Metody projektowania i wdrażania systemów informatycznych

Sprawozdanie z projektu

Kampa Agnieszka

WCY21KA1S1

80452

Spis treści

Temat

Aplikacja do wykonania ataku na algorytm XTR dla użytku firmy zajmującej się badaniem bezpieczeństwa informacji

Analiza biznesowa - opis fragmentu organizacji

Fragmentem firmy zajmującym się badaniem algorytmów kryptograficznych jest Laboratorium Badawcze Kryptologii.

Laboratorium Badawcze Kryptologii stanowi kluczowy element struktury firmy zajmującej się badaniem bezpieczeństwa informacji. Jego głównym zadaniem jest analiza, ocena i rozwijanie nowych metod kryptograficznych oraz badanie ich odporności na różnego rodzaju ataki. Laboratorium to składa się z zespołu wykwalifikowanych kryptologów, matematycznych analityków oraz inżynierów oprogramowania, którzy wspólnie pracują nad identyfikacją potencjalnych słabości w algorytmach kryptograficznych i opracowaniem skutecznych strategii ochrony przed nimi.

LBK odpowiada za szczegółową analizę różnorodnych algorytmów kryptograficznych stosowanych w branży. W ramach swoich działań zespół przeprowadza testy mające na celu ocenę bezpieczeństwa algorytmów oraz identyfikację możliwych wektorów ataków. W tym kontekście szczególną uwagę zwraca się na nowoczesne algorytmy i protokoły, które mogą być używane w przyszłości, a także na ocenie obecnych standardów kryptograficznych pod kątem ich odporności na ataki kwantowe.

Struktura organizacyjna LBK

1. Dyrektor LBK

a) Główny Kryptolog

- Zespół Kryptologów

b) Główny Inżynier Oprogramowania

- Zespół Inżynierów Oprogramowania

c) Analityk ds. Bezpieczeństwa

d) Koordynator Projektów

e) Asystent Badawczy

Diagram obiektów

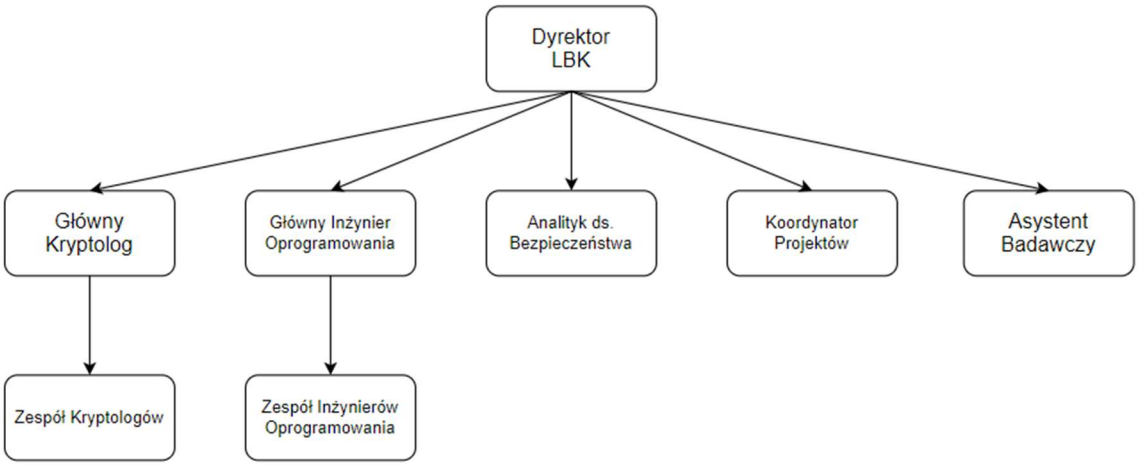
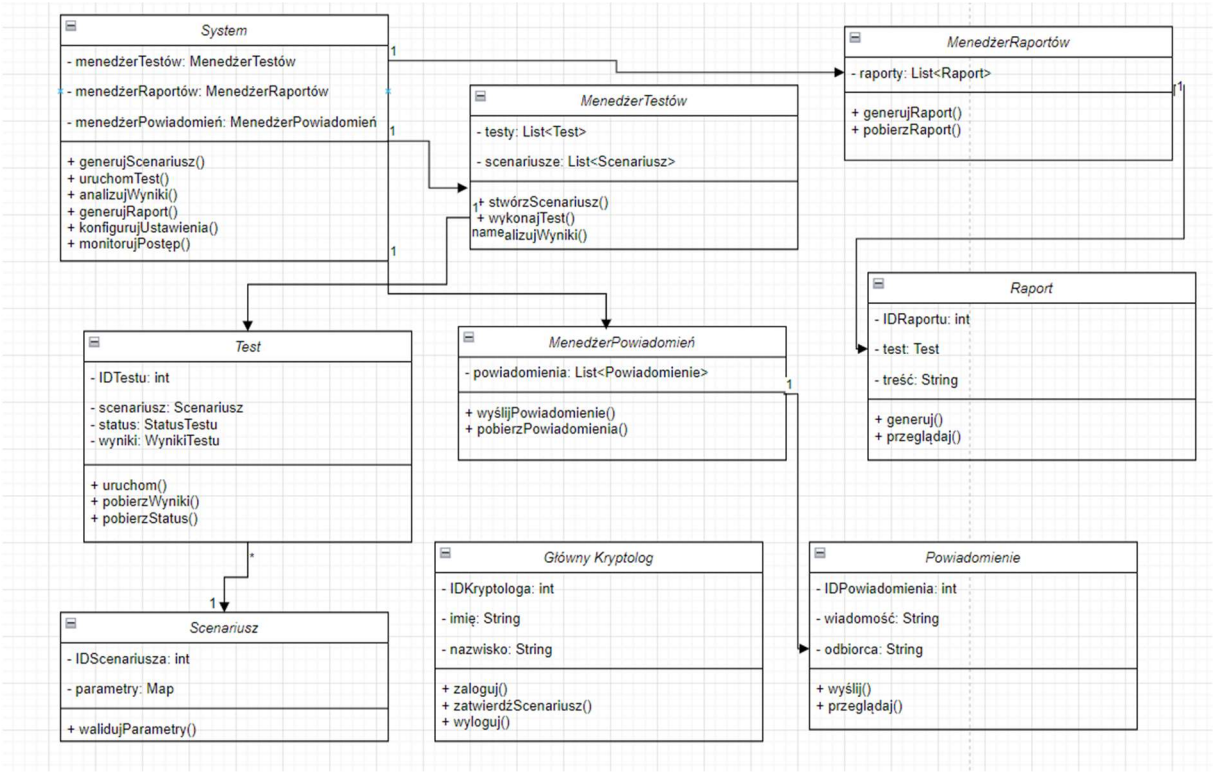


Diagram klas



Opis problemów i wyzwań

- Manualne przeprowadzanie ataków:

Przeprowadzanie kryptoanalizy ręcznie jest niezwykle czasochłonne i wymaga znacznych zasobów ludzkich. Każdy krok procesu musi być wykonany ręcznie, co może prowadzić do długotrwałych i nieefektywnych procedur. Kryptolodzy muszą ręcznie kodować różne warianty ataków, testować je i analizować wyniki. Bez automatyzacji, każdy drobny błąd może wymagać powtórzenia całego procesu, co dodatkowo zwiększa czas i koszty. Taka sytuacja jest nieefektywna i może prowadzić do opóźnień w projekcie oraz ograniczenia zakresu testów, które można przeprowadzić w dostępnym czasie.

