WYDZIAŁ ELEKTRONIKI
I TECHNIK INFORMACYJNYCH

Eventix

Dokumentacja projektu

Julia Jodczyk, Daniel Kobiałka, Maksymilian Łazarski, Hubert Soroka, Bartłomiej Ściseł, Maciej Tymoftyjewicz

Analiza i projektowanie systemów informacyjnych 2024L

Spis treści

1. Opis systemu 3
2. Analiza UX 4
2.1 Aktorzy 4
2.2 Przypadki użycia 5
2.2.1 Lista 5
2.2.2 Użytkownik prywatny 6
2.2.3 Organizator 12
2.2.4 Administrator 18
2.2.5 Uniwersalne 20
2.3 Słownik pojęć 21
3. Projekt systemu 22
3.1 Ogólny model architektury 22
3.2 Technologie 23

1. Opis systemu

Celem projektu jest stworzenie systemu, będącego platformą umożliwiającą sprzedaż biletów na wydarzenia organizowane przez podmioty zewnętrzne. Korzystać z niego będą zarówno użytkownicy prywatni, dokonujący zakupu biletów jak i przedstawiciele firm, ogłaszający swoje wydarzenia i prowadzący sprzedaż biletów w ramach systemu.

2. Analiza UX

2.1 Aktorzy

Użytkownik prywatny

Osoba indywidualna, która poszukuje biletów na wydarzenia kulturalne, sportowe, rozrywkowe lub inne imprezy. Klient ten może być zarówno stałym użytkownikiem naszej platformy, jak i nowym użytkownikiem, który korzysta z niej po raz pierwszy.

Organizator

Organizator w naszym systemie sprzedaży biletów to osoba lub instytucja odpowiedzialna za tworzenie i zarządzanie wydarzeniami, na które oferowane są bilety. Organizatorzy mogą być zarówno profesjonalnymi firmami eventowymi, jak i indywidualnymi osobami lub grupami, które chcą zorganizować koncert, festiwal, konferencję lub inne wydarzenie. Aby móc korzystać z systemu organizator musi złożyć specjalny wniosek, który musi zostać zaakceptowany przez osoby zarządzające systemem.

Administrator

Administrator w naszym systemie sprzedaży biletów ma uprawnienia do zarządzania całą platformą z poziomu administracyjnego panelu kontrolnego. Jego głównym celem jest utrzymanie systemu we właściwym stanie, zarządzanie jego wewnętrznymi ustawieniami oraz wsparcie dla organizatorów i użytkowników końcowych.

2.2 Przypadki użycia

2.2.1 Lista

Użytkownik prywatny:

- UC1: Przeglądanie listy wydarzeń
- UC2: Wyszukanie wydarzenia
- UC3: Wyświetlenie strony wydarzenia
- UC4: Stworzenie prywatnego konta
- UC5: Zakup biletu
- UC6: Zwrot biletu
- UC7: Wyświetlanie historii zakupionych biletów

Organizator

- UC8: Stworzenie wniosku o utworzenie profilu
- UC9: Stworzenie wydarzenia
- UC10: Edycja danych wydarzenia
- UC11: Anulowanie wydarzenia
- UC12: Przeglądanie statystyk wydarzenia
- UC13: Stworzenie wniosku o usunięcie profilu.
- UC14: Dodanie lokalizacji

Administrator

- UC15: Rejestrowanie organizatora
- UC16: Usuwanie organizatora

Uniwersalne

- UC17: Zalogowanie się na konto

2.2.2 Użytkownik prywatny

Przypisane przypadki użycia:

- UC1: Przeglądanie listy wydarzeń
- UC2: Wyszukanie wydarzenia
- UC3: Wyświetlenie strony wydarzenia
- UC4: Stworzenie prywatnego konta
- UC5: Zakup biletu
- UC6: Zwrot biletu
- UC7: Wyświetlanie historii zakupionych biletów

Nazwa	**Przeglądanie listy wydarzeń**
ID	UC1
Aktor główny	Użytkownik prywatny
Aktorzy	Użytkownik prywatny
Priorytet	Główne
Opis	Użytkownik może przeglądać wszystkie dostępne wydarzenia
Cel	Pozwolenie użytkownikowi na poznanie pełnej oferty planowanych wydarzeń
Wyzwalanie	Użytkownik postanawia przejrzeć listę wydarzeń
War. początkowe	• System posiada niezerową liczbę aktywnych wydarzeń
War. końcowe	• Użytkownik przejrzał listę wydarzeń
W. funkcjonalne	• System wyświetla wszystkie aktualne wydarzenia zarejestrowane w systemie
W. niefunkcjonalne	• Widok wydarzeń powinien być estetyczny
Scenariusz	1. Użytkownik otwiera widok listy wydarzeń 2. System wyświetla ofertę wydarzeń

