

Politechnika Warszawska

Inżynieria Oprogramowania 1

PLATFORMA DO ZNAJDOWANIA WYDARZEŃ W OKOLICY WERSJA 1.0

Wojciech Gajda 304494

Mieszko Mirgos 313360

Franciszek Majkusiak 313338

Wojciech Kutak 313305

Prowadzący: inż. Maciej Bednarz

Data oddania: **1 marca 2023**

Spis treści

1	Wstęp	5
1.1	Słownik pojęć	5
1.2	Cel projektu	6
1.3	O projekcie	6
2	Architektura systemu	7
2.1	Serwer centralny	7
2.2	Aplikacja webowa dla organizatorów	7
2.3	Aplikacja mobilna dla użytkowników	7
2.4	Symulacja aplikacji mobilnej	8
2.4.1	Strona główna	8
2.4.2	Menu	9
2.4.3	Panel wydarzeń	10
2.4.4	Panel rezerwacji	11
2.4.5	Panel danych szczegółowych rezerwacji	12
3	Funkcjonalności systemu	13
3.1	User stories	13
3.2	Use case diagram	14
4	Struktura systemu	15
4.1	Class diagram	15
4.1.1	Klasa Event	15
4.1.2	Klasa Organizer	16
4.1.3	Klasa User	16
4.1.4	Klasa Extended User	16
4.1.5	Klasa Reservation	16
4.1.6	Klasa Category	16
4.1.7	Klasa Schema	16
4.2	Stany obiektów	17
4.2.1	User	17
4.2.2	Event	17
4.2.3	Reservation	18
4.2.4	Organizer	18
5	Aktywności w systemie	19
5.1	Przeglądanie eventów	19
5.2	Tworzenie eventu	19
5.3	Edycja eventu	20
5.4	Anulowanie eventu	20
5.5	Tworzenie kategorii	21

5.6	Przeglądanie kategorii	21
5.7	Logowanie do konta organizatora	22
5.8	Tworzenie konta organizatora	22
5.9	Potwierdzenie konta organizatora	23
5.10	Edycja konta organizatora	23
5.11	Usuwanie konta organizatora	24
5.12	Tworzenie rezerwacji	24
5.13	Usunięcie rezerwacji	25
6	Diagramy sekwencji	26
6.1	Przeglądanie eventów	26
6.2	Tworzenie eventu	27
6.3	Edycja eventu	28
6.4	Anulowanie eventu	29
6.5	Tworzenie kategorii	30
6.6	Przeglądanie kategorii	31
6.7	Rejestracja	32
6.8	Potwierdzenie konta	33
6.9	Logowanie	34
6.10	Edycja konta	35
6.11	Usunięcie konta	36
6.12	Tworzenie rezerwacji	37
6.13	Usunięcie rezerwacji	38
7	Komunikacja w systemie	39
7.1	Obiekty komunikacji	39
7.1.1	Event	39
7.1.2	Category	40
7.1.3	EventStatus	40
7.1.4	ReservationDTO	41
7.1.5	Organizer	41
7.2	Endpointy	43
7.2.1	Dot. konta organizatora	43
7.2.2	Dot. Eventu	45
7.2.3	Dot. kategorii	48
7.2.4	Dot. Rezerwacji	49
7.3	API serwera w standardzie OpenAPI 3.0	50

Spis rysunków

1	Strona główna aplikacji mobilnej	8
2	Menu aplikacji	9

3	Panel wydarzeń	10
4	Panel rezerwacji	11
5	Panel danych szczegółowych rezerwacji	12
6	User use cases	14
7	Organiser use cases	14
8	Class diagram	15
9	Diagram stanów - User	17
10	Diagram stanów - Event	17
11	Diagram stanów - Reservation	18
12	Diagram stanów - Organizer	18
13	Diagram aktywności - Przeglądane eventów	19
14	Diagram aktywności - Tworzenie eventu	19
15	Diagram aktywności - Edycja eventu	20
16	Diagram aktywności - Usuwanie eventu	20
17	Diagram aktywności - Tworzenie kategorii	21
18	Diagram aktywności - Przeglądanie kategorii	21
19	Diagram aktywności - Login	22
20	Diagram aktywności - Tworzenie konta organizatora	22
21	Diagram aktywności - potwierdzenie konta organizatora	23
22	Diagram aktywności - Edycja konta organizatora	23
23	Diagram aktywności - Edycja eventu	24
24	Diagram aktywności - Rezerwacja	24
25	Diagram aktywności - Edycja eventu	25
26	Diagram sekwencji - Przeglądanie eventów	26
27	Diagram sekwencji - Tworzenie eventu	27
28	Diagram sekwencji - Edycja eventu	28
29	Diagram sekwencji - Anulowanie eventu	29
30	Diagram aktywności - Tworzenie kategorii	30
31	Diagram sekwencji - Przeglądanie kategorii	31
32	Diagram aktywności - Rejestracja	32
33	Diagram sekwencji - Potwierdzenie konta	33
34	Diagram sekwencji - Logowanie	34
35	Diagram sekwencji - Edycja konta	35
36	Diagram sekwencji - Usunięcie konta	36
37	Diagram aktywności - Rezerwacja	37
38	Diagram sekwencji - Usunięcie rezerwacji	38
39	Pola obiektu Event z określeniem typów pól	39
40	Pola obiektu Category z określeniem typów pól	40
41	Pola obiektu EventStatus z określeniem typów pól	40
42	Pola obiektu ReservationDTO z określeniem typów pól	41
43	Pola obiektu Organizer z określeniem typów pól	42

1 Wstęp

Niniejszy dokument stanowi specyfikację systemu rezerwacji miejsc. System integruje organizatorów wydarzeń z ich uczestnikami ułatwiając komunikację między nimi. System pozwala organizatorom na ogłaszanie eventów, natomiast uczestnikom na dokonywanie rezerwacji na nie ze szczegółowym wyborem miejsca ze schematu. Dla uczestnika wydarzeń przygotowany zostanie frontend zaczerpnięty z serwisów ogłoszeniowych, a dla organizatorów przygotowany zostanie intuicyjny panel.

1.1 Słownik pojęć

- Użytkownik - osoba bezpośrednio używająca systemu
- Uczestnik wydarzenia (User) - użytkownik aplikacji mobilnej, przegląda eventy, rezerwuje miejsca na eventach i zarządza swoimi rezerwacjami.
- Organizator (Organizer) - użytkownik aplikacji webowej, tworzy eventy i nimi administruje, konto organizatora wymaga uwierzytelnienia.
- Rezerwacja (Reservation) - relacja łącząca użytkownika z eventem, może zawierać miejsce użytkownika na schemacie i dodatkowe informacje.
- Wydarzenie (Event) - Wydarzenie, tworzone przez organizatora, na które może zapisać się użytkownik. Może opcjonalnie posiadać schemat miejsc.
- Schemat (Schema) - Definiuje rozkład miejsc w wydarzeniu, nie jest wymagany do istnienia wydarzenia.
- Kategoria (Category)- Obiekt reprezentujący kategorię, który może być połączony z wydarzeniem, pozwala na łatwiejsze filtrowanie.

1.2 Cel projektu

Celem projektu jest opracowanie systemu ułatwiającego komunikację pomiędzy organizatorami wydarzeń, a ich uczestnikami. Na rynku polskim brakuje platform pokroju działającego w USA findEvent.com. Nie istnieje zatem jednolita platforma, na której można by odnaleźć wydarzenia w najbliższej okolicy, jak również dokonać na nie rezerwacji.

Obserwując sukcesy platform pozwalających na zamieszczanie ogłoszeń, takich jak OLX.pl i podobne można wnioskować, że za sukces komunikacji sprzedając - kupujący odpowiada swoboda, jaką daje platforma. Sprzedający nie musi spełniać wygórowanych wymagań, ani opłacać abonamentu aby zamieścić ogłoszenie. Analogicznie klient nie musi podać wstępnie żadnych danych, aby przeglądać ogłoszenia. W naszym systemie to organizator wydarzenia jest sprzedawcą, a uczestnik kupującym. Podobnie jak OLX.pl, pierwsza wersja systemu, mająca na celu stworzenie bazy użytkowników, nie zawiera transakcji.

1.3 O projekcie

Platforma integruje organizatorów wydarzeń i uczestników wydarzeń.

Dla organizatorów udostępniony zostanie webowy panel organizatora. Za pośrednictwem panelu organizator wprowadza informacje o wydarzeniach wraz z datą i lokalizacją wydarzenia. Ponadto jeżeli wydarzenie odbywa się w sali o znanym rozkładzie miejsc organizator może zaplanować schemat pomieszczenia i określić liczbę dostępnych miejsc.

Dla użytkowników udostępniona zostanie aplikacja mobilna. W aplikacji użytkownikowi wyświetlane są wydarzenia dodane przez organizatorów. Użytkownik może przeglądać, filtrować i sortować wydarzenia (np. do pewnej odległości). Po wyborze wydarzenia użytkownik może otrzymać więcej informacji o wydarzeniu i dokonać rezerwacji (z puli dostępnych miejsc lub wybrać ze schematu jeśli został określony przez organizatora). Rezerwacja wiąże uczestnika z wydarzeniem i zawiera dodatkowe informacje ją identyfikujące.

2 Architektura systemu

Na system realizujący opisane w kolejnych rozdziałach funkcjonalności składają trzy niezależnie działające komponenty komunikujące się pomiędzy sobą. Jest to:

- Serwer centralny
- Webowa aplikacja dla organizatorów
- Aplikacja mobilna dla użytkowników

2.1 Serwer centralny

System centralny stanowi back-end całej platformy. Połączony z bazą danych serwis przechowuje informacje o eventach, uczestniach, organizatorach, rezerwacjach i innych klasach składowych. System centralny dokonuje pełnej kontroli nad działaniem platformy w celu zapewnienia jej stabilności i uniknięcia kolizji. Dodatkowo system wystawia odpowiednie REST API, jedno dla aplikacji webowej i drugie dla aplikacji mobilnej. Metody API pozwalają na pobieranie informacji z serwera, jak również ich modyfikację o ile jest to zgodne z rolą danego klienta.

2.2 Aplikacja webowa dla organizatorów

Aplikacja webowa to strona internetowa wystawiona na znanym publicznym adresie IP. Strona ma charakter panelu z możliwością logowania. Z poziomu panelu organizator - główny użytkownik panelu - może utworzyć nowe wydarzenie i wprowadzić informację o nim. Ponadto panel umożliwia wprowadzenie opcjonalnego schematu miejsc oraz podgląd aktualnych rezerwacji. W przypadku braku połączenia z serwerem nie zostanie utworzony event. W przypadku błędu serwera w trakcie tworzenia eventu, event nie utworzy się, a organizator zostanie poinformowany o nieutworzeniu eventu.

2.3 Aplikacja mobilna dla użytkowników

Aplikacja mobilna to aplikacja na urządzenia z systemem Android lub iOS. W aplikacji możliwe jest przeglądanie listy wydarzeń zamieszczonych na platformie wraz z dodatkowymi funkcjami filtracji i sortowania. Do przeglądania wydarzeń nie ma potrzeby rejestracji, ani podawania żadnych danych.

Użytkownik aplikacji może wybrać wydarzenie z listy i dokonać na nie rezerwacji. W takiej sytuacji urządzenie z którego dokonała się rezerwacja stanie się nośnikiem rezerwacji, przechowując informację o niej. W przypadku utraty połączenia z serwerem centralnym użytkownik może wyświetlać dane dot. swoich rezerwacji, aczkolwiek nie może przeglądać nowych eventów ani zapisywać się na nie. W przypadku błędu połączenia z serwerem po próbie zapisania się na wydarzenie z listy, rezerwacja nie zostanie utworzona, a użytkownik zostanie powiadomiony.

2.4 Symulacja aplikacji mobilnej

2.4.1 Strona główna

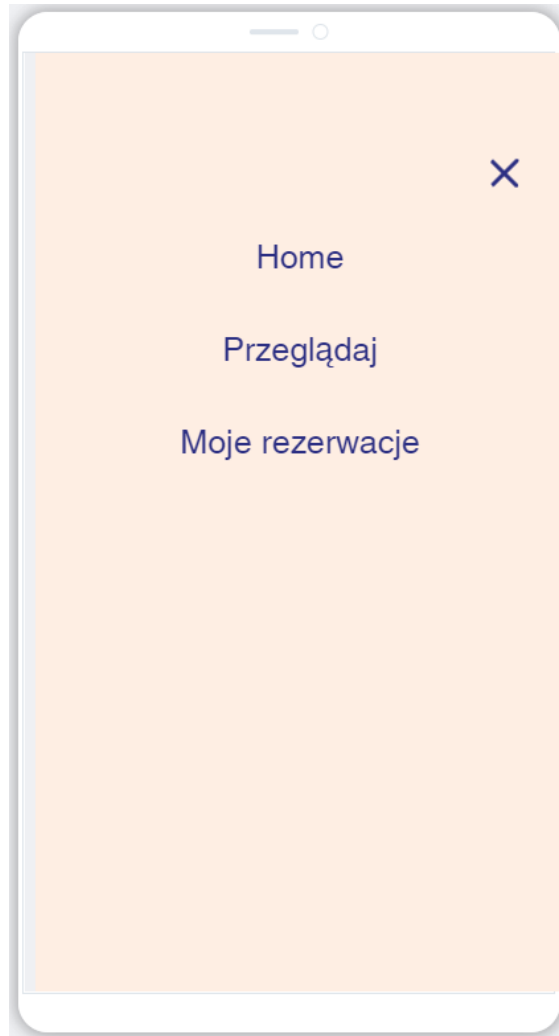
Strona główna przedstawia krótki opis strony.