- Ograniczona dokładność wyników:

Bez zaawansowanego narzędzia do automatyzacji ataków, analizy mogą być mniej dokładne. Ręczne testowanie i analiza mogą nie wychwycić subtelnych słabości w algorytmie XTR. Dedykowana aplikacja mogłaby systematycznie i precyzyjnie przeprowadzać setki, jeśli nie tysiące, testów w krótkim czasie, co jest niemal niemożliwe do osiągnięcia ręcznie. Dokładność wyników jest kluczowa dla zrozumienia bezpieczeństwa algorytmu oraz dla rozwoju skutecznych strategii kryptoanalizy. Brak narzędzia może prowadzić do niekompletnych lub niedokładnych wniosków, co z kolei wpływa na jakość raportów i rekomendacji dostarczanych klientom.

- Brak możliwości skalowania testów:

Skalowanie testów kryptograficznych jest kluczowe dla oceny odporności algorytmów na różne scenariusze ataków. Manualne przeprowadzanie testów jest nie tylko czasochłonne, ale także trudno skalowalne. Bez automatyzacji, testowanie algorytmu XTR na większych instancjach czy w różnych konfiguracjach staje się praktycznie niemożliwe. Dedykowana aplikacja umożliwia uruchamianie wielu testów równolegle i w różnych warunkach, co pozwala na uzyskanie bardziej reprezentatywnych wyników i lepsze zrozumienie potencjalnych zagrożeń. Brak możliwości skalowania ogranicza zespół do małych, ręcznie zarządzanych testów, co nie odzwierciedla rzeczywistych warunków i zagrożeń.

Wizja rozwiązania

Aby skutecznie przeciwdziałać problemom związanym z niemożnością przeprowadzania testów na dużą skalę bez dedykowanej aplikacji, można stworzyć innowacyjną aplikację wspierającą zespół badawczy w analizie algorytmu XTR. Można zawrzeć w niej następujące rozwiązania:

- Automatyzacja ataków: Aplikacja będzie umożliwiała automatyczne generowanie i przeprowadzanie różnorodnych ataków na algorytm XTR. Dzięki temu zespół będzie mógł szybko i skutecznie testować różne scenariusze ataków, co pozwoli na pełniejsze zrozumienie jego słabości.
- Skalowalność: Aplikacja będzie zoptymalizowana pod kątem wydajności i skalowalności, co umożliwi przeprowadzanie testów na dużą skalę. Dzięki temu zespół będzie mógł łatwo przetestować algorytm XTR na różnych instancjach i w różnych warunkach, co pozwoli na uzyskanie bardziej reprezentatywnych wyników.

- Integracja z symulatorami kwantowymi: W celu przeprowadzania testów wyżarzania kwantowego, aplikacja będzie integrowana z symulatorami kwantowymi.
- Monitoring postępu testów: Aplikacja będzie zapewniała narzędzia do monitorowania postępu testów i analizy wyników.
- Dokumentacja i raportowanie: Aplikacja będzie automatycznie generować raporty z przeprowadzonych testów oraz dokumentować wszystkie kroki analizy. Dzięki temu zespół będzie miał pełną kontrolę nad procesem testowania i łatwy dostęp do wyników.

Interesariusze

- Zespoły badawcze ds. kryptografii: Zespoły zajmujące się badaniami kryptograficznymi będą zainteresowane aplikacją, ponieważ umożliwia ona skuteczne przeprowadzanie testów na algorytm XTR i wyżarzanie kwantowe, co może prowadzić do odkrycia nowych słabości i zagrożeń.

- Firmy zajmujące się bezpieczeństwem informacji: Firmy oferujące usługi z zakresu bezpieczeństwa informacji mogą być zainteresowane aplikacją jako narzędziem wspierającym ich działalność.

- Organizacje rządowe i instytucje państwowe: Organizacje rządowe i instytucje państwowe odpowiedzialne za kwestie bezpieczeństwa narodowego mogą być zainteresowane aplikacją jako narzędziem wspierającym ich działania w zakresie kryptografii. Skuteczne testowanie algorytmów kryptograficznych może przyczynić się do wzmocnienia bezpieczeństwa komunikacji i danych wrażliwych.

- Przedsiębiorstwa z sektora IT: Przedsiębiorstwa z sektora IT, w tym dostawcy usług chmurowych, dostawcy oprogramowania, dostawcy usług sieciowych itp., mogą być zainteresowane aplikacją jako narzędziem wspierającym ich prace nad zapewnieniem bezpieczeństwa swoich produktów i usług.

- Uczelnie i instytucje akademickie: Uczelnie i instytucje akademickie prowadzące badania z zakresu kryptografii mogą być zainteresowane aplikacją jako narzędziem wspierającym ich prace badawcze. Aplikacja może być wykorzystywana do celów naukowych oraz w ramach programów dydaktycznych.

Analiza biznesowa - diagramy

Diagram AS-IS

Aktualnie atak na algorytm wykonuje się ręcznie co jest przyczyną wielu błędów oraz niedopatrzeń.

Ręczne tworzenie oprogramowania czy raportów jest niezwykle czasochłonne. Oprócz tego w celu zachowania wysokiej jakości wykonywanego ataku potrzebna jest duża ilość zasobów ludzkich.