Nazwa	**Wyszukanie wydarzenia**
ID	UC2
Aktor główny	Użytkownik prywatny
Aktorzy	Użytkownik prywatny
Priorytet	Opcjonalne
Opis	Użytkownik może wyszukać po nazwie konkretne wydarzenie
Cel	Pozwolenie użytkownikowi na szybsze znajdowanie interesujących go wydarzeń
Wyzwalanie	Użytkownik postanawia wyszukać wydarzenie w systemie
War. początkowe	• Użytkownik jest w stanie przeglądać listę wydarzeń (UC1) • System posiada niezerową liczbę aktywnych wydarzeń
War. końcowe	• Użytkownik pomyślnie wyszukał wydarzenie
W. funkcjonalne	• System posiada wyszukiwarkę umożliwiającą wyszukanie wydarzenia
W. niefunkcjonalne	• Wyszukiwarka powinna być umiejscowiona w widocznym i intuicyjnym miejscu na stronie listy wydarzeń
Scenariusz	1. Użytkownik wprowadza nazwę konkretnego wydarzenia 2. System aktualizuje prezentowaną listę wydarzeń [E1: Wyszukanie nie zwraca żadnych wyników] .
Scenariusz wyjątku	E1. Wyszukanie nie zwraca żadnych wyników a. Nie zostało odnalezione wydarzenie zawierające podaną frazę. System informuje użytkownika o braku wyników wyszukiwania

Nazwa	**Wyświetlenie strony wydarzenia**
ID	UC3
Aktor główny	Użytkownik prywatny
Aktorzy	Użytkownik prywatny
Priorytet	Główne
Opis	Użytkownik może wyświetlić stronę poświęconą konkretnemu wydarzeniu
Cel	Umożliwienie użytkownikowi poznania szczegółów dotyczących wydarzenia
Wyzwalanie	Użytkownik postanawia dowiedzieć się więcej o konkretnym wydarzeniu
War. początkowe	• Użytkownik jest w stanie przeglądać listę wydarzeń (UC1) • System posiada niezerową liczbę aktywnych wydarzeń
War. końcowe	• Użytkownik pomyślnie wyświetlił stronę wydarzenia
W. funkcjonalne	• System wyświetla szczegółowe dane o wydarzeniu
W. niefunkcjonalne	• Widok wydarzenia powinien być estetyczny • Dane wydarzenia powinny być przedstawione w sposób czytelny
Scenariusz	1. Użytkownik znajduje interesujące go wydarzenie w liście wydarzeń i wybiera je 2. System wyświetla użytkownikowi dane dotyczące wybranego wydarzenia

Nazwa	**Stworzenie prywatnego konta**
ID	UC4
Aktor główny	Użytkownik prywatny
Aktorzy	Użytkownik prywatny
Priorytet	Główne
Opis	Użytkownik prywatny tworzy konto w systemie
Cel	Umożliwienie użytkownikowi szerszej interakcji z systemem
Wyzwalanie	Użytkownik chce uzyskać dostęp do akcji wymagających posiadania konta
War. początkowe	• Użytkownik nie jest zalogowany
War. końcowe	• Użytkownik pomyślnie założył konto • Użytkownik może zalogować się na konto
W. funkcjonalne	• System pozwala na założenie konta • System dokonuje walidacji • System informuje o błędach lub sukcesach
W. niefunkcjonalne	• Formularz musi być prosty i przejrzysty
Scenariusz	1. Użytkownik wchodzi na dowolną stronę aplikacji 2. Użytkownik wchodzi w widok rejestracji 3. System wyświetla widok formularza rejestracji 4. Użytkownik wypełnia formularz 5. Użytkownik przesyła formularz 6. System dokonuje walidacji danych [E1: Dane nie przechodzą walidacji] 7. System tworzy konto użytkownika 8. System potwierdza rejestrację 9. System przekierowuje użytkownika do widoku logowania
Scenariusz wyjątku	E1: Dane nie przechodzą walidacji a. Użytkownik podaje dane, które nie przechodzą walidacji lub są niepełne b. System wyświetla użytkownikowi informację o błędzie

