skrót/pojęcie	Opis
Testy funkcjonalne	Sprawdzają czym zajmuje się dany system (program, aplikacja) według specyfikacji wymagań funkcjonalnych, czyli czy spełnia podstawowe kryteria postawione przez dostawcę oprogramowania przy tworzeniu danego systemu.
Testy нефункционалне	Sprawdzają w jaki sposób system działa, czyli sprawdza się czy system jest bezpieczny, użyteczny, ergonomiczny, wydajny itd.
Re-testy	Są to ponowne testy danego systemu. Wykonywane w sytuacji, gdy np. błąd został naprawiony i naszym zadaniem jest zweryfikowanie czy usterka faktycznie została naprawiona.
Regresja	To sprawdzenie systemu w przypadku gdy zostały wprowadzone w nim zmiany pod kątem możliwości pojawienia się nowych błędów w elementach systemu, które wcześniej działały poprawnie.
Testy modułowe	Nazywane też testami jednostkowymi, sprawdzają poszczególne komponenty systemu indywidualnie i niezależnie od siebie.
Testy integracyjne	Sprawdzają jak poszczególne moduły (jednostki) systemu współpracują ze sobą razem (w grupach).
Testy systemowe	Sprawdzają czy utworzony system działa poprawnie jako całość.
Testy pozytywne	Są to testy, których rezultat powinien dawać oczekiwany efekt – pozytywny bądź negatywny. Wiemy czego można się po systemie spodziewać.
Testy negatywne	Są to testy, gdzie poniekąd próbujemy „przechytrzyć” system i próbujemy na różne sposoby wywołać błędy w jego działaniu, których tworzący go programista mógł nie zauważyć.
Testy akceptacyjne	Gdy system albo przynajmniej jego duża funkcjonalność powinny już działać, wg zamysłów programistów, możemy sprawdzać czy spełnia on oczekiwania odbiorców – klienta i użytkowników. Celem tych testów nie jest wykrywanie błędów.
Defekt	Awaria bądź błąd systemu, która powoduje że funkcja bądź moduł funkcji systemu nie działa bądź nie działa poprawnie.
Zestaw testowy	Układ przypadków testowych testów pozytywnych i negatywnych mających za zadanie sprawdzenie modułu (jednostki) systemu.
Scenariusz testowy	Spis elementów do sprawdzenia które są konieczne do wykonania.
Przypadek testowy	Zbiór instrukcji do wykonania i oczekiwanych reakcji systemu przy wykonywaniu testów pozytywnych i negatywnych.
Dysk sieciowy	Dysk do którego jest dostęp poprzez sieć Internet znajdujący się w innej lokalizacji.

6. PLAN TESTÓW

6.1. SPOSOBY ORGANIZACJI TESTÓW

W trakcie wykonywania testów przeprowadzona zostanie ścieżka postępowania zgodna z przygotowanymi przypadkami testowymi znajdującymi się w narzędziu TestRail pod adresem:

https://annamariastefanska92.testrail.io/index.php?/suites/view/5&group_by=cases:section_id&group_order=asc&display_deleted_cases=0

Błędy wykryte podczas testów zostaną zarejestrowane w narzędziu TestRail oraz w JIRA (<https://anqla13.atlassian.net/jira/software/c/projects/MPK/boards/3>).

Wykonany zostanie szereg testów funkcjonalnych, modułowych, a także testy systemowe.

6.2. WYMAGANE WSPARCIE

Kierownik: Anna Stefańska

Testerzy: Zespół XYZ

6.3. ZAKRES TESTÓW

Testy aplikacji MrBuggy6 Project obejmować będą następujące funkcjonalności:

- Ustawianie hasła użytkownika
- Logowanie i wylogowanie użytkownika
- Edycja profilu użytkownika
- Dodawanie nowych pracowników oraz edytowanie ich danych personalnych
- Dodawanie klientów oraz edytowanie ich danych personalnych
- Przypisywanie pracowników do usługi oraz klientów
- Zarządzanie „Opportunities” (CRUD)
- Opcje filtrowania

6.4. HARMONOGRAM PRZEPROWADZENIA TESTÓW

Zadanie	Data
1. Przygotowanie przypadków testowych	06-16.07.2022
2. Przygotowanie danych testowych	
3. Testy dymne	
4. Wykonanie testów systemowych	
5. Raportowanie błędów	
6. Sporządzenie dokumentacji zbiorczej defektów	
7. Sporządzenie raportu z przeprowadzonych testów	16.07.2022

6.5. NARZĘDZIA WSPIERAJĄCE PRZEPROWADZENIE TESTÓW

Narzędzie	Opis
TestRail	Repozytorium przypadków testowych
Generator haseł	Aplikacja dostępna pod adresem URL: (https://generator.blulink.pl/) generująca hasła zgodne z wybranymi wymaganiami
Generator imion i nazwisk	Aplikacja dostępna pod adresem URL: (https://random-data-generator.com/), generująca przypadkowe imiona i nazwiska. Oparta o bazę najpopularniejszych imion i nazwisk w Polsce.
Generator adresu email	Aplikacja dostępna pod adresem URL: (https://yopmail.com/pl/email-generator) generująca losowe adresy e-mail.