Diagram:

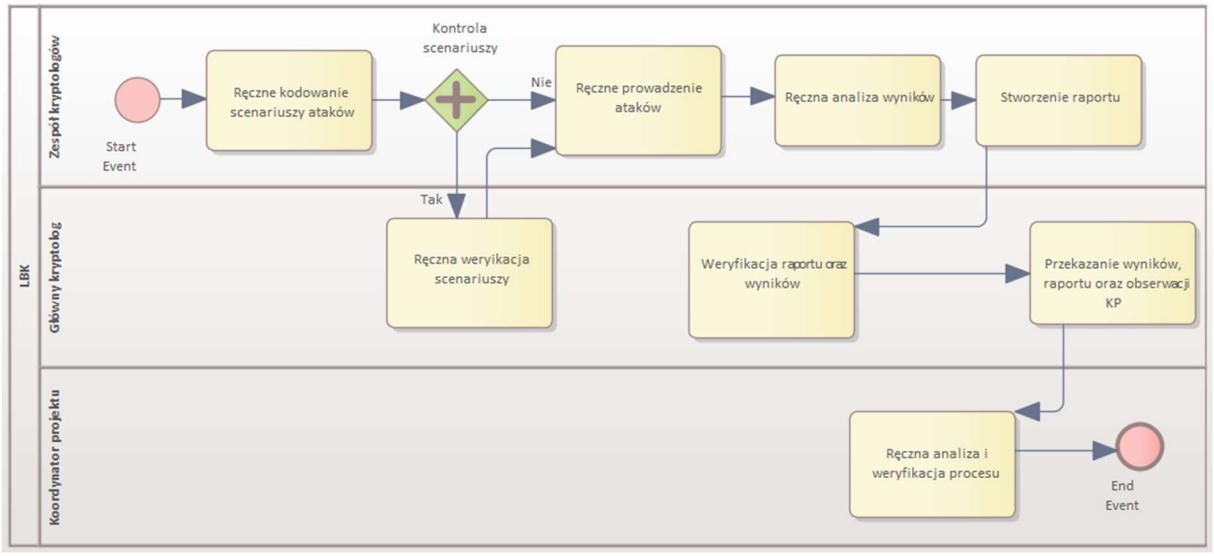
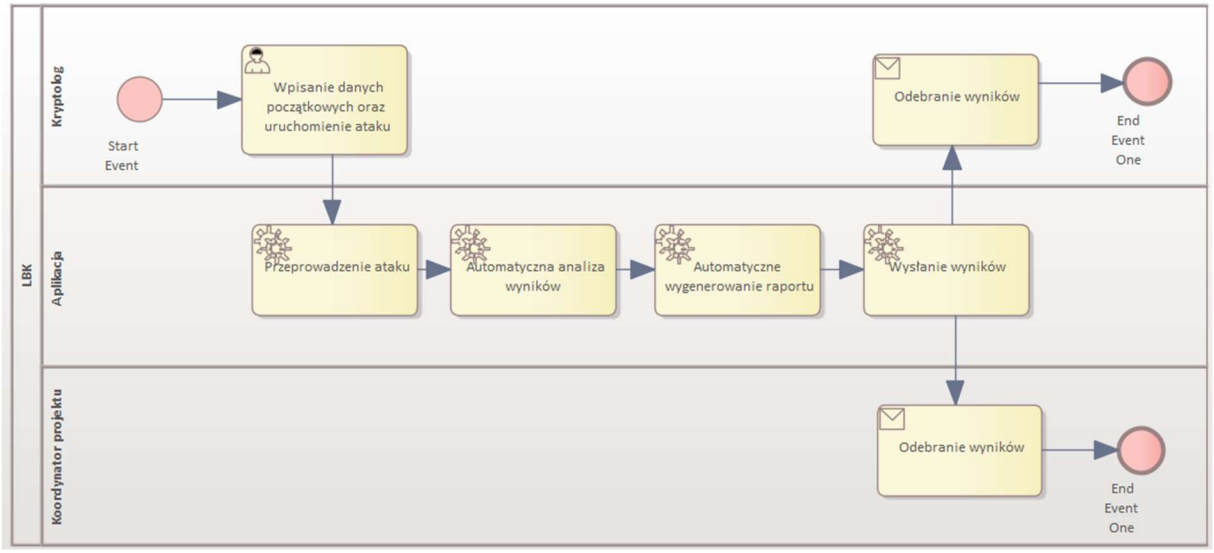


Diagram TO-BE

Wdrożenie aplikacji do automatycznego przeprowadzania ataku pozwoli na oszczędność zasobów ludzkich, czasu oraz zminimalizuję liczbę potencjalnych pomyłek i przeoczeń.

Diagram:



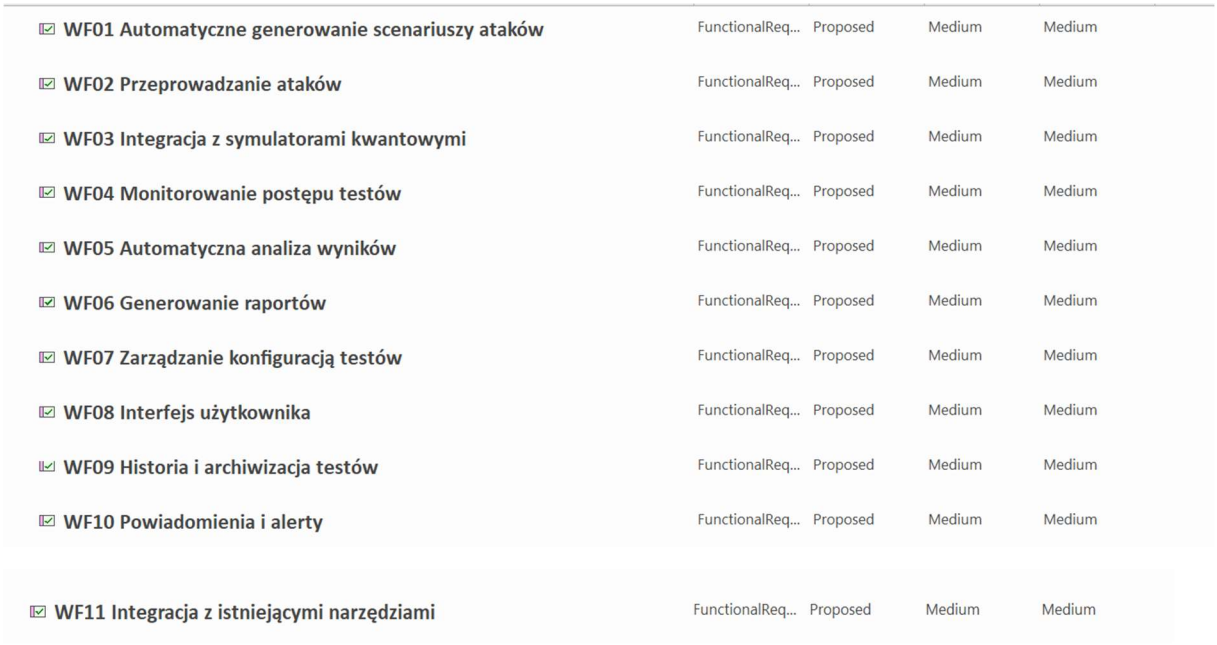
Analiza systemowa - wymagania

Wymagania funkcjonalne

Nr	Nazwa	Opis
WF01	Automatyczne generowanie scenariuszy ataków	Opis: System powinien generować różnorodne scenariusze ataków na algorytm XTR na podstawie zdefiniowanych parametrów i reguł. Zakres danych: Parametry ataków (np. typ ataku, intensywność, cel), reguły generowania. Reguły walidacji: Parametry muszą być zgodne z zdefiniowanymi typami ataków; intensywność ataku nie może przekraczać maksymalnych wartości określonych w specyfikacji.
WF02	Przeprowadzanie ataków	Opis: System powinien automatycznie przeprowadzać ataki na algorytm XTR zgodnie z wygenerowanymi scenariuszami. Zakres danych: Scenariusze ataków, logi z przeprowadzonych ataków, wyniki ataków. Reguły walidacji: Scenariusze muszą być poprawnie sformułowane; logi muszą zawierać pełną historię działań.
WF03	Integracja z symulatorami kwantowymi	Opis: System powinien mieć możliwość integracji z symulatorami kwantowymi w celu przeprowadzania testów wyżarzania kwantowego. Zakres danych: Interfejsy API symulatorów kwantowych, konfiguracje testów kwantowych. Reguły walidacji: Integracje muszą być zgodne z dokumentacją API symulatorów; konfiguracje muszą być weryfikowane pod kątem zgodności z wymaganiami testów kwantowych.
WF04	Monitorowanie postępu testów	Opis: System powinien zapewniać narzędzia do monitorowania w czasie rzeczywistym postępu testów kryptograficznych. Zakres danych: Aktualny status testów, procent ukończenia, logi bieżących działań. Reguły walidacji: Status musi być aktualizowany co najmniej raz na minutę; logi muszą być kompletne i czytelne.
WF05	Automatyczna analiza wyników	Opis: System powinien automatycznie zbierać i analizować dane z przeprowadzonych testów, identyfikując potencjalne słabości algorytmu XTR. Zakres danych: Wyniki testów, wykryte słabości, statystyki dotyczące ataków. Reguły walidacji: Analizy muszą być oparte na poprawnych i kompletnych danych z testów; wykryte słabości muszą być klasyfikowane według zdefiniowanych kategorii.
WF06	Generowanie raportów	Opis: System powinien automatycznie generować szczegółowe raporty z wynikami testów i analizą.

		Zakres danych: Wyniki testów, analiza słabości, zalecenia. Reguły walidacji: Raporty muszą zawierać wszystkie wymagane sekcje; format raportów musi być zgodny z ustalonym szablonem.
WF07	Zarządzanie konfiguracją testów	Opis: System powinien umożliwiać definiowanie i zarządzanie różnymi konfiguracjami testów, w tym parametrami wejściowymi i warunkami brzegowymi. Zakres danych: Konfiguracje testów, parametry wejściowe, warunki brzegowe. Reguły walidacji: Konfiguracje muszą być zapisywane zgodnie z określonym formatem; parametry wejściowe muszą mieścić się w zdefiniowanych zakresach.
WF08	Interfejs użytkownika	Opis: System powinien posiadać intuicyjny interfejs użytkownika umożliwiający łatwą nawigację i zarządzanie procesami testowymi. Zakres danych: Elementy interfejsu, dostępne opcje, konfiguracje użytkownika. Reguły walidacji: Interfejs musi być zgodny z zasadami użyteczności; wszystkie funkcje muszą być dostępne z poziomu interfejsu.
WF09	Historia i archiwizacja testów	Opis: System powinien umożliwiać przechowywanie i przeglądanie historii przeprowadzonych testów oraz archiwizację wyników. Zakres danych: Historia testów, wyniki testów, metadane testów. Reguły walidacji: Dane historyczne muszą być przechowywane przez określony czas; archiwizacja musi być zgodna z polityką przechowywania danych.
WF10	Powiadomienia i alerty	Opis: System powinien wysyłać powiadomienia i alerty w przypadku wykrycia krytycznych słabości algorytmu XTR lub zakończenia testów. Zakres danych: Typy powiadomień, kryteria wyzwalania, kanały komunikacji. Reguły walidacji: Powiadomienia muszą być wysyłane natychmiast po wykryciu krytycznej słabości; wszystkie powiadomienia muszą być rejestrowane.
WF11	Integracja z istniejącymi narzędziami	Opis: System powinien umożliwiać integrację z istniejącymi narzędziami używanymi przez Laboratorium Badawcze Kryptologii, takimi jak narzędzia do analizy kryptograficznej. Zakres danych: Interfejsy API narzędzi, konfiguracje integracji. Reguły walidacji: Integracje muszą być zgodne z dokumentacją API; dane wymieniane między

		systemami muszą być kompletne i zgodne z oczekiwaniami.



Wymagania niefunkcjonalne

Numer	Nazwa	Opis
WNF01	Bezpieczeństwo danych	System musi zapewniać bezpieczeństwo przechowywanych i przetwarzanych danych. Wszystkie dane muszą być szyfrowane zarówno w tranzycie, jak i w spoczynku; system musi być zgodny z najnowszymi standardami bezpieczeństwa (np. OWASP, GDPR).
WNF02	Wydajność	System powinien działać wydajnie, zapewniając szybki czas odpowiedzi na działania użytkownika. Czas odpowiedzi systemu nie może przekraczać 2 sekund dla 95% zapytań; czas przetwarzania testów nie może przekraczać 1 minuty dla 95% przypadków.
WNF03	Skalowalność	System musi być skalowalny, aby obsłużyć rosnącą liczbę użytkowników i zwiększone obciążenie testami. System musi obsługiwać co najmniej 100 jednoczesnych użytkowników i 1000 równoległych testów bez zauważalnego spadku wydajności.
WNF04	Dostępność	System musi być dostępny przez 99,9% czasu w ciągu miesiąca.
WNF05	Użyteczność	System powinien być łatwy w użyciu i intuicyjny dla użytkowników.
WNF06	Zgodność	System musi być zgodny z obowiązującymi standardami i regulacjami. System musi być zgodny z obowiązującymi standardami i regulacjami.

WNF01 Bezpieczeństwo danych	NonfunctionalRequirement	Proposed
WNF02 Wydajność	NonfunctionalRequirement	Proposed
WNF03 Skalowalność	NonfunctionalRequirement	Proposed
WNF04 Dostępność	NonfunctionalRequirement	Proposed
WNF05 Użyteczność	NonfunctionalRequirement	Proposed
WNF06 Zgodność	NonfunctionalRequirement	Proposed

Macierz wymagań i aktywności w Enterprise Architect

Source \ Target	activities::Automatyczna analiz	activities::Automatyczne wyger	activities::Odebranie wyników	activities::Odebranie wyników	activities::Przeprowadzenie ata	activities::Wpisanie danych poc	activities::Wpisanie danych poc	activities::Wysłanie wyników
Wymagania funkcjonalne::WF01 Automatyczne generowanie scenarius...						↑	↑	
Wymagania funkcjonalne::WF02 Przeprowadzanie ataków					↑			
Wymagania funkcjonalne::WF03 Integracja z symulatorami kwantowymi					↑	↑	↑	
Wymagania funkcjonalne::WF04 Monitorowanie postępu testów	↑	↑						
Wymagania funkcjonalne::WF05 Automatyczna analiza wyników	↑							
Wymagania funkcjonalne::WF06 Generowanie raportów		↑						
Wymagania funkcjonalne::WF07 Zarządzanie konfiguracją testów					↑		↑	↑
Wymagania funkcjonalne::WF08 Interfejs użytkownika	↑	↑	↑	↑	↑	↑	↑	↑
Wymagania funkcjonalne::WF09 Historia i archiwizacja testów	↑	↑						↑
Wymagania funkcjonalne::WF10 Powiadomienia i alerty			↑	↑				↑
Wymagania funkcjonalne::WF11 Integracja z istniejącymi narzędziami					↑	↑	↑	