Nazwa	Zakup biletu
ID	UC5
Aktor główny	Użytkownik prywatny
Aktorzy	Użytkownik prywatny
Priorytet	Główne
Opis	Użytkownik prywatny dokonuje zakupu biletu za pośrednictwem systemu
Cel	Umożliwienie użytkownikowi zdalnego kupna biletów
Wyzwalanie	Użytkownik postanawia kupić bilet na interesujące go wydarzenie
War. początkowe	• Użytkownik posiada konto (UC4) • Użytkownik jest zalogowany (UC17) • System posiada niezerową liczbę aktywnych wydarzeń • Dostępne są bilety na wybrane wydarzenie
War. końcowe	• Użytkownik pomyślnie zakupił bilet
W. funkcjonalne	• System generuje rezerwację biletu i udostępnia sposób płatności • System wyświetla informacje o opcjach biletu
W. niefunkcjonalne	• Interfejs kupna biletu powinien być przejrzysty • Transakcja kupna biletu musi być bezpieczna
Scenariusz	1. Użytkownik wchodzi na stronę wydarzenia (UC3) 2. Użytkownik wybiera opcję zakupu biletu 3. System wyświetla widok zamówienia biletu, w tym jego warianty (jeśli istnieją) 4. Użytkownik wybiera bilet i potwierdza zakup 5. Użytkownik realizuje płatność [E1: Błąd w realizacji] 6. System potwierdza dokonanie zakupu 7. System wysyła użytkownikowi jego kopię biletu
Scenariusz wyjątku	E1: Błąd w realizacji a. Użytkownikowi nie udało się sfinalizować transakcji b. System informuje o błędzie transakcji i pozwala na powrót do strony wydarzenia

Nazwa	Zwrot biletu
ID	UC6
Aktor główny	Użytkownik prywatny
Aktorzy	Użytkownik prywatny
Priorytet	Drugorzędne
Opis	Użytkownik anuluje swoją rezerwację biletu i otrzymuje zwrot pieniędzy
Cel	Umożliwienie użytkownikowi sposobu na zwrot biletu
Wyzwalanie	Użytkownik postanawia zwrócić zakupiony w systemie bilet
War. początkowe	• Użytkownik posiada konto (UC4) • Użytkownik jest zalogowany (UC17) • Użytkownik zakupił bilet na wydarzenie (UC5) • Wydarzenie jeszcze nie odbyło się • Wydarzenie nie zostało anulowane
War. końcowe	• Pomyślnie zwrócono bilet • Następuje zwrot pieniędzy do użytkownika
W. funkcjonalne	• System pozwala na usuwanie rezerwacji i zwrot pieniędzy
W. niefunkcjonalne	• Zwrot pieniędzy powinien być bezpieczny • Interfejs zwrotu powinien być prosty
Scenariusz	1. Użytkownik przechodzi do strony z listą zakupionych biletów 2. Użytkownik wybiera bilet do usunięcia spośród swoich rezerwacji 3. Użytkownik potwierdza intencję zwrotu [E1: Anulowanie] 4. System rejestruje prośbę 5. System potwierdza zwrot biletu 6. System zleca zwrot pieniędzy
Scenariusz wyjątku	E1: Anulowanie a. Użytkownik postanawia nie zwracać biletu

Nazwa	**Wyświetlanie historii zakupionych biletów**
ID	UC7
Aktor główny	Użytkownik prywatny
Aktorzy	Użytkownik prywatny
Priorytet	Główne
Opis	Użytkownik może przeglądać historię zakupionych biletów
Cel	Pozwolenie użytkownikowi na przejrzenie historii zakupionych przez niego biletów
Wyzwalanie	Użytkownik postanawia przejrzeć historię zakupionych biletów
War. początkowe	• Użytkownik posiada konto (UC4) • Użytkownik jest zalogowany (UC17) • Użytkownik zakupił bilet na wydarzenie (UC5)
War. końcowe	• Użytkownik przejrzał historię zakupionych biletów
W. funkcjonalne	• System zachowuje historię zakupu biletów dla użytkowników • Użytkownicy mogą zapoznać się ze swoją historią zakupów
W. niefunkcjonalne	-
Scenariusz	1. Użytkownik otwiera widok historii zakupionych biletów 2. System wyświetla listę zakupionych biletów posortowaną w kolejności zakupu, wyświetlającą podstawowe dane dot. biletów [O1: Wyświetlenie szczegółów]
Scenariusz opcjonalny	O1: Wyświetlenie szczegółów a. Użytkownik wybiera bilet na interesujące go wydarzenie b. System wyświetla użytkownikowi szczegółowe informacje o wybranym wydarzeniu