Rysunek 1: Strona główna aplikacji mobilnej

2.4.2 Menu

Menu aplikacji umożliwia użytkownikowi przełączanie się pomiędzy panelami. Do wyboru jest panel główny ("Strona główna"), panel wydarzeń ("Przeglądaj") i panel rezerwacji ("Moje rezerwacje").



Rysunek 2: Menu aplikacji

2.4.3 Panel wydarzeń

W panelu wydarzeń znajdują się wszystkie dostępne wydarzenia. Użytkownik może zapisać się na wydarzenie poprzez kliknięcie przycisku "Rezerwuj". Jeśli rezerwacja się powiedzie, w panelu "Moje rezerwacje" pojawi się rezerwacja z dostępnym podglądem kodu QR rezerwacji. W przypadku braku internetu panel jest niedostępny.



Rysunek 3: Panel wydarzeń

2.4.4 Panel rezerwacji

Panel umożliwia użytkownikowi przeglądanie aktualnych rezerwacji. Po kliknięciu przycisku "QR" na rezerwacji otworzy się panel rezerwacji z dokładnymi danymi biletu oraz kodem QR biletu.



Rysunek 4: Panel rezerwacji

2.4.5 Panel danych szczegółowych rezerwacji

Panel prezentuje użytkownikowi kod QR rezerwacji oraz szczegółowe dane rezerwacji.



Rysunek 5: Panel danych szczegółowych rezerwacji

3 Funkcjonalności systemu

3.1 User stories

Jako użytkownik mogę ...

- Jako użytkownik mogę otworzyć aplikację i wyświetlić listę wydarzeń, żebym mógł korzystać z aplikacji
- Jako użytkownik mogę przeglądać, sortować i filtrować wydarzenia po nazwie, odległości i kategorii, aby ułatwić wybór
- Jako użytkownik mogę wybrać wydarzenie i dokonać rezerwacji, żeby mieć przypisane miejsce
- Jako użytkownik mogę stać się uczestnikiem wydarzenia po dokonaniu rezerwacji, żeby móc posługiwać się tą rezerwacją
- Jako użytkownik mogę w trakcie rezerwacji wybrać miejsce ze schematu, żeby móc zarezerwować konkretne miejsce

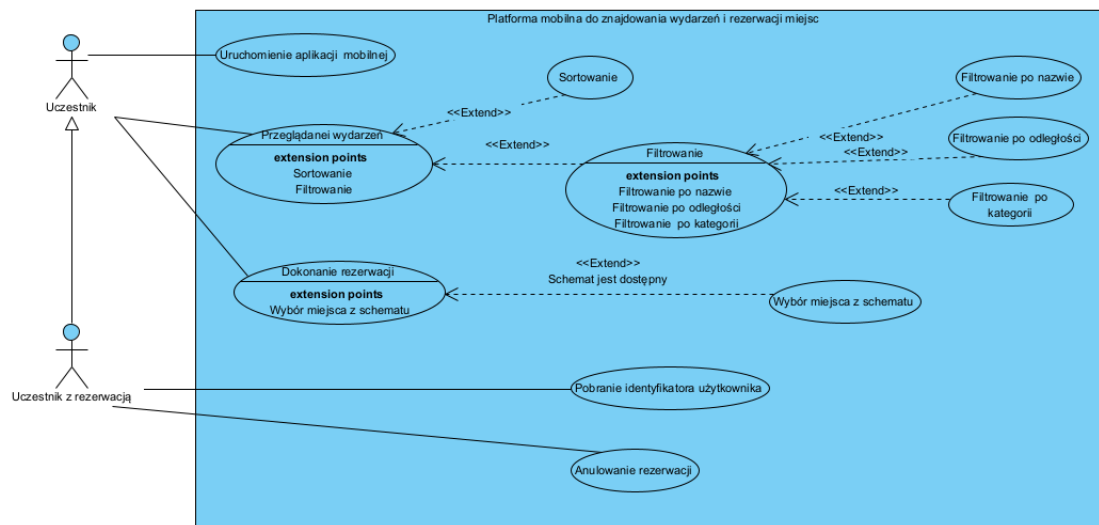
Jako uczestnik wydarzenia mogę ...

- Jako uczestnik wydarzenia mogę wyświetlić unikalny identyfikator rezerwacji
- Jako uczestnik wydarzenia mogę anulować rezerwację

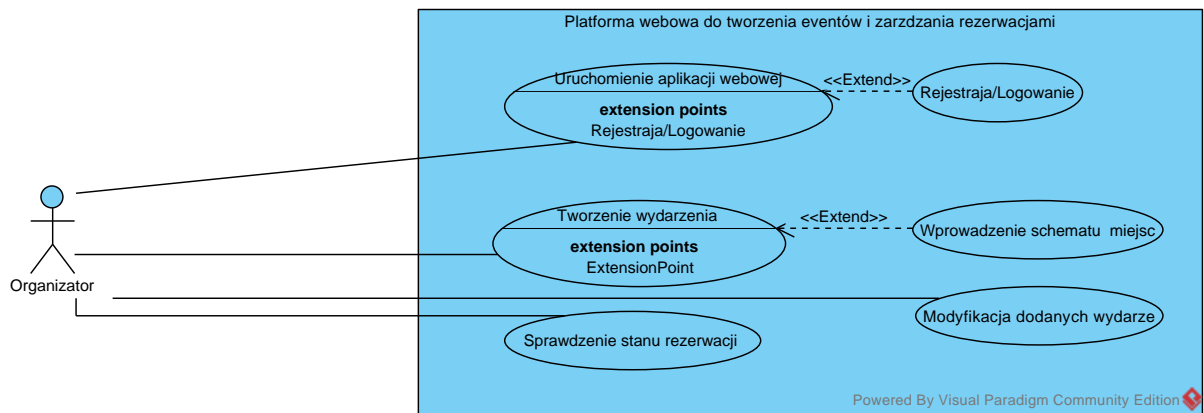
Jako organizator wydarzenia mogę ...

- Jako organizator wydarzenia mogę otworzyć panel organizatora, żebym mógł korzystać z systemu
- Jako organizator wydarzenia mogę utworzyć nowe wydarzenie, żeby było ono widoczne dla uczestników
- Jako organizator wydarzenia mogę dodać kategorię wydarzenia, żeby ułatwić użytkownikom wybór
- Jako organizator wydarzenia mogę wprowadzić rozkład miejsc dla danego wydarzenia, aby użytkownicy mogli zarezerwować konkretne miejsca
- Jako organizator wydarzenia mogę edytować i usuwać swoje wydarzenie
- Jako organizator wydarzenia mogę utworzyć konto organizatora i logować się do niego, żeby przeglądać utworzone przeze mnie wydarzenia
- Jako organizator wydarzenia mogę sprawdzić ile osób na chwilę obecną zapisało się na dany event, żeby przygotować się na ich przybycie

3.2 Use case diagram



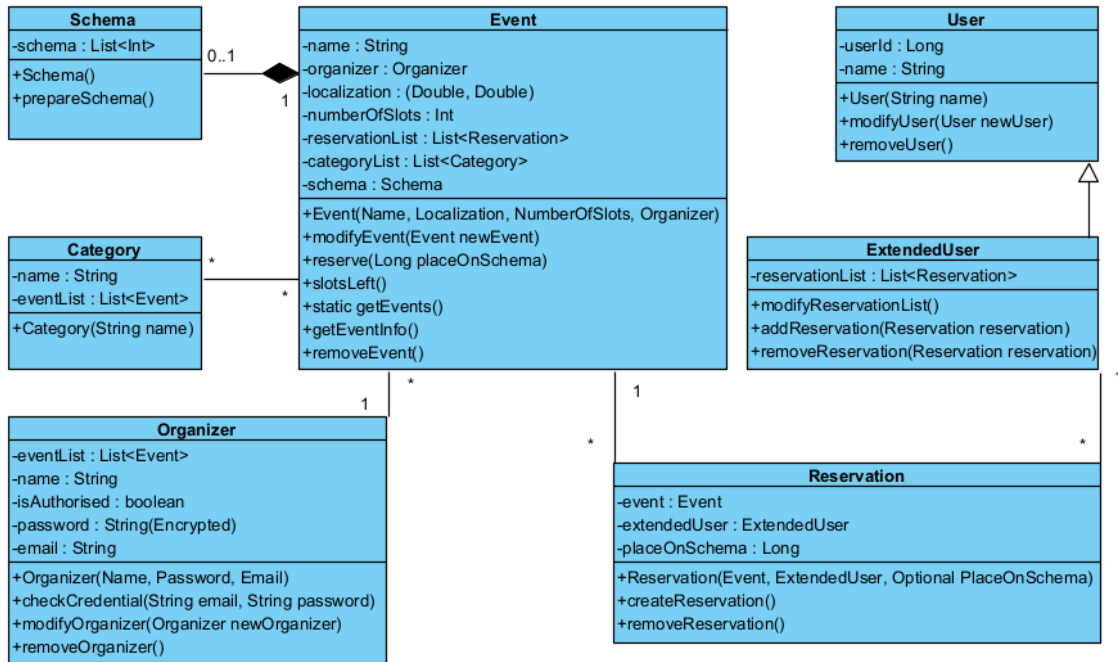
Rysunek 6: User use cases



Rysunek 7: Organiser use cases

4 Struktura systemu

4.1 Class diagram



Rysunek 8: Class diagram

4.1.1 Klasa Event

Klasa Event odwzorowuje wydarzenie. Przechowuje dane o wydarzeniu:

- Nazwa
- Organizator (one to one)
- Lokalizacja jako współrzędne geograficzne
- Liczbę dostępnych miejsc
- Listę rezerwacji (one to many)
- Kategoria wydarzenia (one to many)
- Schemat miejsc (one to one)

Klasa może zostać utworzona i zarządzana przez organizatora.

4.1.2 Klasa Organizer

Klasa Organizer odwzorowuje organizatora eventu. Przechowuje dane organizatora, umożliwia założenie konta, logowanie (checkCredentials) oraz modyfikacje konta organizatora. Tylko organizator może tworzyć nowy event, przekazując siebie jako jeden z parametrów konstruktora klasy

4.1.3 Klasa User

Klasa User przechowuje dane użytkownika. Umożliwia przeglądanie eventów, modyfikacje konta i oraz utworzenie rezerwacji.

4.1.4 Klasa Extended User

Klasa Extended User metody Usera i rozszerza je o zarządzanie istniejącymi rezerwacjami użytkownika.

4.1.5 Klasa Reservation

Klasa Reservation reprezentuje rezerwację w ramach określonego wydarzenia, dokonywaną przez użytkownika. Przechowuje dane dotyczące

- wydarzenia Event (many to one)
- użytkownika ExtendedUser, którego dotyczy dana rezerwacja (many to one)
- dodatkowe dane (na chwilę obecną wyłącznie opcjonalne współrzędne na schemacie miejsc).

4.1.6 Klasa Category

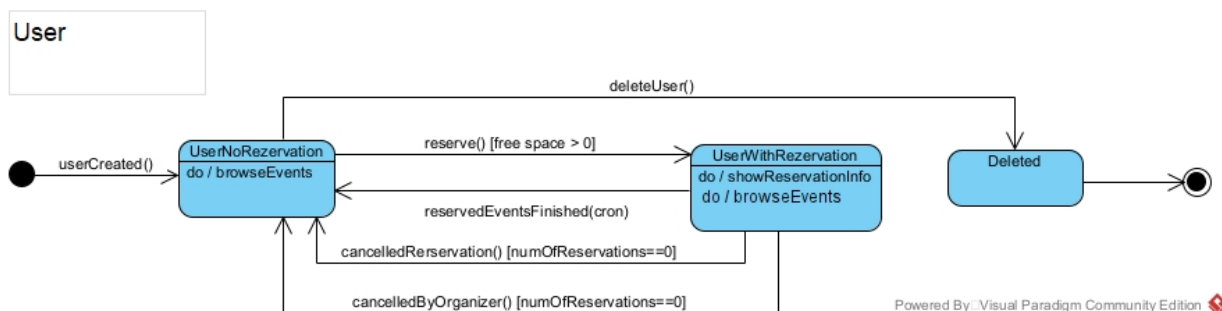
Klasa odwzoruje kategorie wydarzeń. Kategorie są globalne. Mogą zostać utworzone przez organizatora. Wydarzenie może należeć do 0, 1 lub więcej kategorii. Kategorie służą do filtrowania wydarzeń.

4.1.7 Klasa Schema

Klasa odwzorowuje schemat miejsca w którym odbywa się event. Klasa istnieje tylko w obrębie klasy Event.