Analiza systemowa – przypadki użycia

Przypadki użycia dla każdego wymagania

WF01: Automatyczne generowanie scenariuszy ataków

Przypadek użycia 1

Nazwa: Generowanie podstawowych scenariuszy ataków
Aktorzy: Kryptolog
Scenariusz:

1. Kryptolog loguje się do systemu.
2. Kryptolog wybiera opcję generowania nowych scenariuszy ataków.
3. Kryptolog definiuje parametry ataków (typ ataku, intensywność, cel).
4. System waliduje parametry.
5. System generuje scenariusze ataków na podstawie wprowadzonych parametrów.
6. System wyświetla wygenerowane scenariusze ataków.

Przypadek użycia 2

Nazwa: Edycja istniejących scenariuszy ataków
Aktorzy: Inżynier oprogramowania
Scenariusz:

1. Inżynier oprogramowania loguje się do systemu.
2. Inżynier wybiera opcję edycji istniejących scenariuszy ataków.
3. Inżynier przegląda listę wygenerowanych scenariuszy.
4. Inżynier wybiera scenariusz do edycji.
5. Inżynier modyfikuje parametry scenariusza.
6. System waliduje nowe parametry.
7. System zapisuje zaktualizowany scenariusz ataku.

WF02: Przeprowadzanie ataków

Przypadek użycia 1

Nazwa: Wykonanie pojedynczego ataku
Aktorzy: Kryptolog
Scenariusz:

1. Kryptolog loguje się do systemu.
2. Kryptolog wybiera wygenerowany scenariusz ataku.
3. Kryptolog uruchamia atak.
4. System przeprowadza atak na algorytm XTR zgodnie z wybranym scenariuszem.

5. System rejestruje logi z przeprowadzonych działań.
6. System wyświetla wyniki ataku.

Przypadek użycia 2

Nazwa: Przeprowadzenie serii ataków
Aktorzy: Asystent badawczy
Scenariusz:

1. Asystent badawczy loguje się do systemu.
2. Asystent badawczy wybiera opcję uruchomienia serii ataków.
3. Asystent badawczy wybiera kilka scenariuszy ataków.
4. System przeprowadza serię ataków zgodnie z wybranymi scenariuszami.
5. System rejestruje logi z przeprowadzonych działań.
6. System wyświetla zbiorcze wyniki przeprowadzonych ataków.

WF03: Integracja z symulatorami kwantowymi

Przypadek użycia 1

Nazwa: Konfiguracja integracji z symulatorem kwantowym
Aktorzy: Główny inżynier oprogramowania
Scenariusz:

1. Główny inżynier oprogramowania loguje się do systemu.
2. Główny inżynier oprogramowania przechodzi do sekcji konfiguracji integracji.
3. Główny inżynier oprogramowania wprowadza dane API symulatora kwantowego.
4. System waliduje dane API.
5. System zapisuje konfigurację integracji.

Przypadek użycia 2

Nazwa: Uruchomienie testu wyżarzania kwantowego
Aktorzy: Analityk ds. bezpieczeństwa
Scenariusz:

1. Analityk ds. bezpieczeństwa loguje się do systemu.
2. Analityk ds. bezpieczeństwa wybiera opcję uruchomienia testu wyżarzania kwantowego.
3. Analityk ds. bezpieczeństwa konfiguruje parametry testu kwantowego.
4. System waliduje parametry testu.
5. System uruchamia test z użyciem symulatora kwantowego.
6. System rejestruje wyniki testu.
7. System wyświetla wyniki testu kwantowego.

WF04: Monitorowanie postępu testów

Przypadek użycia 1

Nazwa: Monitorowanie pojedynczego testu
Aktorzy: Kryptolog
Scenariusz:

1. Kryptolog loguje się do systemu.
2. Kryptolog wybiera opcję monitorowania bieżących testów.
3. System wyświetla listę aktywnych testów.
4. Kryptolog wybiera test do monitorowania.
5. System wyświetla aktualny status testu, procent ukończenia i logi działań.

Przypadek użycia 2

Nazwa: Monitorowanie serii testów
Aktorzy: Koordynator projektów
Scenariusz:

1. Koordynator projektów loguje się do systemu.
2. Koordynator projektów przechodzi do sekcji monitorowania testów.
3. System wyświetla zbiorcze informacje o wszystkich aktywnych testach.
4. Koordynator projektów przegląda statusy poszczególnych testów.
5. System aktualizuje dane co minutę, umożliwiając bieżące monitorowanie.

WF05: Automatyczna analiza wyników

Przypadek użycia 1

Nazwa: Analiza wyników pojedynczego ataku
Aktorzy: Asystent badawczy
Scenariusz:

1. Asystent badawczy loguje się do systemu.
2. Asystent badawczy wybiera opcję analizy wyników.
3. Asystent badawczy wybiera wyniki konkretnego ataku.
4. System automatycznie analizuje wyniki ataku.
5. System identyfikuje potencjalne słabości algorytmu XTR.
6. System wyświetla wyniki analizy wraz z wykrytymi słabościami.

Przypadek użycia 2

Nazwa: Zbiorcza analiza wyników serii ataków
Aktorzy: Kryptolog
Scenariusz:

1. Kryptolog loguje się do systemu.
2. Kryptolog wybiera opcję zbiorczej analizy wyników.
3. Kryptolog wybiera zestaw wyników z serii ataków.
4. System automatycznie analizuje dane z przeprowadzonych testów.
5. System identyfikuje i klasyfikuje słabości algorytmu.
6. System wyświetla zbiorczą analizę wraz z statystykami dotyczącymi ataków.

WF06: Generowanie raportów

Przypadek użycia 1

Nazwa: Generowanie raportu z pojedynczego testu
Aktorzy: Analityk ds. bezpieczeństwa
Scenariusz:

1. Analityk ds. bezpieczeństwa loguje się do systemu.
2. Analityk ds. bezpieczeństwa wybiera opcję generowania raportu.
3. Analityk ds. bezpieczeństwa wybiera wyniki konkretnego testu.
4. System automatycznie generuje raport z wynikami testu i analizą.
5. System wyświetla raport do przeglądu.
6. Analityk ds. bezpieczeństwa zapisuje raport.