2.2.3 Organizator

Przypisane przypadki użycia:

- UC8: Stworzenie wniosku o utworzenie profilu
- UC9: Stworzenie wydarzenia
- UC10: Edycja danych wydarzenia
- UC11: Anulowanie wydarzenia
- UC12: Przeglądanie statystyk wydarzenia
- UC13: Stworzenie wniosku o usunięcie profilu
- UC14: Dodanie lokalizacji

Nazwa	**Stworzenie wniosku o utworzenie profilu**
ID	UC8
Aktor główny	Organizator
Aktorzy	Organizator
Priorytet	Główne
Opis	Podmiot składa wniosek o utworzenie konta z uprawnieniami organizatora wydarzeń
Cel	Złożenie wniosku o założenie konta
Wyzwalanie	Organizator postanawia złożyć wniosek o konto w systemie
War. początkowe	• Organizator nie posiada konta w systemie
War. końcowe	• Powstanie wniosku
W. funkcjonalne	• System pozwala na utworzenie nowego wniosku
W. niefunkcjonalne	• Interfejs tworzenia wniosku musi być przejrzysty
Scenariusz	1. Organizator wchodzi w widok tworzenia wniosku 2. System prezentuje formularz tworzenia wniosku 3. Organizator wypełnia formularz 4. Organizator przesyła formularz 5. System dokonuje walidacji danych [E1: Dane nie przechodzą walidacji] 6. System zapisuje wniosek
Scenariusz wyjątku	E1: Dane nie przechodzą walidacji a. Organizator podaje dane, które nie przechodzą walidacji lub są niepełne. b. System informuje organizatora o błędzie.

Nazwa	Stworzenie wydarzenia
ID	UC9
Aktor główny	Organizator
Aktorzy	Organizator
Priorytet	Główne
Opis	Organizator tworzy wydarzenie dla swojej organizacji
Cel	Stworzenie nowego wydarzenia
Wyzwalanie	Organizator wchodzi na stronę aplikacji w celu stworzenia wydarzenia
War. początkowe	• Organizator posiada konto w systemie (UC15) • Organizator jest zalogowany (UC17)
War. końcowe	• Powstanie nowego wydarzenia w systemie
W. funkcjonalne	• System pozwala na utworzenie nowego wydarzenia
W. niefunkcjonalne	• Interfejs tworzenia wydarzenia musi być przejrzysty • Wydarzenie powinno być tworzone z jak najmniejszym opóźnieniem
Scenariusz	1. Organizator wybiera opcję utworzenia wydarzenia 2. System prezentuje kreator wydarzenia 3. Organizator podaje informacje o wydarzeniu 4. System dokonuje walidacji danych [E1: Dane nie przechodzą walidacji] 5. System rejestruje wydarzenie
Scenariusz wyjątku	E1: Dane nie przechodzą walidacji a. Organizator podaje dane, które nie przechodzą walidacji lub są niepełne b. System informuje organizatora o błędzie

Nazwa	**Edycja danych wydarzenia**
ID	UC10
Aktor główny	Organizator
Aktorzy	Organizator
Priorytet	Drugorzędne
Opis	Organizator może edytować dane dot. wydarzenia
Cel	Umożliwienie organizatorowi zmianę szczegółów eventu
Wyzwalanie	Pojawiły się zmiany dot. organizacji eventu, które muszą zostać wprowadzone w systemie
War. początkowe	• Organizator posiada konto (UC15) • Organizator jest zalogowany (UC17) • Organizator ma utworzone co najmniej jedno aktywne wydarzenie (UC9) • Wydarzenie się jeszcze nie odbyło • Wydarzenie nie zostało anulowane
War. końcowe	• Organizator pomyślnie dokonał edycji danych wydarzenia
W. funkcjonalne	• System udostępnia widok edycji wydarzenia • Wydarzenia mogą być w pewnym stopniu edytowalne
W. niefunkcjonalne	• Prezentowany widok edycji powinien być intuicyjny i łatwo dostępny
Scenariusz	1. Organizator otwiera widok listy swoich wydarzeń 2. System przygotowuje widok i prezentuje go organizatorowi 3. Organizator wybiera wydarzenie, które chce edytować 4. System wyświetla organizatorowi widok edycji 5. Organizator edytuje wybrane informacje 6. System dokonuje walidacji danych [E1: Dane nie przechodzą walidacji] 7. System aktualizuje wydarzenie 8. System powiadamia organizatora o pomyślnej edycji 9. System powiadamia o zaistniałych zmianach klientów posiadających bilety na wydarzenie
Scenariusz wyjątku	E1: Dane nie przechodzą walidacji a. Organizator podaje dane, które nie przechodzą walidacji lub są niepełne b. System informuje organizatora o błędzie