6.6. SPOSOBY KOMUNIKACJI PODCZAS PROWADZENIA TESTÓW

Komunikacja:

Zdalna z użyciem przeglądarki internetowej umożliwiającej korzystanie z platformy komunikacyjnej Slack oraz pocztą elektroniczną.

6.7. ŚRODOWISKO TESTOWE I JEGO OGRANICZENIA

Środowisko testowe:

- Komputer stacjonarny klasy PC z systemem operacyjnym Windows 10 Home 64-bit (v. 21H2)
- Pakiet biurowy: Microsoft Word dla Microsoft 360 MSO (wersja 2206 kompilacji 16.0.15330.20216) 64-bitowa
- Przeglądarka Chrome w najnowszej wersji 103.0.5060.114

6.8. KRYTERIA BŁĘDÓW

Poziom błędu	Opis
Krytyczne	Program zachowuje się niepoprawnie, brak możliwości wykonania krytycznych operacji (jeśli istnieje możliwość ich wykonania jest to znacznie utrudnione). Dochodzi do uszkodzenia lub celowego usunięcia danych. Program zwraca wyjątek. Program zostaje zatrzymany i wyłącza się.
Bardzo Poważne	Błędy związane z funkcjonalnościami, możliwe do obejścia. W ich wyniku wykonywane są operacje, które nie powinny być możliwe.
Poważne	Błędy wpływające na bezpieczeństwo danych, występujące w niekrytycznych obszarach funkcjonalnych.
Drobne	Błędny lub wprowadzający w błąd komunikat. Błędy związane z nieprawidłowym wyświetlaniem znaków lub elementów poszczególnych modułów, brak zachowania standardów UX.

6.9. KRYTERIA WEJŚCIA

Kryterium	Spełnienie
Przygotowanie opisu testów	90% zestawów testowych 90% scenariuszy testowych 70% przypadków testowych
Gotowość środowiska testowego*	50%
Dostępność i gotowość narzędzi testowych	100%
Dostępność danych testowych	20%

6.10. KRYTERIA ZAMKNIĘCIA

Kryterium	Spełnienie
Przeprowadzenie i zakończenie z wynikiem pozytywnym testów funkcjonalnych	100% wykonanych przypadków testowych
Przeprowadzenie i zakończenie z wynikiem pozytywnym testów systemowych	100% wykonanych przypadków testowych 50% pozytywnie zakończonych przypadków testowych
Przeprowadzenie i zakończenie z wynikiem pozytywnym testów akceptacyjnych	100% wykonanych przypadków testowych

6.11. KRYTERIA WSTRZYMANIA

Kryterium	Spełnienie
Wystąpienie błędów krytycznych	Aktualizacja systemu operacyjnego
Zawieszenie się aplikacji MrBuggy 6-brak odpowiedzi	Reset aplikacji. Wyłączenie i ponowne włączenie aplikacji.

Ryzyko produktowe	Mitygacja
<p>Wyciek danych osobowych pracowników oraz klientów firmy</p>	<ul style="list-style-type: none"> - rekomendacja wdrożenia 2 Factor Authentication, - testy związane z logowaniem i zmianą hasła mają najwyższy priorytet, - zlecenie wykonania testów bezpieczeństwa podmiotowi zewnętrznemu (obecny zespół nie posiada wystarczającej wiedzy merytorycznej niezbędnej do przeprowadzenia testów bezpieczeństwa)
<p>Aplikacja MrBuggy 6 nie jest wystarczająco intuicyjna i nowi klienci nie posiadają wiedzy jak jej używać</p>	<ul style="list-style-type: none"> - rekomendacja przygotowania instrukcji obsługi aplikacji z dołączonymi screenshot'ami, która będzie dostępna na stronie startowej, - po zakończeniu testów funkcjonalnych rekomendowane jest przeprowadzenie dodatkowej sesji testów użyteczności

Ryzyko projektowe	Mitygacja
<p>niewystarczające kwalifikacje lub przeszkolenie pracowników bądź braki kadrowe</p>	<ul style="list-style-type: none"> -stworzenie jak najbardziej zrównoważonego zespołu pod kątem doświadczenia (w zespole powinny być zarówno osoby początkujące jak i doświadczone) -zapewnienie możliwości dodatkowego przeszkolenia pracowników (np. produktowego lub z dostępnych narzędzi na których pracuje zespół) - w przypadku wystąpienia braków kadrowych jak najszybsze znalezienie nowych członków projektu o odpowiednich kwalifikacjach
<p>wprowadzenie w ostatniej chwili zmian wymagających dokonania licznych przeróbek</p>	<ul style="list-style-type: none"> -należy rozpocząć testowanie od samego początku tworzenia produktu (już na etapie analizy specyfikacji), znacznie zmniejszy to ryzyko późniejszych zmian wynikających z niezgodności z wymaganiami -włączenie do zespołu dodatkowych pracowników, posiadających doświadczenie w tworzeniu zbliżonych aplikacji-mogą oni zwrócić uwagę na elementy wymagające zmian -należy ustalić z klientem wszystkie jego oczekiwania wobec produktu oraz omówić jak rozumie on poszczególne wymagania, takie konsultacje powinny być prowadzone jak najczęściej żeby ewentualne zmiany mogły zostać wprowadzone w trakcie