Przypadek użycia 2

Nazwa: Generowanie zbiorczego raportu z serii testów
Aktorzy: Koordynator projektów
Scenariusz:

1. Koordynator projektów loguje się do systemu.
2. Koordynator projektów wybiera opcję generowania zbiorczego raportu.
3. Koordynator projektów wybiera wyniki serii testów.
4. System automatycznie generuje zbiorczy raport z analizą wyników.
5. System wyświetla raport do przeglądu.
6. Koordynator projektów zapisuje raport.

WF07: Zarządzanie konfiguracją testów

Przypadek użycia 1

Nazwa: Definiowanie nowej konfiguracji testu
Aktorzy: Główny inżynier oprogramowania
Scenariusz:

1. Główny inżynier oprogramowania loguje się do systemu.
2. Główny inżynier oprogramowania wybiera opcję definiowania konfiguracji testu.
3. Główny inżynier oprogramowania wprowadza parametry wejściowe i warunki brzegowe.
4. System waliduje wprowadzone dane.
5. System zapisuje nową konfigurację testu.

Przypadek użycia 2

Nazwa: Edycja istniejącej konfiguracji testu
Aktorzy: Analityk ds. bezpieczeństwa
Scenariusz:

1. Analityk ds. bezpieczeństwa loguje się do systemu.
2. Analityk ds. bezpieczeństwa wybiera opcję zarządzania konfiguracjami testów.
3. Analityk ds. bezpieczeństwa przegląda listę istniejących konfiguracji.
4. Analityk ds. bezpieczeństwa wybiera konfigurację do edycji.
5. Analityk ds. bezpieczeństwa modyfikuje parametry i warunki brzegowe.
6. System waliduje nowe parametry.
7. System zapisuje zaktualizowaną konfigurację testu.

WF08: Interfejs użytkownika

Przypadek użycia 1

Nazwa: Nawigacja po interfejsie
Aktorzy: Kryptolog
Scenariusz:

1. Kryptolog loguje się do systemu.
2. Kryptolog przegląda główny pulpit nawigacyjny.
3. Kryptolog wybiera zakładki dotyczące generowania ataków, monitorowania testów i analizy wyników.
4. System umożliwia łatwą nawigację pomiędzy różnymi funkcjami.

Przypadek użycia 2

Nazwa: Konfiguracja ustawień użytkownika
Aktorzy: Koordynator projektów
Scenariusz:

1. Koordynator projektów loguje się do systemu.
2. Koordynator projektów przechodzi do sekcji ustawień użytkownika.
3. Koordynator projektów modyfikuje ustawienia dotyczące powiadomień i preferencji wyświetlania.
4. System zapisuje zaktualizowane ustawienia.
5. System dostosowuje interfejs użytkownika zgodnie z preferencjami.

WF09: Historia i archiwizacja testów

Przypadek użycia 1

Nazwa: Przeglądanie historii testów
Aktorzy: Asystent badawczy
Scenariusz:

1. Asystent badawczy loguje się do systemu.
2. Asystent badawczy wybiera opcję przeglądania historii testów.
3. System wyświetla listę przeprowadzonych testów.
4. Asystent badawczy przegląda szczegóły wybranego testu, w tym wyniki i logi.

Przypadek użycia 2

Nazwa: Archiwizacja wyników testów
Aktorzy: Kryptolog
Scenariusz:

1. Kryptolog loguje się do systemu.
2. Kryptolog wybiera opcję archiwizacji testów.
3. System wyświetla listę zakończonych testów.
4. Kryptolog wybiera testy do archiwizacji.
5. System przeprowadza archiwizację wybranych testów zgodnie z polityką przechowywania danych.

WF10: Powiadomienia i alerty

Przypadek użycia 1

Nazwa: Konfiguracja powiadomień
Aktorzy: Analityk ds. bezpieczeństwa

Scenariusz:

1. Analityk ds. bezpieczeństwa loguje się do systemu.
2. Analityk ds. bezpieczeństwa przechodzi do sekcji konfiguracji powiadomień.
3. Analityk ds. bezpieczeństwa definiuje typy powiadomień i kryteria wyzwalania.
4. System zapisuje konfigurację powiadomień.

Przypadek użycia 2

Nazwa: Odbieranie powiadomień o zakończeniu testu
Aktorzy: Koordynator projektów
Scenariusz:

1. Koordynator projektów loguje się do systemu.
2. Koordynator projektów odbiera powiadomienie o zakończeniu testu.
3. System wyświetla szczegóły zakończonego testu i wyniki.
4. Koordynator projektów przegląda wyniki testu i podejmuje dalsze działania.

WF11: Integracja z istniejącymi narzędziami

Przypadek użycia 1

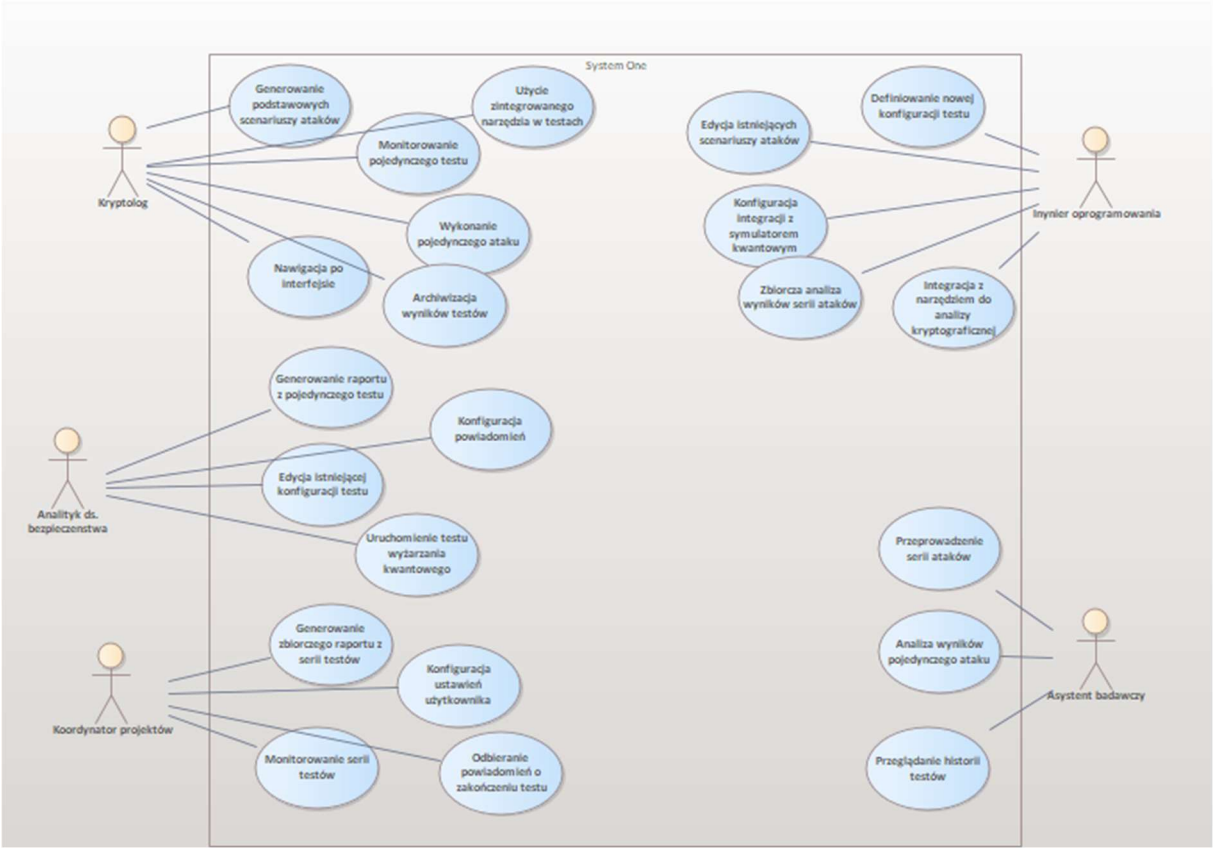
Nazwa: Integracja z narzędziem do analizy kryptograficznej
Aktorzy: Główny inżynier oprogramowania
Scenariusz:

1. Główny inżynier oprogramowania loguje się do systemu.
2. Główny inżynier oprogramowania wybiera opcję integracji z istniejącym narzędziem do analizy kryptograficznej.
3. Główny inżynier oprogramowania wprowadza dane API narzędzia.
4. System waliduje dane API.
5. System konfiguruje integrację z narzędziem do analizy kryptograficznej.
6. System umożliwia wymianę danych między systemami.

Przypadek użycia 2

Nazwa: Użycie zintegrowanego narzędzia w testach
Aktorzy: Kryptolog
Scenariusz:

1. Kryptolog loguje się do systemu.
2. Kryptolog wybiera opcję użycia zintegrowanego narzędzia do analizy kryptograficznej w testach.
3. System przekazuje wyniki testów do zintegrowanego narzędzia.
4. System odbiera i wyświetla wyniki analizy z narzędzia.
5. Kryptolog przegląda i analizuje wyniki.



Macierz przypadków użycia i wymagań w Enterprise Architect

Source \ Target	Use Cases::Analiza wyników po	Use Cases::Archiwizacja wyniki	Use Cases::Definiowanie nowej	Use Cases::Edycja istniejących	Use Cases::Generowanie podst	Use Cases::Generowanie rapor	Use Cases::Generowanie zbior	Use Cases::Integracja z narzęd	Use Cases::Konfiguracja integr	Use Cases::Konfiguracja powia	Use Cases::Konfiguracja ustawi	Use Cases::Monitorowanie poj	Use Cases::Monitorowanie seri	Use Cases::Nawigacja po interf	Use Cases::Odbieranie powiadc	Use Cases::Przeglądanie histori	Use Cases::Przeprowadzenie se	Use Cases::Uruchomienie testu	Use Cases::Użycie zintegrowan	Use Cases::Zbiorcza analiza wy
Wymagania funkcjonalne::WF01 Automatyczne generowanie scen...				↑	↑															
Wymagania funkcjonalne::WF02 Przeprowadzanie ataków																	↑			
Wymagania funkcjonalne::WF03 Integracja z symulatorami kwanto...									↑									↑		
Wymagania funkcjonalne::WF04 Monitorowanie postępu testów												↑	↑							
Wymagania funkcjonalne::WF05 Automatyczna analiza wyników	↑																			↑
Wymagania funkcjonalne::WF06 Generowanie raportów						↑	↑													
Wymagania funkcjonalne::WF07 Zarządzanie konfiguracją testów			↑																	
Wymagania funkcjonalne::WF08 Interfejs użytkownika											↑			↑						
Wymagania funkcjonalne::WF09 Historia i archiwizacja testów		↑														↑				
Wymagania funkcjonalne::WF10 Powiadomienia i alerty										↑					↑					
Wymagania funkcjonalne::WF11 Integracja z istniejącymi narzędzia...								↑											↑	

Architektura

Architektura fizyczna

Warstwa 1: Infrastruktura fizyczna

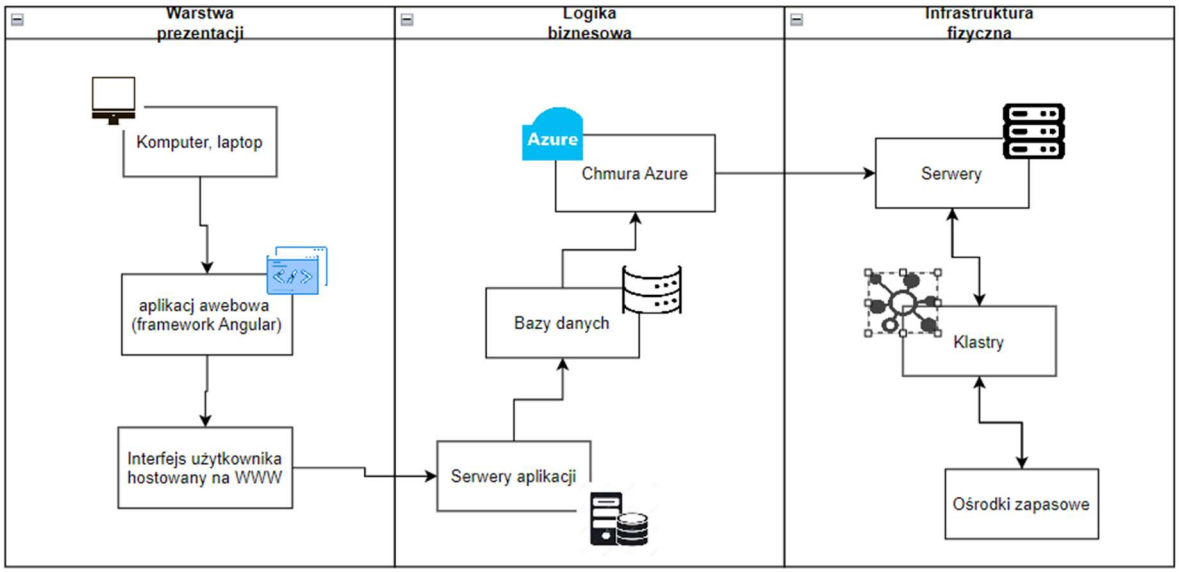
- Serwery: Dedykowane serwery fizyczne lub serwery w chmurze (AWS, Azure, GCP).
- Klastry: W przypadku dużego obciążenia, użycie klastrów serwerowych do skalowania poziomego.
- Ośrodki podstawowe i zapasowe: Główne centrum danych oraz zapasowe centrum danych w przypadku awarii.