4.2 Stany obiektów

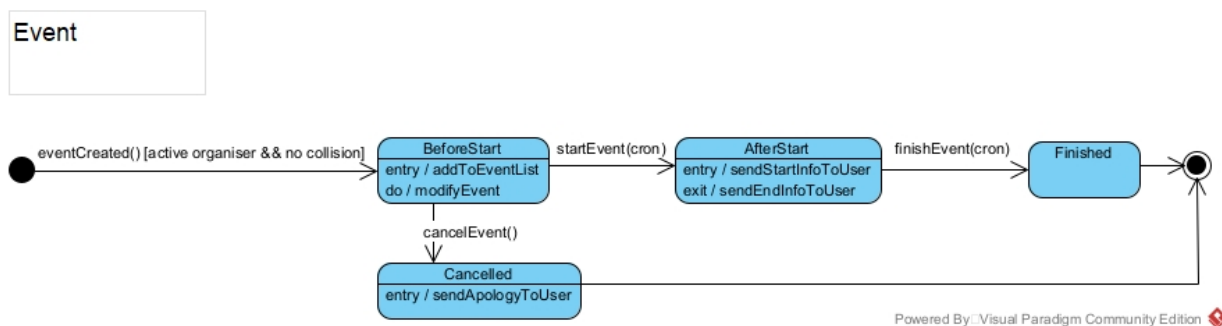
4.2.1 User



Rysunek 9: Diagram stanów - User

Użytkownik systemu ma dwa podstawowe stany: użytkownik bez żadnej rezerwacji oraz użytkownik po dokonaniu rezerwacji na któryś z eventów. Użytkownik bez rezerwacji może przeglądać dostępne eventy. Użytkownik z rezerwacją może wyświetlać informacje o swojej rezerwacji. Przejście między tymi stanami odbywa się w momencie dokonania przez użytkownika rezerwacji.

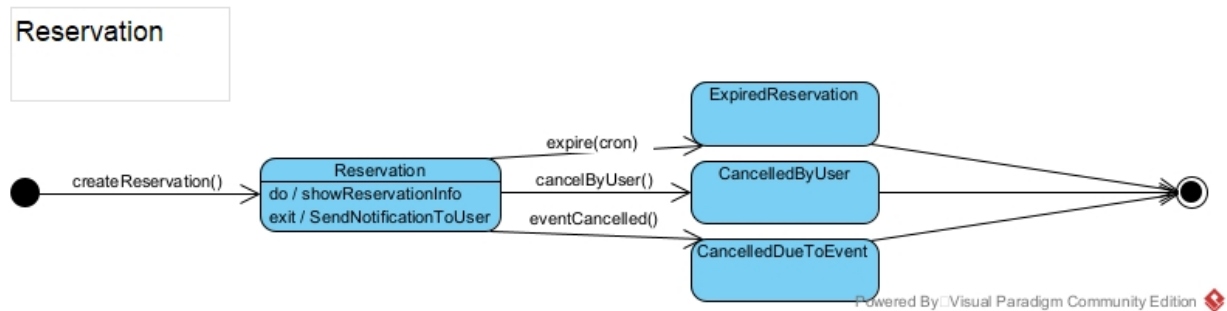
4.2.2 Event



Rysunek 10: Diagram stanów - Event

Event po utworzeniu przez organizatora znajduje się w stanie **BeforeStart**. W tym stanie może on być jeszcze modyfikowany. Z tego stanu również może on być odwołany co skutkuje wysłaniem powiadomienia do wszystkich zarejestrowanych na niego użytkowników. Po rozpoczęciu eventy przechodzi on do stanu **AfterStart**. Stan ten oznacza, że wydarzenie trwa. W chwili rozpoczęcia wydarzenia do użytkowników na nie zarejestrowanych przesyłana jest wiadomość. Analogicznie po zakończeniu wydarzenia do użytkowników przesyłana jest wiadomość końcowa.

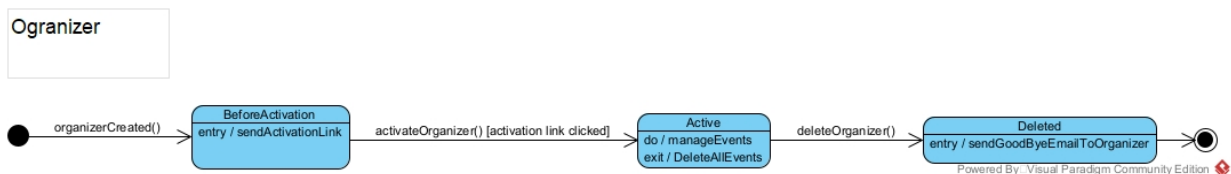
4.2.3 Reservation



Rysunek 11: Diagram stanów - Reservation

Rezerwacja w swoim domyślnym stanie po utworzeniu może wyświetlać informacje o sobie. Z tego stanu może wyjść na 3 różne sposoby: poprzez wygaśnięcie (przeterminowanie), poprzez anulowanie przez użytkownika oraz poprzez odwołanie eventu. Przy wyjściu z tego stanu wysyła powiadomienie do użytkownika.

4.2.4 Organizer

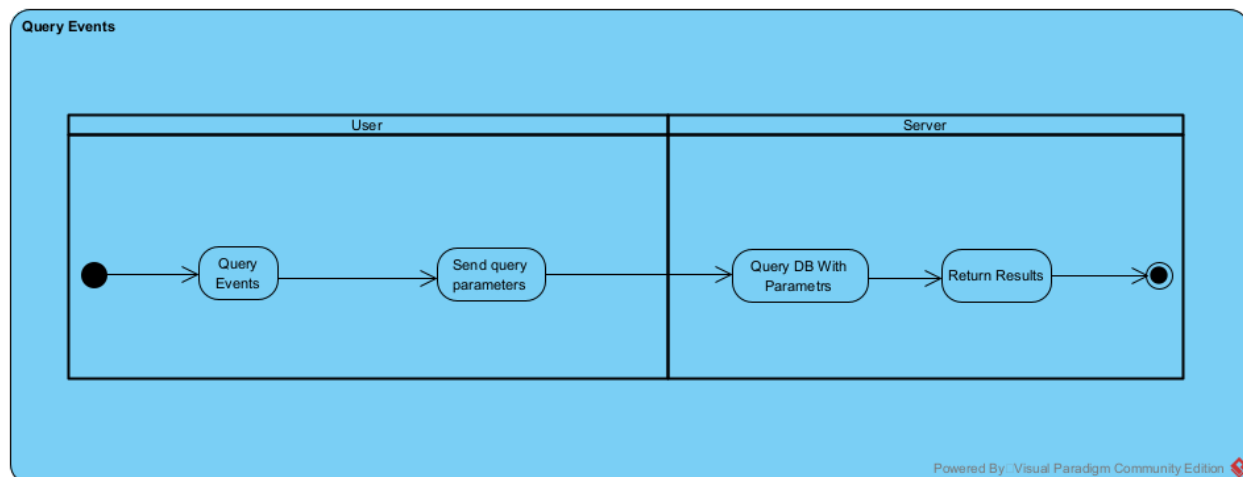


Rysunek 12: Diagram stanów - Organizer

Organizator po utworzeniu przechodzi do stanu BeforeActivation w którym dostaje e-maila z linkiem do aktywacji swojego konta. Aby przejść do kolejnego stanu czyli Active musi on w ten link kliknąć. W stanie aktywnym może on zarządzać swoimi Eventami czyli tworzyć nowe oraz modyfikować już istniejące. Może on też usunąć swoje konto. Skutkuje to przejściem ze stanu Active do Deleted. Usunięcie konta skutkuje automatycznym odwołaniem wszystkich eventów stworzonych przez danego organizatora oraz wysłanie maila z potwierdzeniem usunięcia konta.

5 Aktywności w systemie

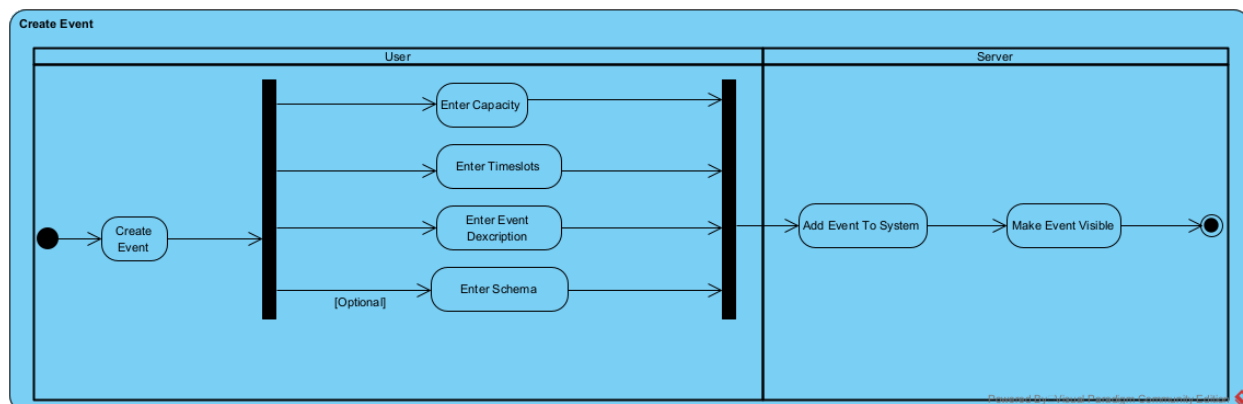
5.1 Przeglądanie eventów



Rysunek 13: Diagram aktywności - Przeglądane eventów

Liste wydarzeń w systemie można pobrać. Można pobrać listę wszystkich eventów lub event o poszczególnym ID. Ponadto można pobrać eventy należące do konkretnej kategorii.

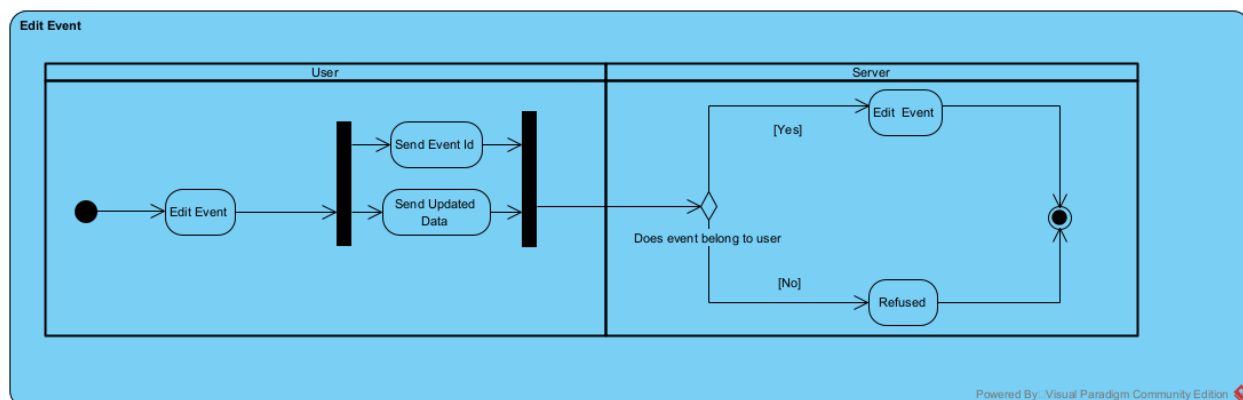
5.2 Tworzenie eventu



Rysunek 14: Diagram aktywności - Tworzenie eventu

Tworzenie eventu wymaga przesłanie danych (ilość miejsc, okna czasowe, opis eventu i opcjonalnie schematu miejsc zależnie od tego czy event ma go posiadać), do serwera, następnie event jest dodawany do systemu a potem jest ustawiany jako widoczny dla użytkowników.

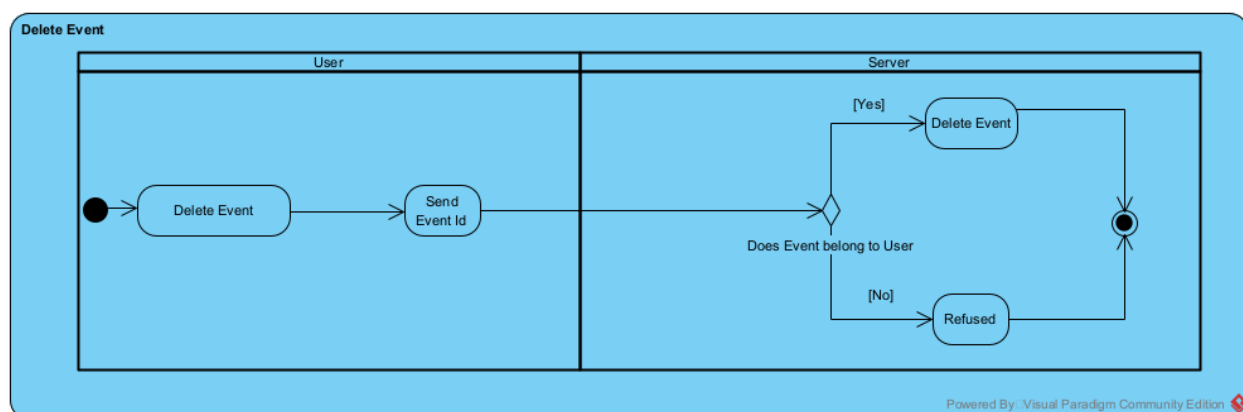
5.3 Edycja eventu



Rysunek 15: Diagram aktywności - Edycja eventu

Edycja eventu pozwala na zmianę danych wydarzenia (nazwy krótkiej i długiej, miejsca, przesunięcie daty w przód). Zmiany można dokonać jedynie dla eventów które jeszcze się nie rozpoczęły. Organizator może edytować tylko własne eventy

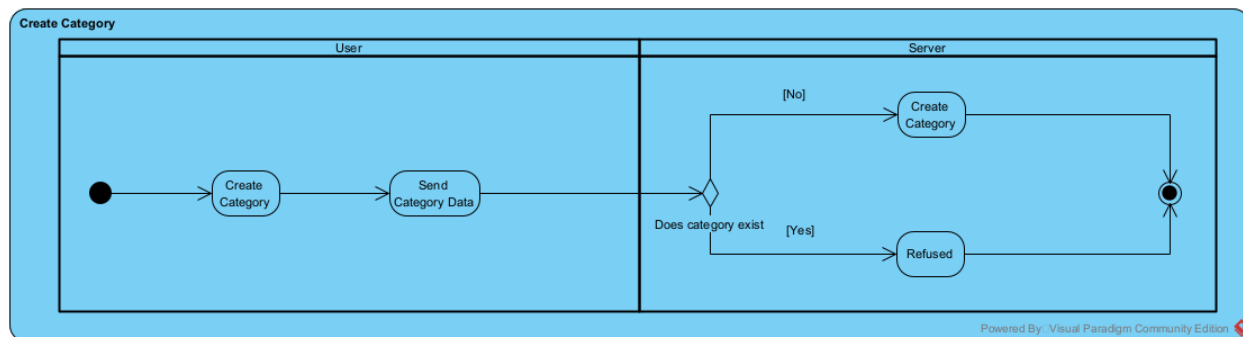
5.4 Anulowanie eventu



Rysunek 16: Diagram aktywności - Usuwanie eventu

Anulowanie eventu polega na oznaczeniu eventu jako cancelled. Anulować można jedynie eventy jeszcze nie rozpoczęte. Organizator może anulować tylko własne eventy.