Nazwa	**Anulowanie wydarzenia**
ID	UC11
Aktor główny	Organizator
Aktorzy	Organizator, Użytkownik prywatny
Priorytet	Drugorzędne
Opis	Organizator może anulować wydarzenie
Cel	Umożliwienie organizatorowi anulowanie eventu
Wyzwalanie	Wydarzenie zostało odwołane
War. początkowe	• Organizator posiada konto (UC15) • Organizator jest zalogowany (UC17) • Organizator ma utworzone co najmniej jedno aktywne wydarzenie (UC9) • Wydarzenie się jeszcze nie odbyło • Wydarzenie nie zostało anulowane
War. końcowe	• Organizator pomyślnie anulował wydarzenie
W. funkcjonalne	• Wydarzenia mogą być anulowane • System dokonuje zwrotu środków dla posiadaczy biletów
W. niefunkcjonalne	• Zwroty powinny być pełne • Zwroty pieniędzy powinny być bezpieczne
Scenariusz	1. Organizator otwiera widok listy swoich wydarzeń 2. System przygotowuje widok i prezentuje go organizatorowi 3. Organizator wybiera wydarzenie, które chce anulować 4. Użytkownik potwierdza intencję anulowania wydarzenia [E1: Anulowanie] 5. System anuluje wydarzenie 6. System powiadamia organizatora o pomyślnym anulowaniu eventu 7. System powiadamia klientów posiadających bilety na wydarzenie o zaistniałych zmianach 8. System zleca zwrot środków
Scenariusz wyjątku	E1: Anulowanie a. Użytkownik postanawia nie anulować wydarzenia

Nazwa	**Przeglądanie statystyk wydarzenia**
ID	UC12
Aktor główny	Organizator
Aktorzy	Organizator
Priorytet	Drugorzędne
Opis	Organizator widzi dla swoich wydarzeń liczbę sprzedanych biletów
Cel	Umożliwienie organizatorowi obserwacji popularności swoich wydarzeń
Wyzwalanie	Organizator postanawia przejrzeć statystyki swoich wydarzeń
War. początkowe	• Organizator posiada konto (UC15) • Organizator jest zalogowany (UC17) • Organizator ma utworzone co najmniej jedno aktywne wydarzenie (UC9)
War. końcowe	• Organizator widzi statystyki sprzedaży biletów na swoje wydarzenia
W. funkcjonalne	• System udostępnia informacje o sprzedanych biletach
W. niefunkcjonalne	• Informacje o biletach i wydarzeniach powinny być prezentowane w sposób przejrzysty
Scenariusz	1. Organizator otwiera widok listy swoich wydarzeń 2. System wyświetla widok listy wydarzeń zawierający informacje o liczbie sprzedanych biletów dla każdego wydarzenia [O1: Sprawdzenie danych szczegółowych]
Scenariusz opcjonalny	O1: Sprawdzenie danych szczegółowych a. Organizator wybiera wydarzenie, o którego sprzedaży biletów chce dowiedzieć się więcej b. System prezentuje organizatorowi widok wybranego wydarzenia rozszerzony o szczegółowe statystyki sprzedaży każdej z kategorii biletów

Nazwa	**Stworzenie wniosku o usunięcie profilu**
ID	UC13
Aktor główny	Organizator
Aktorzy	Organizator
Priorytet	Główne
Opis	Podmiot składa wniosek o usunięcie konta z uprawnieniami organizatora wydarzeń
Cel	Złożenie wniosku o usunięcie konta
Wyzwalanie	Organizator postanawia złożyć wniosek usunięcie konta w systemie
War. początkowe	• Organizator posiada konto w systemie (UC15) • Organizator jest zalogowany (UC17)
War. końcowe	• Powstanie wniosku
W. funkcjonalne	• System pozwala na utworzenie nowego wniosku
W. niefunkcjonalne	• Proces składania wniosku musi być intuicyjny
Scenariusz	1. Organizator zgłasza wniosek o usunięcie aktywnego konta 2. System wyświetla zapytanie o potwierdzenie intencji złożenia wniosku 3. Organizator potwierdza intencję złożenia wniosku 4. System rejestruje wniosek 5. System informuje o złożeniu wniosku

