**Plan testów dla projektu PDFChecker**

**Wersja 1.1**

Spis treści

[PLAN TESTÓW **Error! Bookmark not defined.**](#_Toc511052859)

[1. Metryka dokumentu 3](#_Toc511052860)

[2. Historia dokumentu 3](#_Toc511052861)

[3. Lista załączników](#_Toc511052862) 2

[4. Cel i zakres dokumentu 3](#_Toc511052863)

[5. Słownik pojęć i skrótów 3](#_Toc511052864)

[6. Plan testów 4](#_Toc511052865)

[6.1. Sposoby organizacji testów 5](#_Toc511052866)

[6.2. Wymagane wsparcie 5](#_Toc511052867)

[6.3. Zakres testów 5](#_Toc511052868)

[6.4. Harmonogram przeprowadzenia testów 5](#_Toc511052869)

[6.5. Narzędzia wspierające przeprowadzenie testów 6](#_Toc511052870)

[6.6. Sposoby komunikacji podczas prowadzenia testów 6](#_Toc511052871)

[6.7. Środowisko testowe i jego ograniczenia 7](#_Toc511052872)

[6.8. Kryteria błędów 7](#_Toc511052873)

[6.9. Kryteria *wejścia* 7](#_Toc511052874)

[6.10. Kryteria *zamknięcia* 7](#_Toc511052875)

[6.11. Kryteria wstrzymania 7](#_Toc511052876)

**1. Metryka dokumentu**

|  |  |
| --- | --- |
| **Nazwa produktu** | PDFChecker v1.1 |
| **Etap** | 1 |

**2. Historia dokumentu**

|  |  |  |  |
| --- | --- | --- | --- |
| **Wersja** | **Utworzył** | **Data** | **Opis zmian** |
| 1.0 | Karolina Podbilska | 01.04.2020 | Inicjalna wersja dokumentacji |
| 1.1 | Karolina Podbilska | 02.04.2020 | Uzupełniona wersja dokumentacji |
| 2.0 | Karolina Podbilska | 03.04.2020 | Finalna wersja dokumentacji |

**3. Lista załączników**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nr** | **Nazwa dokumentu** | **Wersja dokumentu** | **Miejsce przechowywania** |
| 1 | Plan testów\_PDFChecker\_v2.0.docx | 2.0 | drive.google.com |
| 2 | Dokumentacja\_SystemIT\_KosinscyIT.docx | 1.0 | drive.google.com |
| 3 | Pytania\_do\_dokumentacji\_PDFChecker\_v1.1.docx | 1.1 | drive.google.com |
| 4 | Dane\_testowe\_PDFChecker\_v1.2.zip | 1.2 | drive.google.com |
| 5 | Techniki\_projektowania\_testów\_PDFChecker\_v1.0.docx | 1.0 | drive.google.com |
| 6 | Scenariusze\_Testowe\_PDFChecker\_v1.1.docx | 1.1 | drive.google.com |
| 7 | Przypadki\_Testowe\_PDFChecker\_v1.1.xlsx | 1.1 | drive.google.com |
| 8 | Zgłaszanie\_defektu\_PDFChecker\_v1.0.xlsx | 1.0 | drive.google.com |
| 9 | Defekty\_zrzuty\_ekranu\_PDFChecker\_v1.0.zip | 1.0 | drive.google.com |
| 10 | Raport\_Testy\_PDFChecker\_v1.11.doc | 1.11 | drive.google.com |

**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 programu PDFChecker v1.1. |
| **Zakres testów** obejmuje weryfikację poprawności działania poszczególnych funkcjonalności programu PDFChecker v1.1. |

**Ograniczenia**Programu PDFChecker w wersji 1.1 nie należy testować na innych środowiskach testowych. Zaleca się ograniczenie do systemu Windows 10, na komputerze o identycznych bądź podobnych parametrach technicznych.

**Wyłączone z zakresu testowania**: wymagania niefunkcjonalne

**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 niefunkcjonalne | 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 przedstawionymi zestawami, scenariuszami i przypadkami testowymi. Jeżeli pojawią się błędy, to zostaną odnotowane i zarejestrowane.

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

**6.2. Wymagane wsparcie**

**Kierownik:** Justyna Kałek

**Testerzy:** Karolina Podbilska

**6.3. Zakres testów**

Testy programu PDFChecker v1.1 obejmować będą:

* sprawdzanie czy program posiada możliwość wczytania plików w dowolnym formacie,
* sprawdzanie jak program się zachowuje przy wczytywaniu plików w formacie PDF utworzonych z innych plików poprzez zmianę rozszerzenia pliku,
* sprawdzanie funkcjonalności programu poprzez pliki PDF, tworzone pod kątem sprawdzenia poprawności działania programu – testy pozytywne,
* sprawdzanie poprawności wyświetlania informacji w generowanych przez program raportach.