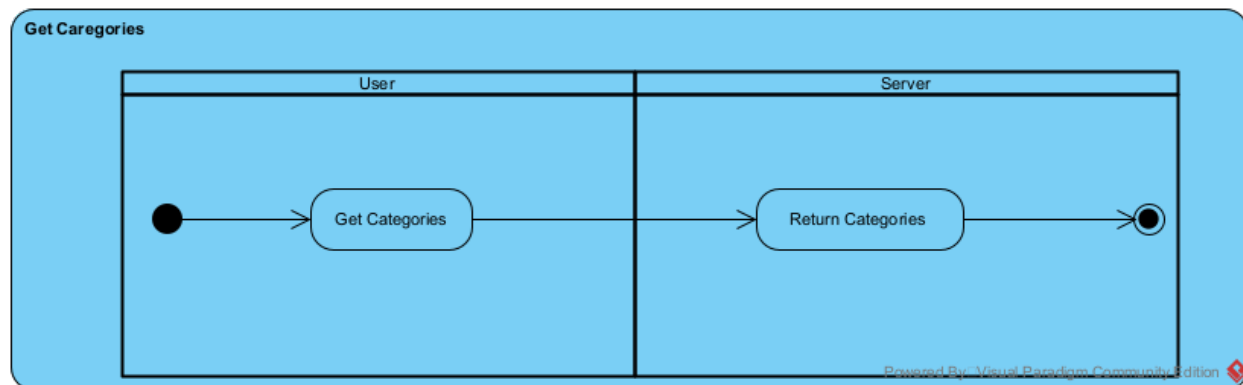
5.5 Tworzenie kategorii



Rysunek 17: Diagram aktywności - Tworzenie kategorii

Tworzenie kategorii pozwala na dodanie do systemu nowej kategorii, w której może znaleźć się wydarzenie. Nie można dodać kategorii już istniejącej. Każdy organizator może dodać kategorie. Kategorie po tym względem przypominają system tagów.

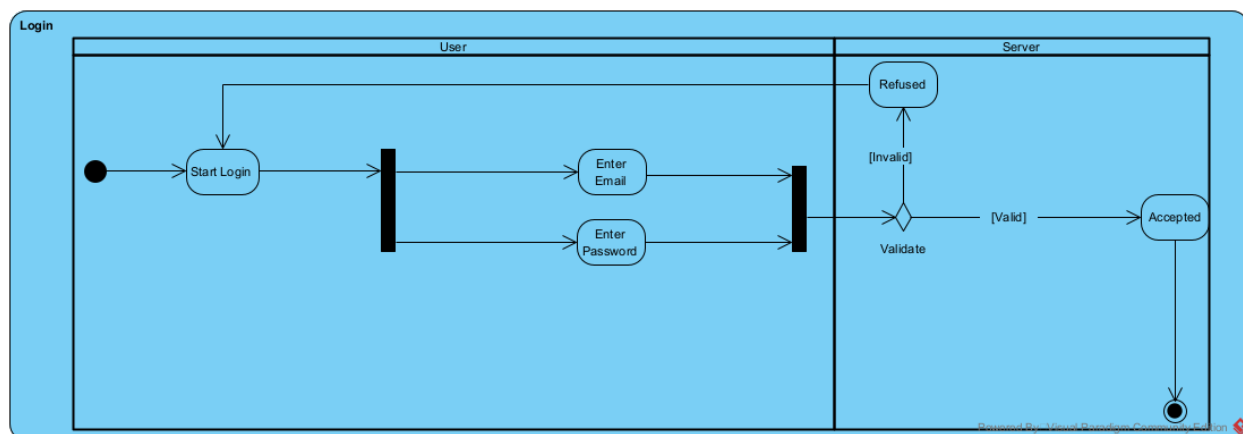
5.6 Przeglądanie kategorii



Rysunek 18: Diagram aktywności - Przeglądanie kategorii

Listę kategorii obecnych w systemie można pobrać korzystając z pojedynczego zapytania.

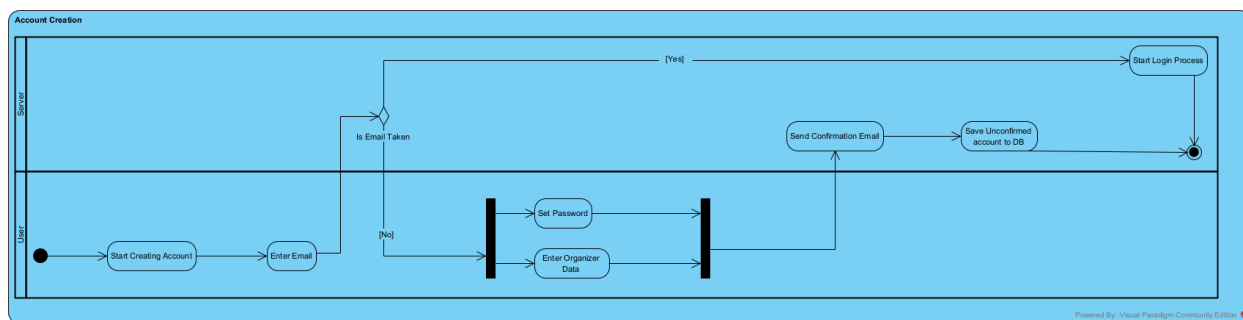
5.7 Logowanie do konta organizatora



Rysunek 19: Diagram aktywności - Login

Logowanie do panelu organizatora wymaga podania odpowiednich danych logowania (emailu i hasła), następnie walidowane są one przeciwko zawartości bazy, w przypadku powodzenia klient uzyskuje dostęp do panelu, w przeciwnym przypadku musi powtórzyć procedurę.

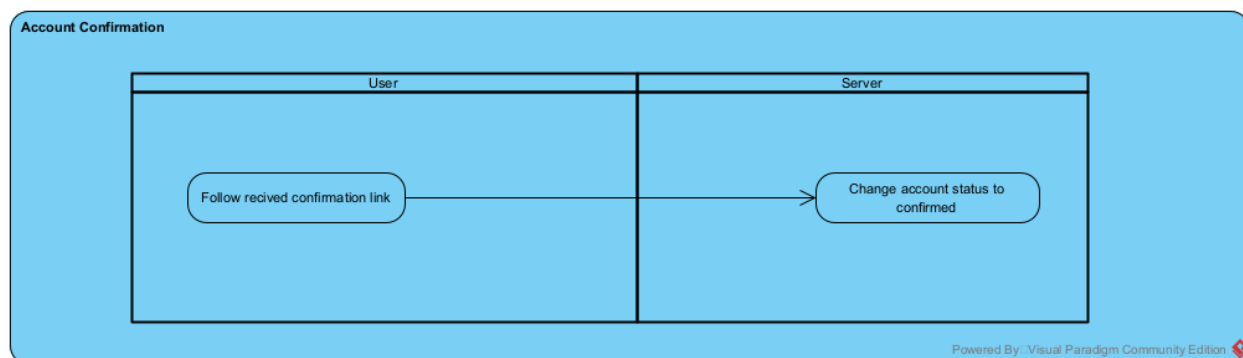
5.8 Tworzenie konta organizatora



Rysunek 20: Diagram aktywności - Tworzenie konta organizatora

Tworzenie konta organizatora wymaga wprowadzenia do systemu maila który jest następnie walidowany przeciwko zawartości bazy. Jeżeli jest to email występujący już w bazie, użytkownik przekierowany zostanie do procesu logowania. W przeciwnym przypadku użytkownik będzie musiał ustawić hasło i pozostałe dane organizatora. Powoduje to wysłanie maila potwierdzającego.

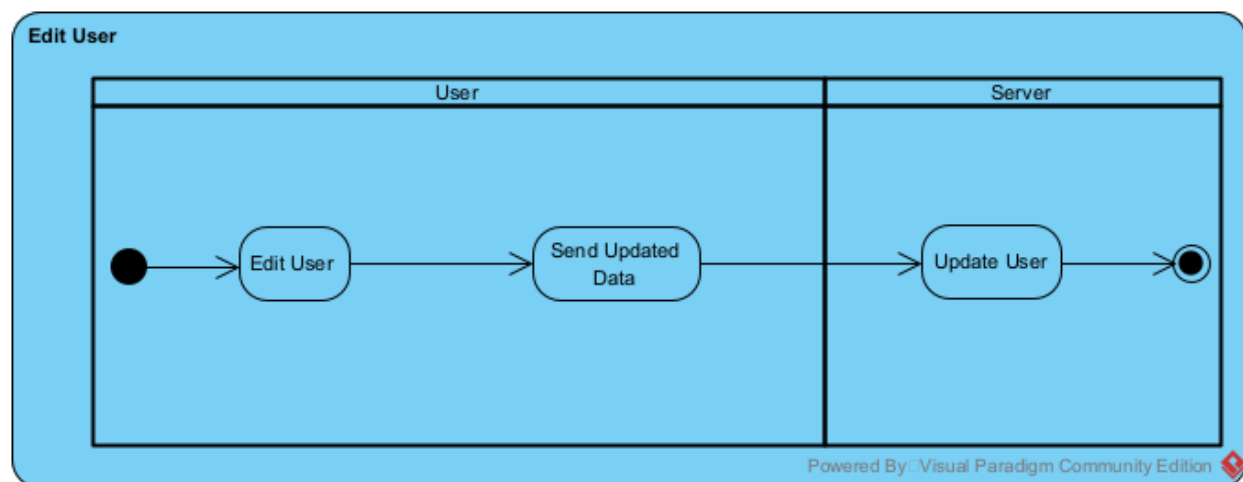
5.9 Potwierdzenie konta organizatora



Rysunek 21: Diagram aktywności - potwierdzenie konta organizatora

Po wprowadzeniu danych konto organizatora nie dostarcza wszystkich funkcjonalności i nie zostało jeszcze potwierdzone. W celu potwierdzenia konta organizatora należy przesłać request zawierający kod z maila.

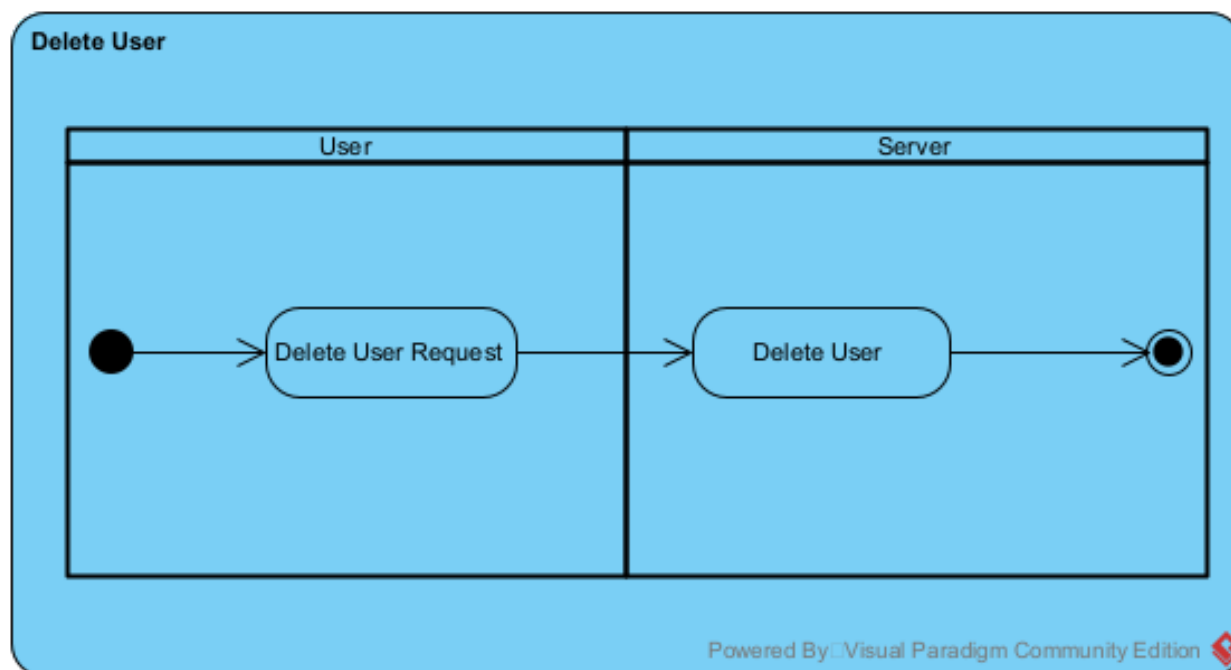
5.10 Edycja konta organizatora



Rysunek 22: Diagram aktywności - Edycja konta organizatora

Edycja konta organizatora umożliwia edycje danych organizatora. Zmienić można nazwę, email i hasło. Organizator może edytować można jedynie swoje konto.