Nazwa	Dodanie lokalizacji
ID	UC14
Aktor główny	Organizator
Aktorzy	Organizator
Priorytet	Główne
Opis	Podmiot dodaje do systemu informacje o lokalizacji przyszłych eventów
Cel	Dodanie lokalizacji do systemu oraz umożliwienie jej użycia przy tworzeniu przyszłych wydarzeń
Wyzwalanie	Organizator postanawia dodać lokalizację do systemu
War. początkowe	• Organizator posiada konto (UC15) • Organizator jest zalogowany (UC17)
War. końcowe	• Powstanie wniosku
W. funkcjonalne	• System pozwala na utworzenie nowego wniosku
W. niefunkcjonalne	• Interfejs tworzenia wniosku musi być przejrzysty
Scenariusz	1. Organizator wybiera opcję dodania lokalizacji [A1: Lokalizacja do wydarzenia ->] 2. System prezentuje formularz tworzenia lokalizacji [<- A1] 3. Organizator wypełnia formularz 4. Organizator przesyła formularz 5. System dokonuje walidacji danych [E1: Dane nie przechodzą walidacji] 6. System zapisuje lokalizację 7. System wyświetla potwierdzenie sukcesu operacji
Scenariusz alternatywny	A1: Lokalizacja do wydarzenia a. Organizator dodaje lokalizację w trakcie tworzenia wydarzenia b. System prezentuje widok tworzenia lokalizacji
Scenariusz wyjątku	E1: Dane nie przechodzą walidacji a. Organizator podaje dane, które nie przechodzą walidacji lub są niepełne. b. System informuje organizatora o błędzie.

2.2.4 Administrator

Przypisane przypadki użycia:

- UC15: Rejestrowanie organizatora
- UC16: Usuwanie organizatora

Nazwa	**Rejestrowanie organizatora**
ID	UC15
Aktor główny	Administrator
Aktorzy	Administrator, organizator
Priorytet	Główne
Opis	Administrator tworzy konto dla organizatora
Cel	Umożliwienie zewnętrznym firmom wykorzystanie systemu do promowania swoich wydarzeń i sprzedaży biletów
Wyzwalanie	Organizator składa wniosek o utworzenie profilu
War. początkowe	• Administrator jest zalogowany (UC17) • Istnieje co najmniej jeden nierozwiązany wniosek
War. końcowe	• Utworzony zostaje profil firmy wewnątrz systemu
W. funkcjonalne	• System umożliwia administratorowi tworzenie nowych kont organizatorów
W. niefunkcjonalne	• Zgłoszenia o rejestrację powinny być odporne na nadużycia • Kontakt z administratorem powinien być szybki
Scenariusz	1. Administrator otwiera widok listy wniosków 2. Administrator weryfikuje wniosek [E1: Odrzucenie wniosku] [O1: Dodatkowa komunikacja] 3. Administrator inicjuje tworzenie konta 4. System prezentuje formularz tworzenia konta uzupełniony danymi podanymi przez wnioskodawcę 5. Administrator przesyła formularz 6. System waliduje zgłoszenie [E2: Dane nie przechodzą walidacj] 7. System tworzy konto
Scenariusz opcjonalny	O1: Dodatkowa komunikacja a. Administrator kontaktuje się z organizatorem za pośrednictwem zewnętrznych narzędzi w celu uszczegółowienia warunków współpracy.
Scenariusz wyjątku	E1: Odrzucenie wniosku a. Administrator odrzuca wniosek. b. System wysyła do wnioskodawcy wiadomość informującą o odrzuceniu zgłoszenia. E2: Dane nie przechodzą walidacji a. Administrator podaje dane, które nie przechodzą walidacji lub są niepełne. b. System informuje administratora o błędzie. c. Administrator informuje organizatora o wadliwych danych.