**6.4. Harmonogram przeprowadzenia testów**

|  |  |
| --- | --- |
| **Zadanie** | **Data** |
| 1. Analiza dokumentacji technicznej Systemu Kosińscy IT | 01.04.2020 |
| 1. Stworzenie przypadków testowych |
| 1. Utworzenie serii plików tekstowych w programie Word | 03.04.2020 |
| 1. Zapisanie utworzonych plików do formatu PDF |
| 1. Uruchomienie programu PDFChecker |
| 1. Sprawdzenie stabilności programu podczas uruchamiania |
| 1. Włączanie i wyłączanie poszczególnych pól systemu: słowa kluczowe, NIP, REGON, kapitał zakładowy |
| 1. Sprawdzenie funkcjonalności dodawania wartości w poszczególnych polach |
| 1. Szczegółowe sprawdzenie walidacji wpisywanych wartości w polach NIP, REGON, kapitał zakładowy |
| 1. Dodawanie plików do programu – pliki inne niż w formacie PDF |
| 1. Dodawanie plików w formacie PDF – pliki ze zmienionym rozszerzeniem |
| 1. Dodawanie dowolnych plików PDF |
| 1. Sprawdzenie funkcjonalności poszczególnych pól programu na wygenerowanych plikach PDF – testy modułowe |
| 1. Sprawdzenie funkcjonalności programu na wygenerowanych plikach PDF – testy integracyjne |
| 1. Sprawdzanie wyników raportów na modułowych testach pozytywnych |
| 1. Sprawdzanie wyników raportów na modułowych testach negatywnych |
| 1. Sprawdzanie wyników raportów na integracyjnych testach negatywnych i pozytywnych |
| 1. Sprawdzanie poprawności generowania raportów |
| 1. Wykonanie testów systemowych programu |
| 1. Wykrywanie błędów w działaniu programu |
| 1. Sporządzenie dokumentacji zbiorczej defektów |
| 1. Sporządzenie raportu z przeprowadzonych testów | 06.04.2020 |

**6.5. Narzędzia wspierające przeprowadzenie testów**

|  |  |
| --- | --- |
| **Narzędzie** | **Opis** |
| Windows 10 Pro wersja 1903 | Windows’owy system operacyjny |
| Narzędzia **Java SE Development Kit 8.0.1310.11** dla systemu Wondows | Pakiet narzędzi Java dla systemu Windows dla możliwości włączenia programu PDFChecker v1.1 |
| Microsoft Edge 44.18362.449.0 | Przeglądarka internetowa |
| Microsoft Office 365 ProPlus 15.0.5215.1000: Word, Excel | Edytory tekstu |

**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 Pro 64-bit (v. 1909)
* Pakiet biurowy: Microsoft Word Professional Plus 2016
* Przeglądarka plików PDF: PDF-XChange Viewer 2.5.322.10
* Zestaw narzędzi programistycznych: Java SE Development Kit 11.0.5
* PDF Checker v.1.1

**Ograniczenia:**

Testy nie powinny być wykonywane na innych wersjach systemu operacyjnego Microsoft niż Windows 10, a także na innych systemach operacyjnych, np. dowolna dystrybucja linux’owa czy system Mac OS od Apple.

**6.8. Kryteria *błędów***

|  |  |
| --- | --- |
| **Poziom błędu** | **Opis** |
| Krytyczne | Program zostaje zatrzymany i wyłącza się |
| Bardzo poważne | Brak wyświetlanych wyników w raporcie, raport nie generuje się, aplikacja przyjmuje uszkodzone pliki |
| Poważne | Nie można zmienić rozmiaru okna programu – może mieć znaczenie przy większej serii danych, niezachowana planowana struktura raportu, aplikacja przyjmuje błędne warunki – numer NIP, numer REGON |
| Drobne | Przy podaniu nieprawidłowej wg specyfikacji technicznej systemu wartości numeru NIP, REGON oraz kapitału zakładowego nie pojawia się żaden komunikat, brak informacji dla użytkownika o prawidłowej strukturze warunków przyjmowanych przez aplikację w poszczególnych elementach |

**6.9.** **Kryteria *wejścia***

|  |  |
| --- | --- |
| **Kryterium** | **Spełnienie** |
| Przygotowanie opisu testów | 90% zestawów testowych  90% scenariuszy testowych  70% przypadków testowych |
| Gotowość środowiska testowego | 80% |
| 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  80% pozytywnie zakończonych przypadków testowych, nie może być żadnych błędów krytycznych ani poważnych, mogą być 3 błędy Normal i maksymalnie 5 błędów drobnych |
| Przeprowadzenie i zakończenie z wynikiem pozytywnym testów systemowych | 100% wykonanych przypadków testowych  90% pozytywnie zakończonych przypadków testowych, nie może być żadnych błędów krytycznych ani poważnych, maksymalnie 3 błędy Normal i maksymalnie 5 błędów drobnych |
| Przeprowadzenie i zakończenie z wynikiem pozytywnym testów akceptacyjnych | 100% wykonanych przypadków testowych  90% pozytywnie zakończonych przypadków testowych, nie może być żadnych błędów krytycznych ani poważnych, maksymalnie 3 błędy Normal i maksymalnie 5 błędów drobnych |

**6.11. Kryteria *wstrzymania***

|  |  |
| --- | --- |
| **Kryterium** | **Spełnienie** |
| Wystąpienie błędów krytycznych | Aktualizacja systemu operacyjnego |
| Zamknięcie wszystkich aktywnych okien w bieżącej sesji po zainstalowaniu aktualizacji wtyczki Flash – restart eksploratora systemu operacyjnego |