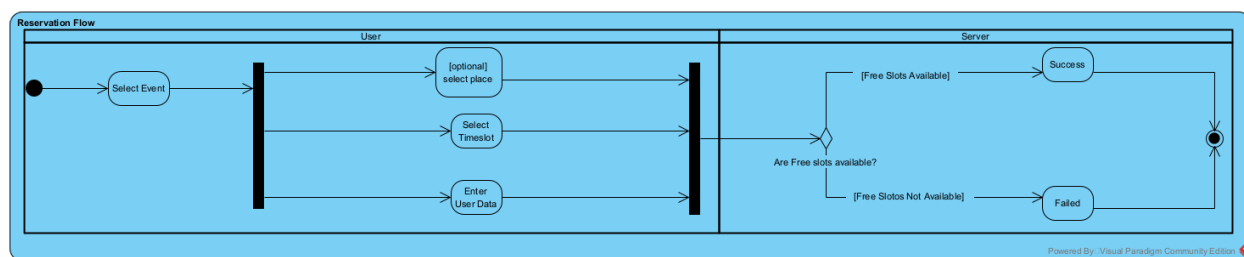
5.11 Usuwanie konta organizatora



Rysunek 23: Diagram aktywności - Edycja eventu

Organizator ma również możliwość usunięcia swojego konta. Może to zrobić po zalogowaniu. Aby zachować koherencję systemu konto oczywiście nie zostanie usunięte jeśli organizator posiada niezakończony event. Ponadto konto nie zostaje w pełni usunięte lecz zahashowane i pozbawione danych wrażliwych.

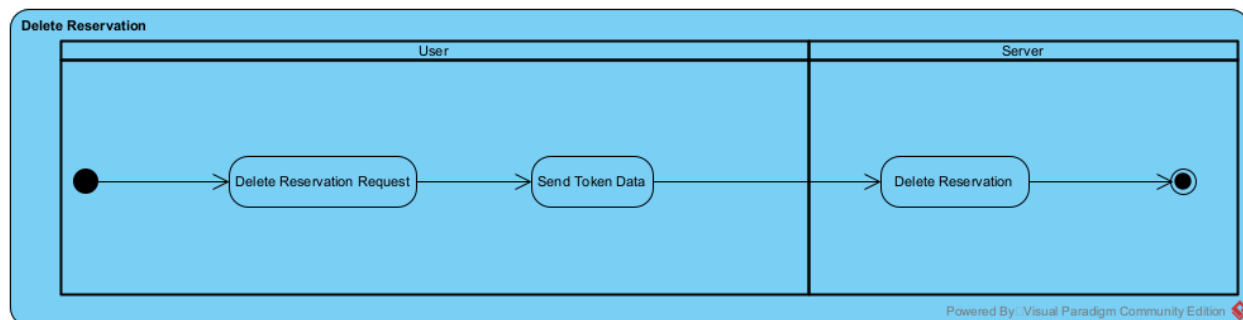
5.12 Tworzenie rezerwacji



Rysunek 24: Diagram aktywności - Rezerwacja

Rezerwacja wymaga wybrania eventu a następnie wprowadzenia danych użytkownika, wybrania okna czasowego i opcjonalnie miejsca, jeżeli event posiada schemat miejsc. Później wybór jest walidowany przeciwko danym dostępnym w bazie i w zależności od wyniku walidacji operacja rezerwacji powodzi się lub nie.

5.13 Usunięcie rezerwacji



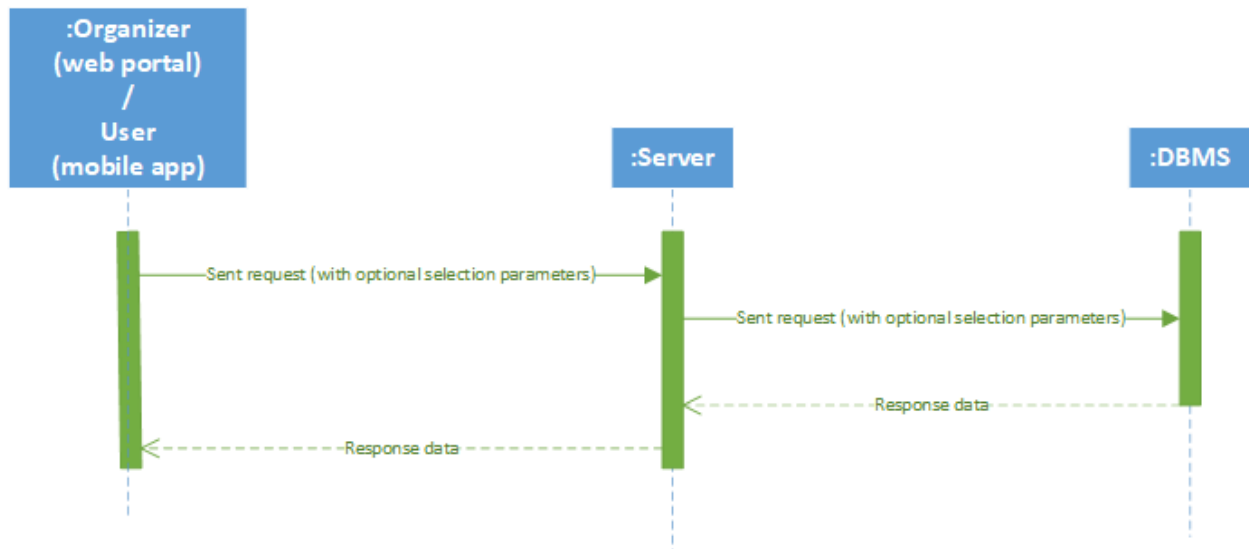
Rysunek 25: Diagram aktywności - Edycja eventu

Usunięcie rezerwacji pozwala na przywrócenia wolnego miejsca w ewencie. Do tego niezbędny jest token otrzymany przy tworzeniu rezerwacji.

6 Diagramy sekwencji

6.1 Przeglądanie eventów

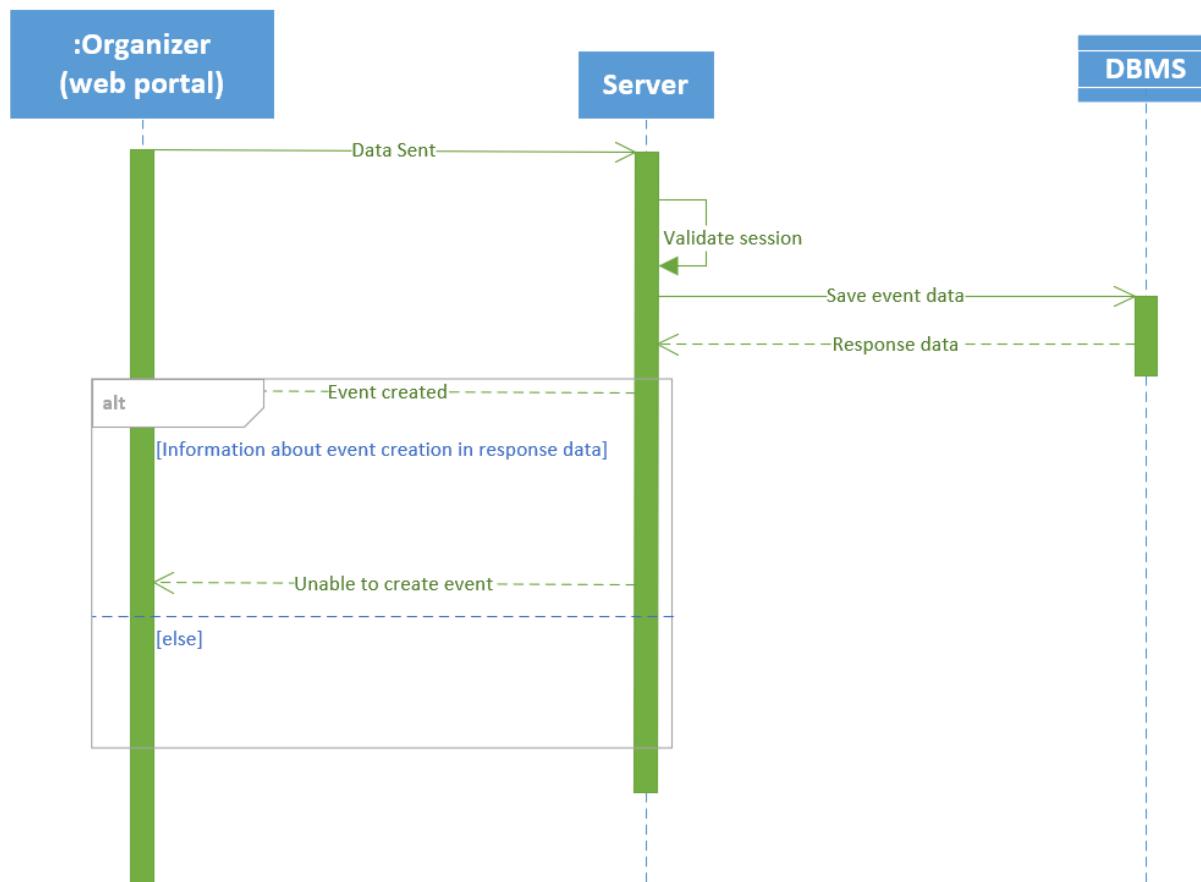
Użytkownicy mogą pobierać listę dostępnych eventów. Można również sprecyzować który event chcemy pobrać lub do jakiej kategorii ma on należeć. W tym celu z portalu webowego lub aplikacji do serwera wysyłane jest zapytanie, z opcjonalnymi parametrami. Server pobiera dane z bazy danych i zwraca wynik do użytkownika.



Rysunek 26: Diagram sekwencji - Przeglądanie eventów

6.2 Tworzenie eventu

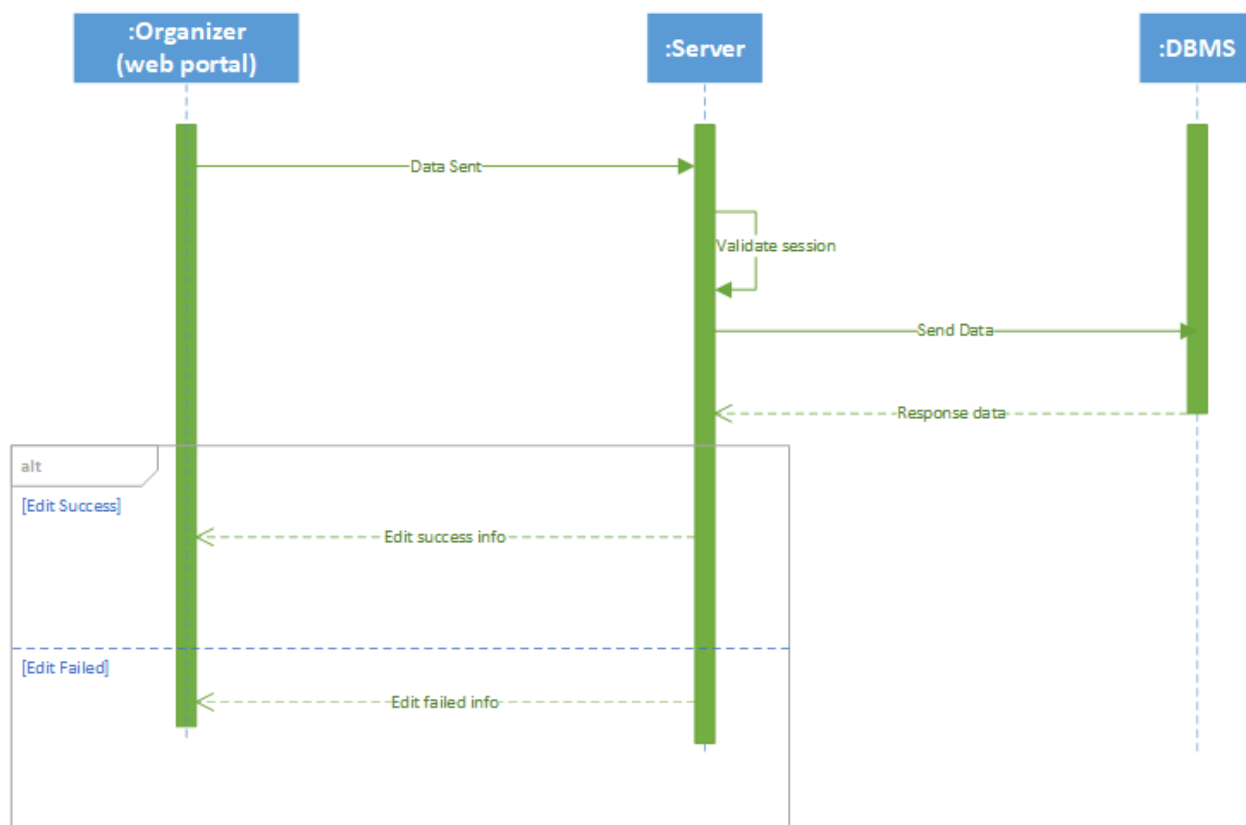
Portal organizatora przesyła dane eventu do serwera (tytuł - krótki opis, nazwa - długi opis, liczbę dostępnych miejsc, opcjonalny schemat miejsc, czas rozpoczęcia, czas zakończenia, kategorie). Następnie serwer wysyła zapytanie do bazy danych w celu zapisania danych. W przypadku uzyskania odpowiedzi Success serwer przesyła wiadomość do portalu o stworzeniu eventu, zaś event jest dostępny na portalu. W przypadku uzyskania odpowiedzi błędu (powtórzona nazwa eventu dla danego organizatora) serwer wysyła komunikat błędu do portalu, zaś event nie został utworzony i nie jest dostępny.