Nazwa	**Usuwanie organizatora**
ID	UC16
Aktor główny	Administrator
Aktorzy	Administrator, organizator
Priorytet	Drugorzędny
Opis	Administrator usuwa konto organizatora, na jego prośbę lub z innych przyczyn
Cel	Umożliwienie usunięcia konta organizatora administratorowi
Wyzwalanie	• Organizator wysyła prośbę do administratora o usunięcie profilu • Administrator podejmuje decyzję o usunięciu profilu z innych przyczyn
War. początkowe	• Administrator jest zalogowany (UC17) • Istnieje konto organizatora w systemie (UC15)
War. końcowe	• Profil organizatora zostaje usunięty z systemu • Wszystkie wydarzenia powiązane z organizatorem zostają usunięte
W. funkcjonalne	• System umożliwia administratorowi usuwanie kont organizatorów • System umożliwia automatyczne usuwanie wydarzeń powiązanych z organizatorem
W. niefunkcjonalne	• System powinien być spójny przed i po usunięciu wydarzenia • System powinien być bezpieczny, aby uniknąć niepożądanego usunięcia profilu
Scenariusz	1. Administrator otwiera widok listy kont użytkowników 2. Administrator wybiera opcję usunięcia konta organizatora 3. System usuwa konto [O1: Usunięcie wydarzeń]
Scenariusz opcjonalny	O1: Usunięcie wydarzeń a. System usuwa również wszystkie wydarzenia powiązane z organizatorem, jeśli istnieją. b. System inicjuje zwrot środków dla posiadaczy biletów

2.2.5 Uniwersalne

Przypisane przypadki użycia:

• UC17: Zalogowanie się na konto

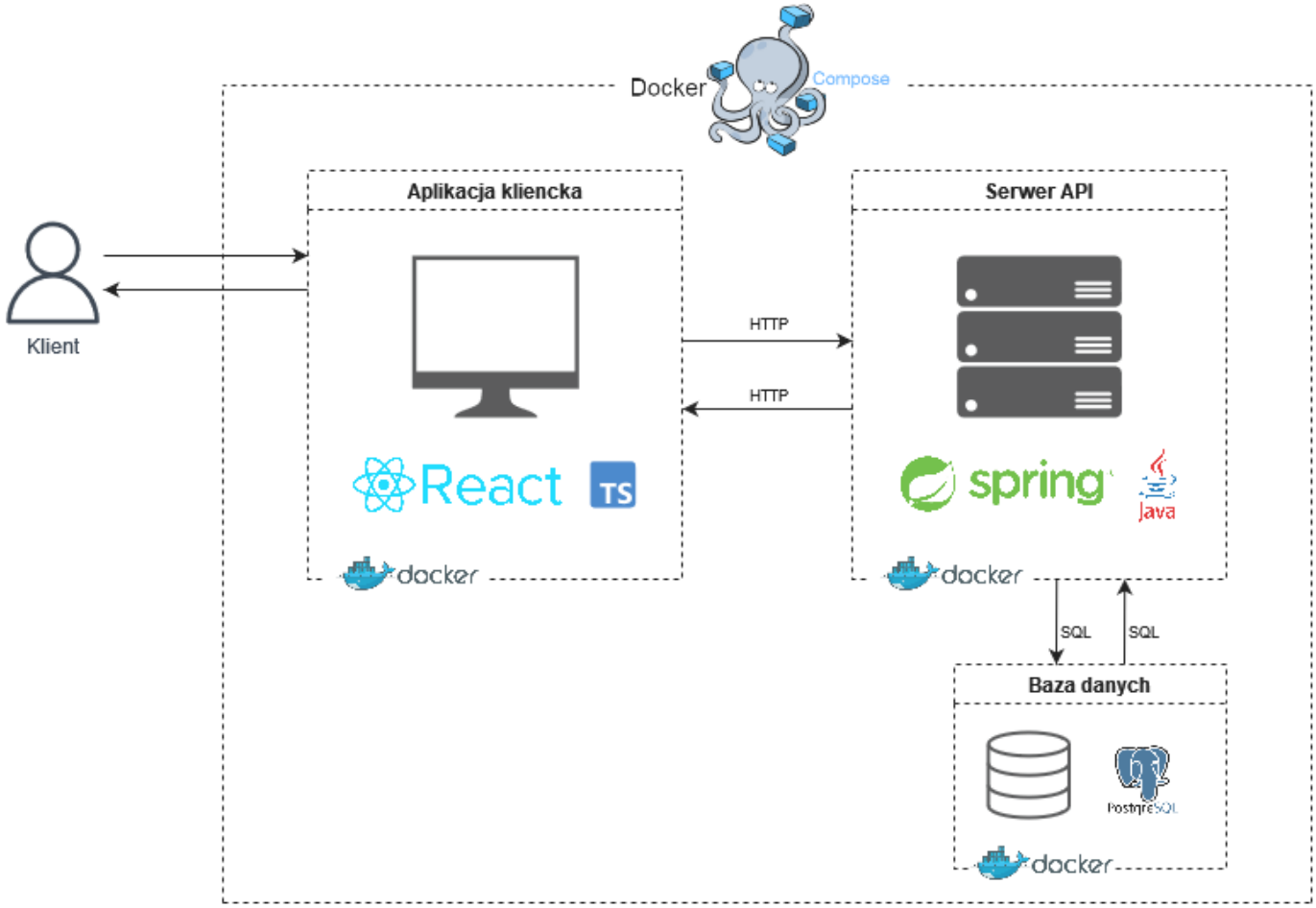
Nazwa	**Zalogowanie się na konto**
ID	UC17
Aktor główny	Dowolny użytkownik systemu
Aktorzy	Dowolny użytkownik systemu
Priorytet	Główne
Opis	Użytkownik loguje się na swoje konto
Cel	Umożliwienie użytkownikowi zalogowania się na swoje konto, z którym powiązane są pewnego rodzaju uprawnienia
Wyzwalanie	Użytkownik chce się zalogować
War. początkowe	• Użytkownik posiada konto (UC4) • Użytkownik nie jest zalogowany
War. końcowe	• Użytkownik jest zalogowany
W. funkcjonalne	• System umożliwia zalogowanie się na wcześniej utworzone konto
W. niefunkcjonalne	• Formularz logowania powinien być intuicyjny
Scenariusz	1. Użytkownik wchodzi na dowolną stronę aplikacji 2. Użytkownik wybiera opcję logowania 3. System prezentuje formularz logowania 4. Użytkownik wypełnia formularz 5. Użytkownik przesyła formularz 6. System dokonuje walidacji danych [E1: Dane nie przechodzą walidacji] 7. System wyświetla potwierdzenie logowania 8. System przekierowuje użytkownika do strony głównej
Scenariusz wyjątku	E1: Dane nie przechodzą walidacji a. Użytkownik podaje dane, które nie przechodzą walidacji lub są niepełne. b. System wyświetla użytkownikowi informację o błędzie.