Warstwa 2: Logika biznesowa

- Serwery aplikacji: Serwery do hostowania aplikacji (np. Tomcat, JBoss).
- Bazy danych: Relacyjne bazy danych (np. MySQL, PostgreSQL) oraz NoSQL (np. MongoDB) do przechowywania wyników testów i logów.
- Usługi przetwarzania: Mikroserwisy lub funkcje w chmurze (AWS Lambda, Azure Functions) do obsługi logiki biznesowej.

Warstwa 3: Warstwa prezentacji

- Interfejs użytkownika: Serwery WWW do hostowania frontendu (np. Nginx, Apache).
- Aplikacje webowe: Frameworki front-endowe (np. React, Angular) do interakcji z użytkownikami.
- Urządzenia końcowe: Komputery, laptopy, tablety do dostępu do aplikacji.



Architektura logiczna

Warstwa 1: Logika aplikacji

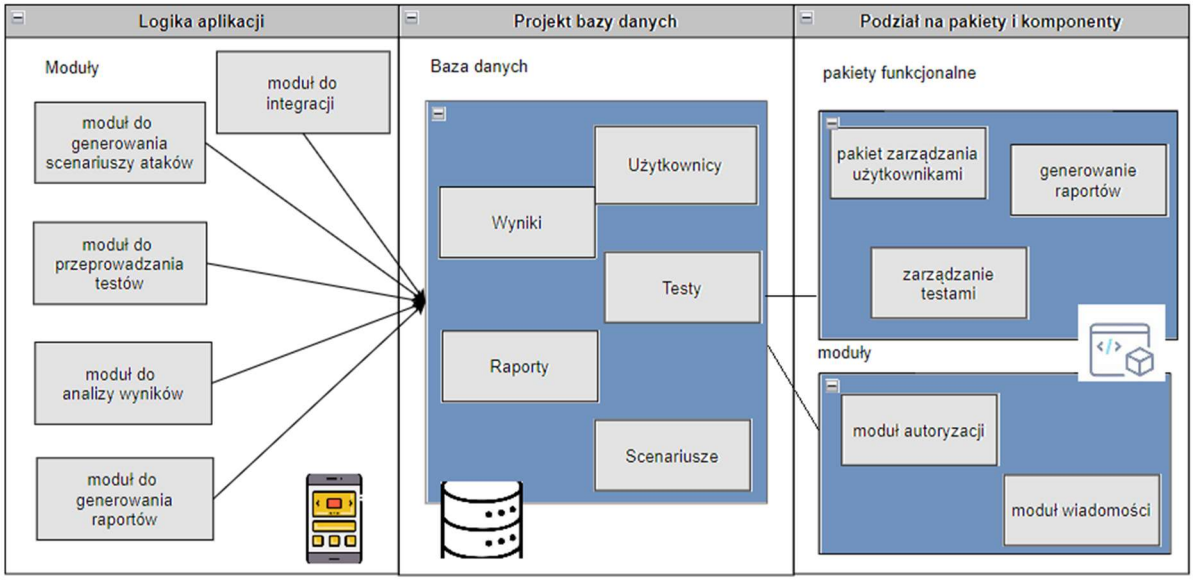
- Komponenty przetwarzania: Moduły do generowania scenariuszy ataków, przeprowadzania testów, analizy wyników i generowania raportów.
- Moduł integracji: Komponenty do integracji z symulatorami kwantowymi i innymi narzędziami kryptograficznymi.

Warstwa 2: Projekt bazy danych

- Tabele i encje: Projekt tabel i encji, w tym tabele dla użytkowników, testów, scenariuszy, wyników i raportów.
- Relacje i klucze: Relacje między tabelami, klucze główne i obce.

Warstwa 3: Podział na pakiety i komponenty

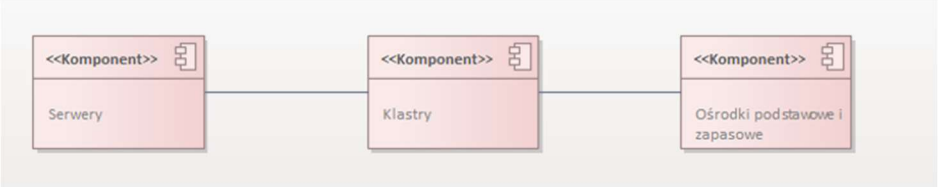
- Pakiety funkcjonalne: Podział na pakiety odpowiadające za różne funkcjonalności, np. zarządzanie użytkownikami, zarządzanie testami, generowanie raportów.
- Reusable komponenty: Moduły i biblioteki, które mogą być używane w różnych częściach aplikacji, np. moduł autoryzacji.



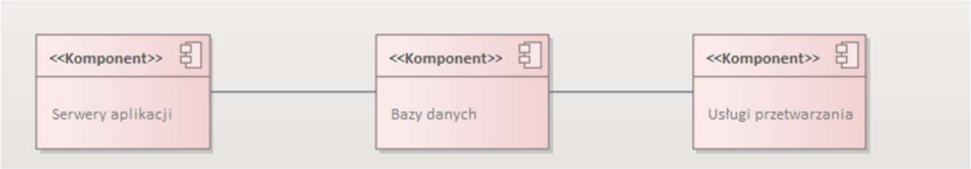
Diagramy

Architektura fizyczna

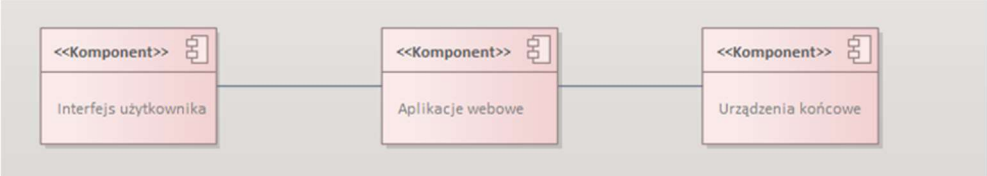
Warstwa 1: Infrastruktura fizyczna



Warstwa 2: Logika biznesowa



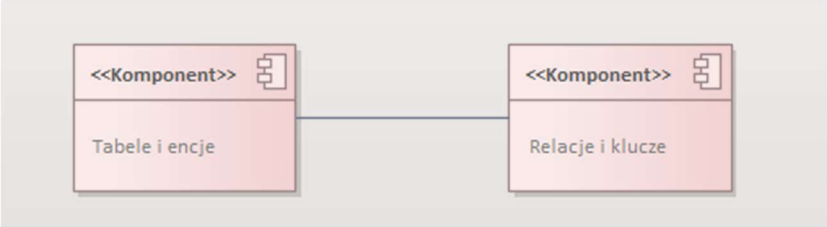
Warstwa 3: Warstwa prezentacji



Architektura logiczna

Warstwa 1: Logika aplikacji

Warstwa 2: Projekt bazy danych



Warstwa 3: Podział na pakiety i komponenty