Rysunek 27: Diagram sekwencji - Tworzenie eventu

6.3 Edycja eventu

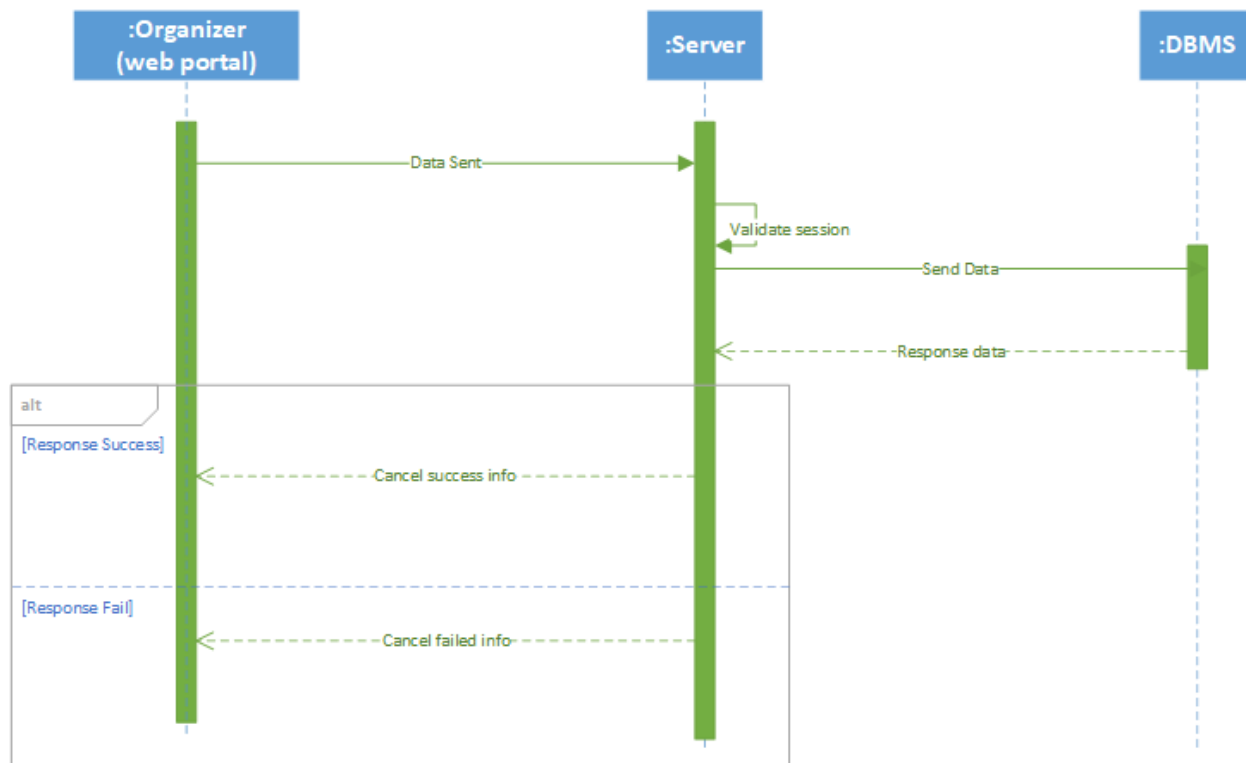
Edycja eventu polega na zmianie pól w obiekcie eventu. Przesyłając odpowiednie DTO rządzą zmiany w bazie. Przed ich wprowadzeniem sprawdzana jest poprawność tokenu sesji oraz przynależność eventu do konta z którego zarządzano zmiany. W zależności od wyniku autoryzacji event jest edytowany lub nie.



Rysunek 28: Diagram sekwencji - Edycja eventu

6.4 Anulowanie eventu

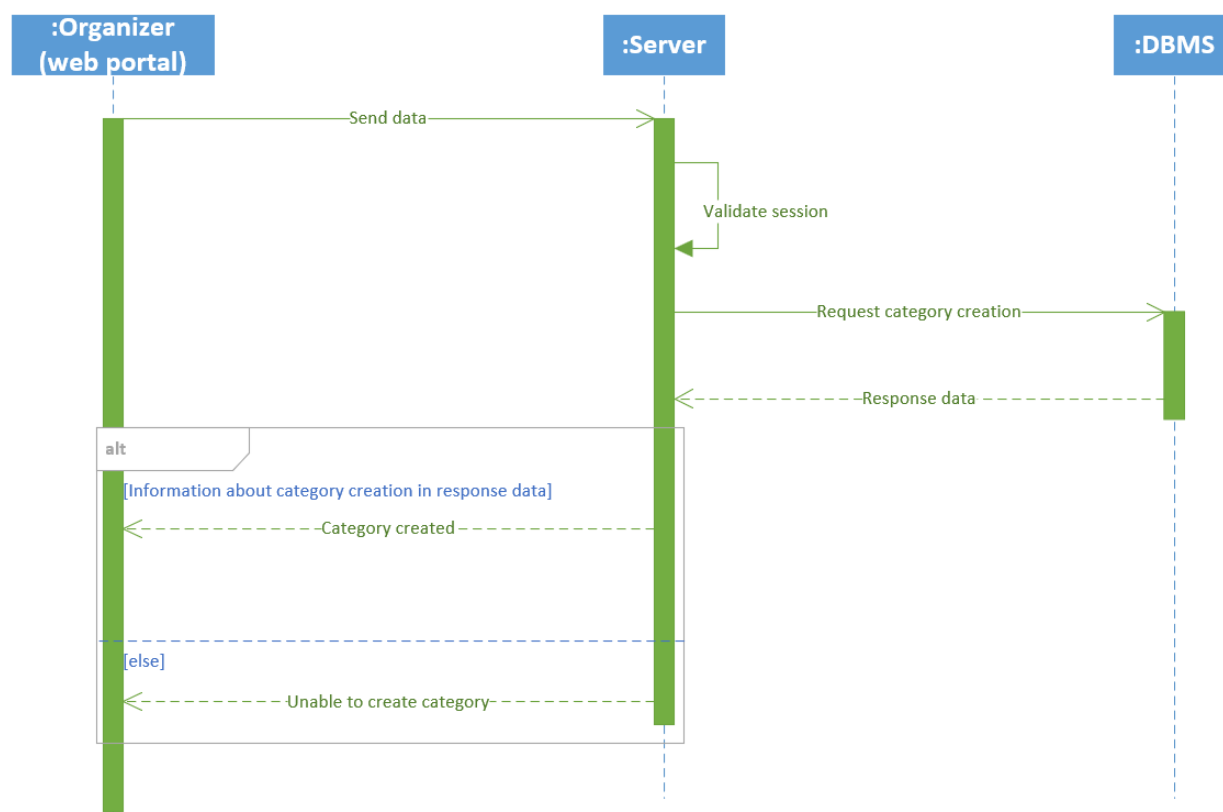
Edycja eventu polega na oznaczeniu eventu jako anulowany. Anulować można jedynie własne i nierozpoczęte wydarzenia. Po autoryzacji, serwer dokonuje zmiany w bazie danych lub zwraca błąd.



Rysunek 29: Diagram sekwencji - Anulowanie eventu

6.5 Tworzenie kategorii

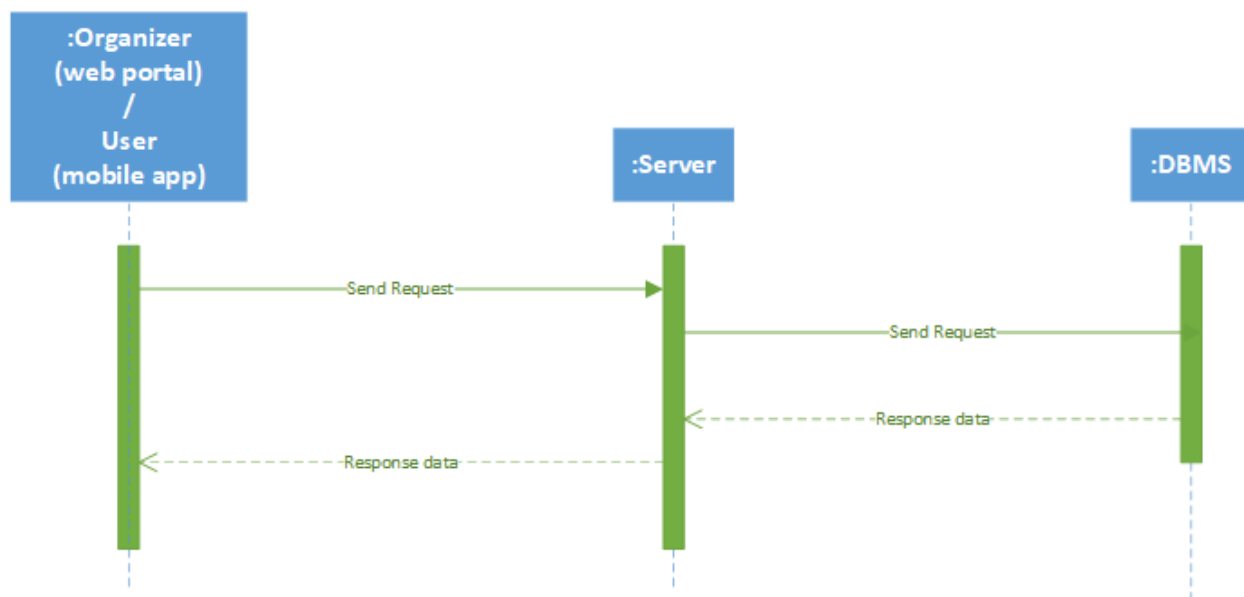
Portal organizatora przesyła dane kategorii do serwera (nazwa). Następnie serwer wysyła zapytanie do bazy danych w celu zapisania danych. W przypadku uzyskania odpowiedzi Success serwer przesyła wiadomość do portalu o stworzeniu kategorii, zaś utworzona kategoria jest ogólnodostępna w ramach portalu. W przypadku uzyskania odpowiedzi błędu (kategoria już istnieje) serwer wysyła stosowny komunikat do portalu, zaś kategoria nie zostaje utworzona i nie jest dostępna.



Rysunek 30: Diagram aktywności - Tworzenie kategorii

6.6 Przeglądanie kategorii

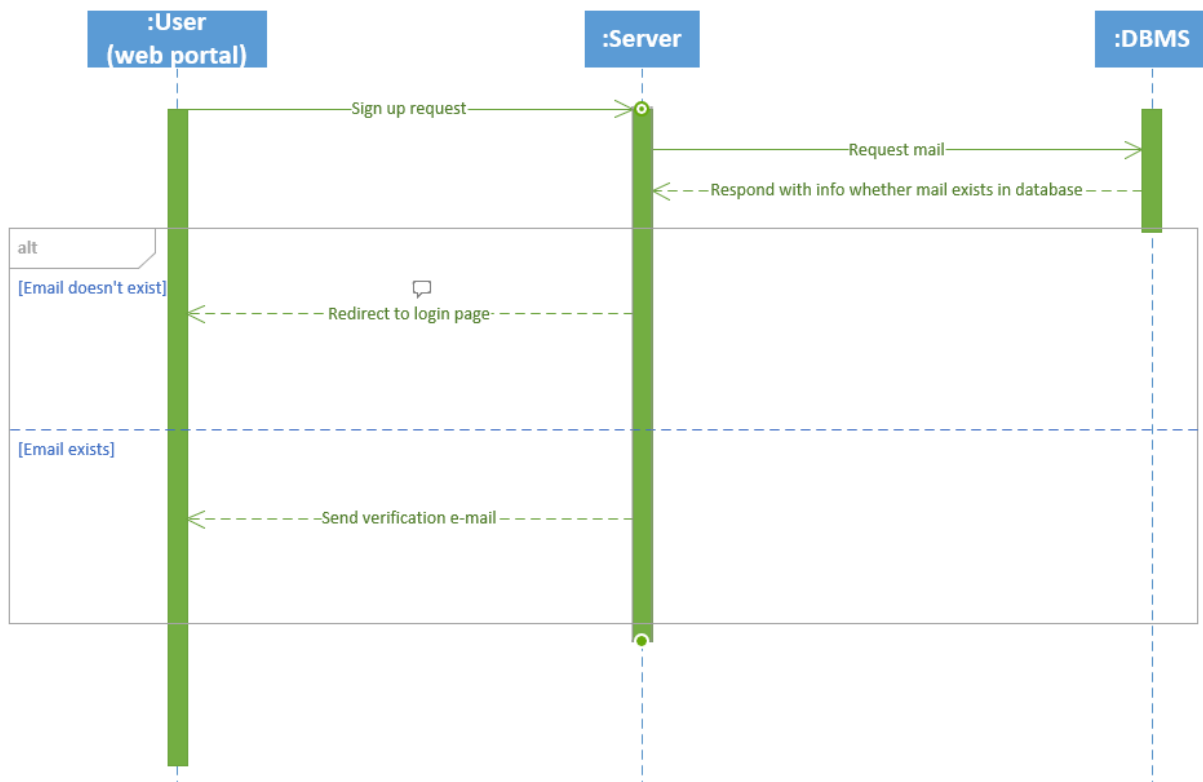
Przeglądanie kategorii polega na pobraniu listy wszystkich kategorii w bazie. Poza krytycznym błędem serwera lub bazy zawsze zwraca liste kategorii.