2.3 Słownik pojęć

- **Wydarzenie, event** - zorganizowane zdarzenie, na które sprzedawane są bilety w ramach systemu
- **Profil** - konto użytkownika w systemie
- **Organizacja, firma** - podmiot zewnętrzny wykorzystujący system w celu promocji oraz sprzedaży biletów na swoje wydarzenia
- **Bilet** - dokument uprawniający posiadacza do wzięcia udziału w wydarzeniu

3. Projekt systemu

3.1 Ogólny model architektury



Architektura systemu oparta jest na prosty modelu trójwarstwowym, składającym się z aplikacji klienckiej, serwera API oraz bazy danych, co zapewnia prosty podział obowiązków każdego z komponentów i modularność systemu.

Aplikacja kliencka odpowiada za prezentację danych systemu oraz zapewnia możliwość interakcji klientów z systemem. Jej głównym elementem jest graficzny interfejs wyświetlany użytkownikom końcowym korzystającym z platformy przy pomocy przeglądarki internetowej.

Serwer API odpowiada za realizację całej logiki biznesowej systemu. Odpowiada na żądania użytkowników korzystających z aplikacji klienckiej oraz udostępnia punkty końcowe pozwalające na pobranie lub modyfikację danych gromadzonych w systemie. Serwer API jest zaimplementowany w stylu REST i może współpracować z innymi systemami niż aplikacja kliencka zrealizowana w tym projekcie.

Baza danych przechowuje oraz organizuje wszystkie dane systemowe. Komunikuje się jedynie z serwerem API.

3.2 Technologie

W aplikacji klienckiej wykorzystany został framework React w połączeniu z językiem programowania wysokiego poziomu TypeScript. Aplikacja stworzona została przy użyciu generatora Create React App zapewniającego środowisko deweloperskie oraz wykorzystującego bundler Webpack i transpilator Babel.

Do stworzenia serwera API użyto języka Java oraz frameworku Spring Boot. Podmoduły Spring takie jak: web, security, data-jpa i jdbc zapewniają odpowiednio poprawne działanie podstaw aplikacji webowej, bezpieczeństwo oraz mapowanie klas w Javie na rekordy w bazie danych.

Jako bazę danych wykorzystano PostgreSQL w wersji 16.

Za konteneryzację poszczególnych komponentów odpowiedzialne jest narzędzie Docker. Tworzone kontenery zarządzane są przy pomocy narzędzia Docker Compose. Całą aplikację można uruchomić z użyciem polecenia “docker compose up” na dowolnej maszynie wspierającej Dockera.