Rysunek 31: Diagram sekwencji - Przeglądanie kategorii

6.7 Rejestracja

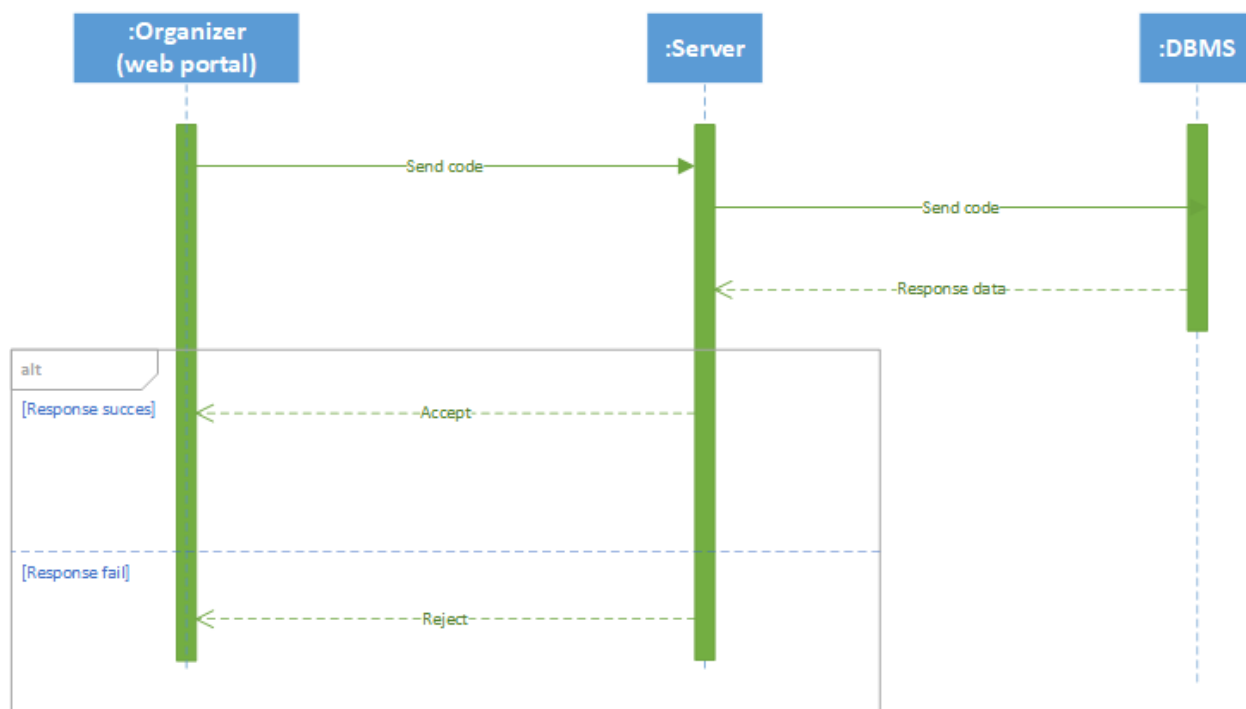
Portal organizatora wysyła wiadomość do serwera z danymi do rejestracji (nazwa, adres email, hasło). Następnie serwer wysyła zapytanie do bazy danych w celu weryfikacji czy istnieje już konto z danym adresem email. Po otrzymaniu odpowiedzi serwer jeśli adres email jest zajęty to serwer wysyła do portalu wiadomość o przekierowaniu do strony logowania. Jeśli adres email jest dostępny serwer wysyła do usera email weryfikacyjny. Po zweryfikowaniu serwer wysyła zapytanie do bazy w celu zapisania tam danych konta. Po powrocie z bazy z sukcesem, user poprawnie utworzył konto i może z niego korzystać.



Rysunek 32: Diagram aktywności - Rejestracja

6.8 Potwierdzenie konta

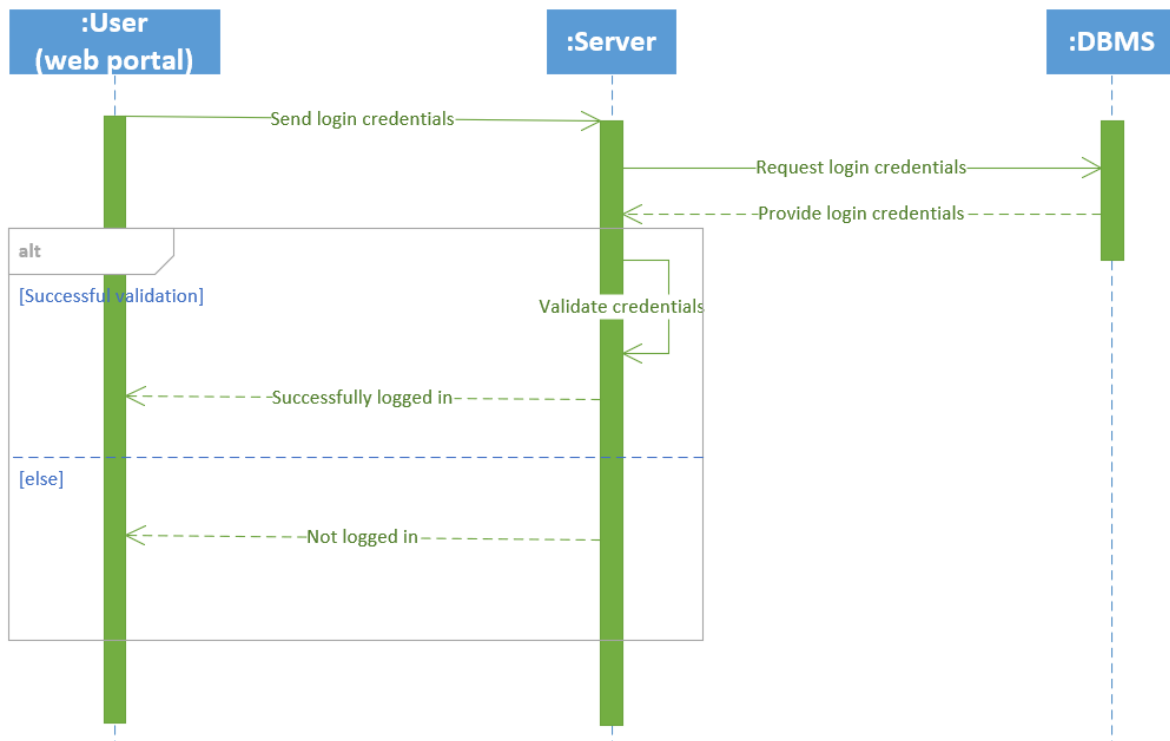
Potwierdzenie konta ma na celu potwierdzenie wiarygodności adresu email podanego w rejestracji. Organizator przesyła do servera kod, a ten sprawdzany jest bazie celem jego zweryfikowania. W odpowiedzi zwraca akceptację lub odrzucenie kodu.



Rysunek 33: Diagram sekwencji - Potwierdzenie konta

6.9 Logowanie

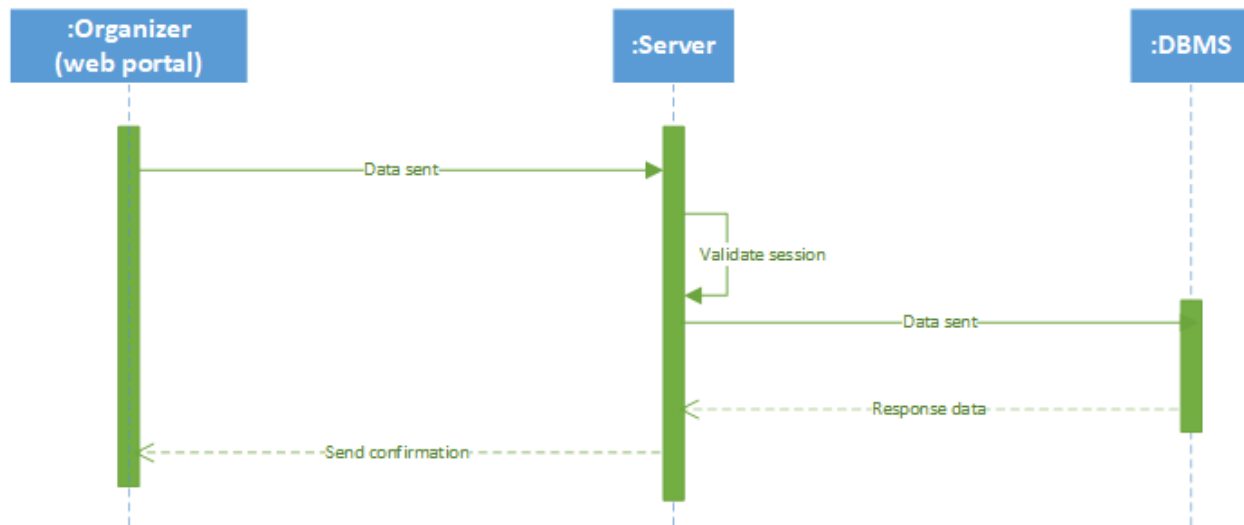
Portal organizatora wysyła wiadomość do serwera z danymi logowania (adres email, hasło). Następnie serwer wysyła zapytanie do bazy danych aby uzyskać dane logowania z bazy. Następnie serwer dokonuje weryfikacji danych uzyskanych od użytkownika i bazy danych. Jeśli walidacja zakończyła się sukcesem serwer przesyła do portalu informację o poprawnym zalogowaniu, w przeciwnym przypadku wysyła informację o nieudanym logowaniu.



Rysunek 34: Diagram sekwencji - Logowanie

6.10 Edycja konta

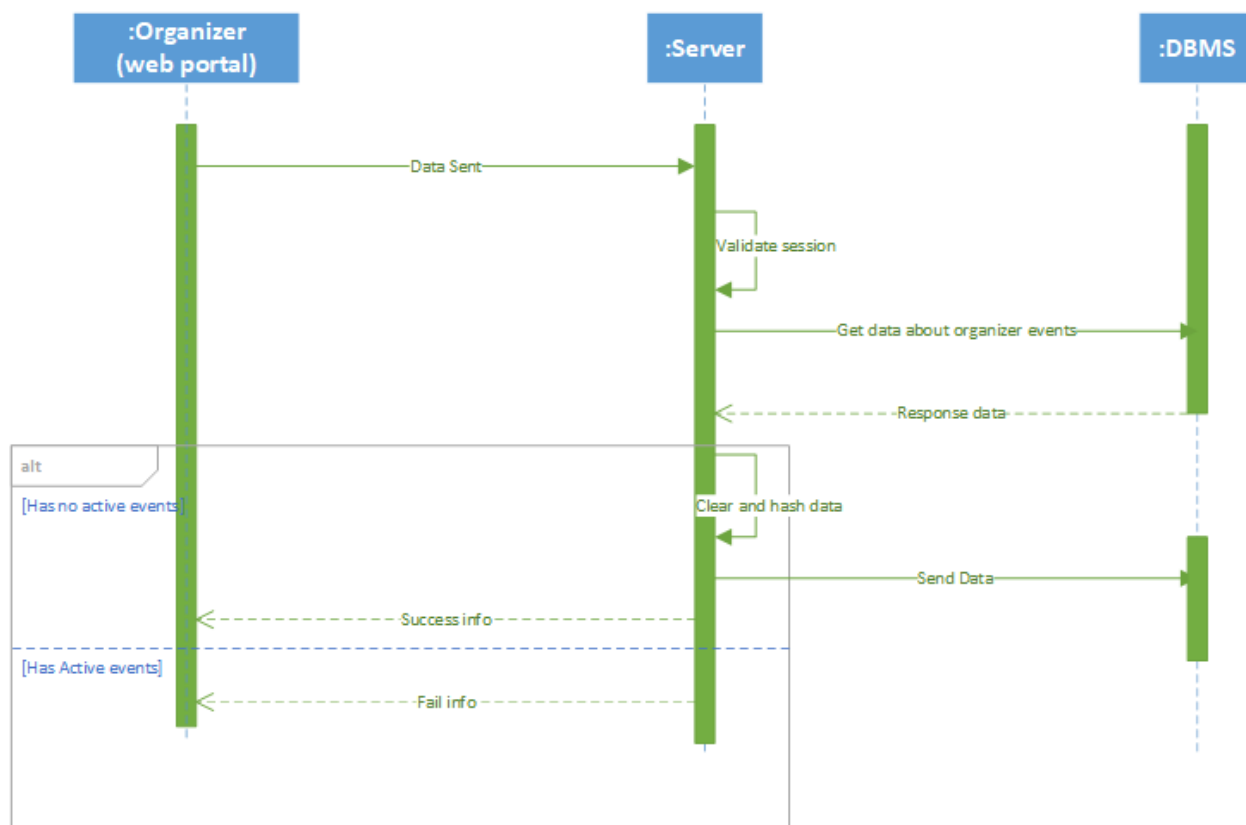
Zalogowany organizator przesyła obiekt DTO zawierający dane które mają ulec zastąpieniu. Serwer wprowadza dane do bazy i odsyła potwierdzenie.



Rysunek 35: Diagram sekwencji - Edycja konta

6.11 Usunięcie konta

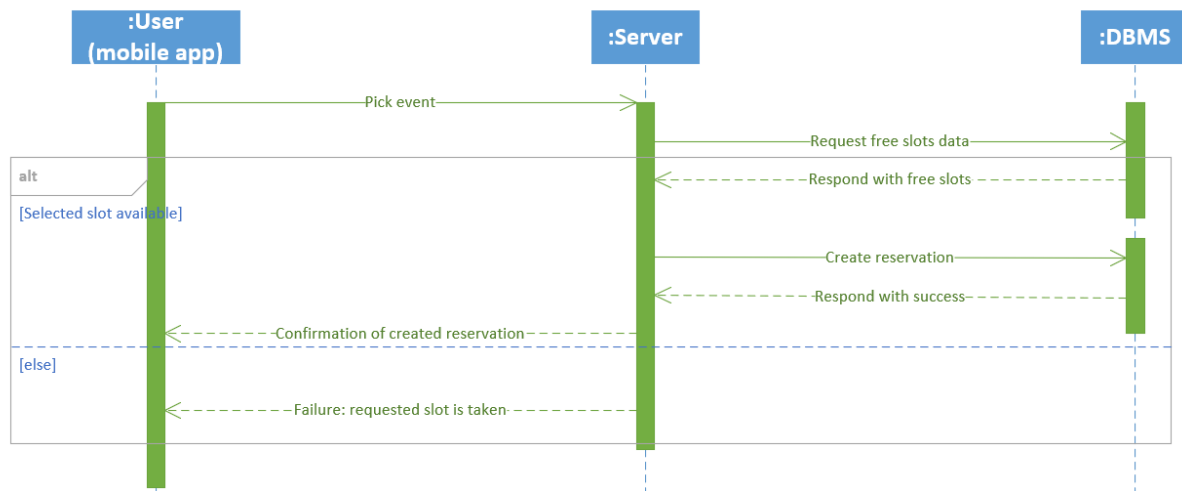
Zalogowany organizator może zażądać usunięcia konta. W pierwszej kolejności sprawdzane są uprawnienia do wykonania operacji oraz czy do konta nie są przypisane aktywne eventy. Serwer pobiera dane z BD i oczyszcza je z danych wrażliwych. Zahashowane dane wprowadza do bazy i zwraca potwierdzenie operacji



Rysunek 36: Diagram sekwencji - Usunięcie konta

6.12 Tworzenie rezerwacji

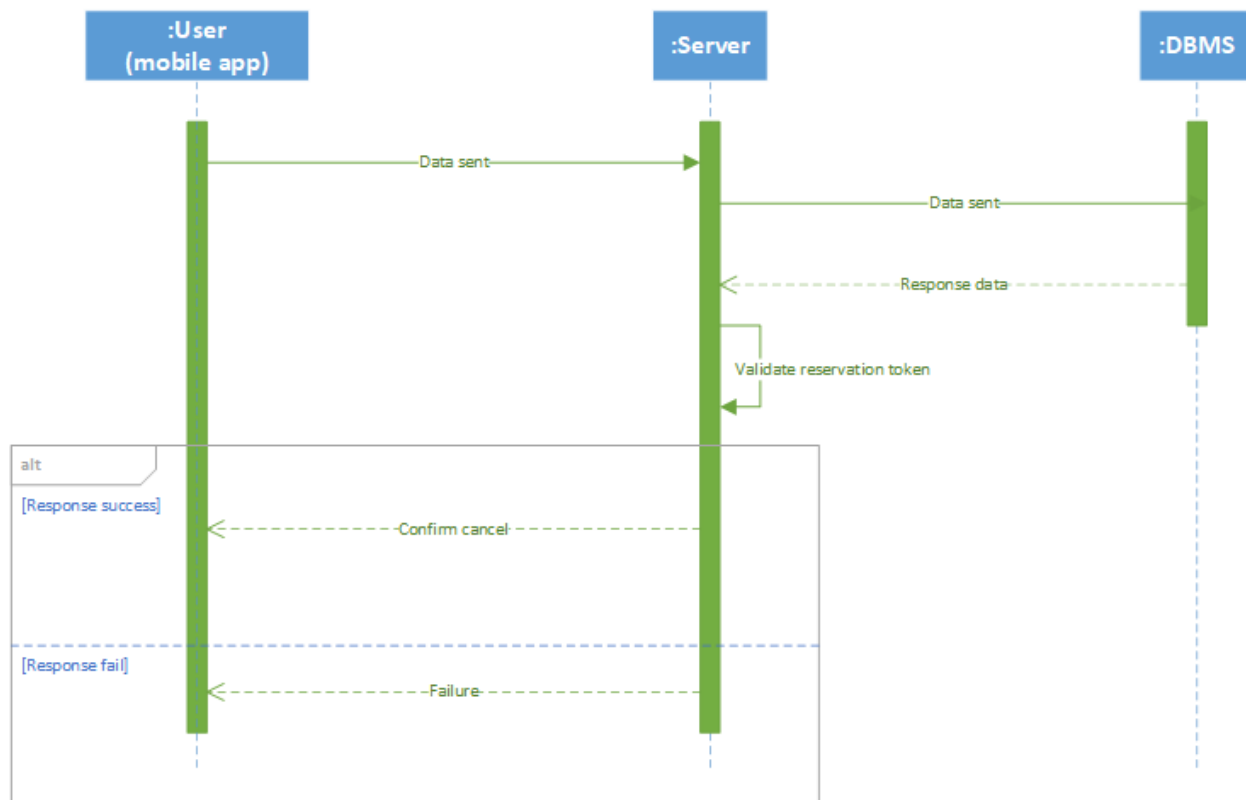
Aplikacja usera przesyła wiadomość do serwera z danymi rezerwacji (Id eventu, Id miejsca). Serwer przesyła zapytanie do bazy danych czy istnieją dostępne miejsca. Jeśli tak to rezerwacja jest zapisana w bazie danych i serwer przesyła do aplikacji usera potwierdzenie stworzenia rezerwacji. Jeśli nie serwer przesyła do aplikacji usera informację o braku wolnych miejsc.



Rysunek 37: Diagram aktywności - Rezerwacja

6.13 Usunięcie rezerwacji

Uczestnik dysponujący tokenem rezerwacji może wycofać ją z systemu umożliwiając tym samym zarezerwowanie miejsca przez inną osobę. Użytkownik przesyła do systemu informacje o rezerwacji z tokenem i jeśli są one poprawne rezerwacja zostanie wycofana. W przeciwnym wypadku zwrócony zostanie błąd. Wycofać można rezerwacje jedynie na wydarzenia jeszcze nie rozpoczęte.



Rysunek 38: Diagram sekwencji - Usunięcie rezerwacji

7 Komunikacja w systemie

7.1 Obiekty komunikacji

7.1.1 Event

Obiekt Event odwzorowuje wydarzenie w systemie. Obiekt wykorzystywany jest przy tworzeniu, aktualizowaniu i pobieraniu wydarzenia z bazy. W przypadku gdy niektóre z pól nie są wykorzystywane, przyjmują one wartość null lub nie występują w JSON-ie.



Event ▾ {	
id*	integer(\$int64) example: 10
freePlace	integer(\$int64) example: 2
title	string example: Short description of Event
startTime	integer(\$int64) example: 1673034164
endTime	integer(\$int64) example: 1683034164
name	string example: Long description of Event
placeSchema	string example: Seralized place schema
status	EventStatus string example: done event status Enum: ▾ [inFuture, pending, done, cancelled]
categories	▾ [Category > {...}]
}	

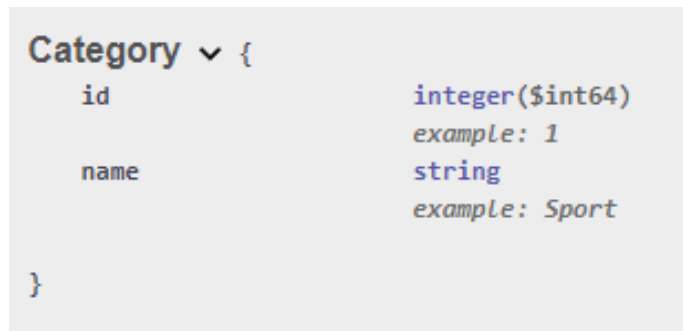
Rysunek 39: Pola obiektu Event z określeniem typów pól

Przykładowy JSON:

```
{
  "id": 10,
  "freePlace": 2,
  "title": "Short description of Event",
  "startTime": 1673034164,
  "endTime": 1683034164,
  "name": "Long description of Event",
  "placeSchema": "Seralized place schema",
  "status": "done",
  "categories": [
    {
      "id": 1,
      "name": "Sport"
    }
  ]
}
```

7.1.2 Category

Obiekt Category reprezentuje kategorię wydarzenia. Kategorie służą do filtrowania wydarzeń. Wydarzenie może należeć jednocześnie do wielu kategorii. Do tworzenia nowych kategorii ma prawo każdy zalogowany organizator.



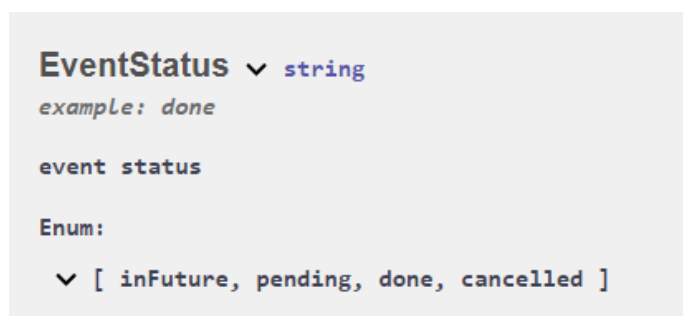
Rysunek 40: Pola obiektu Category z określeniem typów pól

Przykładowy JSON:

```
{  
  "id": 1,  
  "name": "Sport"  
}
```

7.1.3 EventStatus

Obiekt EventStatus reprezentuje status wydarzenia. Jest to typ wyliczeniowy przyjmujący jeden z ze stanów: planowane, trwające, zakończone, odwołane.



Rysunek 41: Pola obiektu EventStatus z określeniem typów pól

7.1.4 ReservationDTO

Obiekt ReservationDTO to DTO zwracane uczestnikowi wydarzenia po pomyślnym dokonaniu rezerwacji. Zawiera wszystkie informacje dotyczące rezerwacji i powinien być przechowywany w pamięci urządzenia mobilnego.

```
ReservationDTO {  
  eventId          integer($int64)  
                  example: 1  
  placeId          integer($int64)  
                  example: 12  
  reservationToken string  
                  example: df0d69cbe68fb6e2b27aa88f6f94497e  
}
```

Rysunek 42: Pola obiektu ReservationDTO z określeniem typów pól

Przykładowy JSON:

```
{  
  "eventId": 1,  
  "placeId": 12,  
  "reservationToken": "df0d69cbe68fb6e2b27aa88f6f94497e"  
}
```

7.1.5 Organizer

Obiekt Organizer odwzorowuje organizatora wydarzenia w systemie. Obiekt wykorzystywany jest przy tworzeniu, aktualizowaniu konta organizatora. Zawiera informacje personalne organizatora i listę nadzorowanych przez niego wydarzeń. W przypadku gdy niektóre z pól nie są wykorzystywane, przyjmują one wartość null lub nie występują w JSON-ie.

Organizer ▾ {	
id	integer(\$int64) example: 10
name	string example: theUser
email	string example: john@email.com
password	string example: 12345
events	▾ [Event > {...}]
status	string User Status
Enum:	
▸ Array [2]	
}	

Rysunek 43: Pola obiektu Organizer z określeniem typów pól

Przykładowy JSON:

```
{
  "id": 10,
  "name": "theUser",
  "email": "john@email.com",
  "password": "12345",
  "events": [
    {
      "id": 10,
      "freePlace": 2,
      "title": "Short description of Event",
      "startTime": 1673034164,
      "endTime": 1683034164,
      "name": "Long description of Event",
      "placeSchema": "Serialized place schema",
      "status": "done",
      "categories": [
        {
          "id": 1,
          "name": "Sport"
        }
      ]
    }
  ],
  "status": "pending"
}
```

7.2 Endpointy

7.2.1 Dot. konta organizatora

Metoda: POST, ścieżka: /organizer

Endpoint umożliwia utworzenie nowego konta organizatora. Utworzone w ten sposób konto jest nieaktywne/oczekujące i wymaga powierzenia kodem przesłanym na email.

Parametry:

- name (string) - nazwa organizatora, wymagany
- email (string) - email organizatora, wymagany
- password (string) - hasło dla tworzonego konta, wymagany

Request body: *brak*

Odpowiedzi:

- Kod 201 - Poprawnie utworzono konto organizatora, odpowiedź zawiera obiekt Organizer (application/json).
- Kod 400 - Konto organizatora o podanych danych już istnieje.

Metoda: POST, ścieżka: /organizer/{id}

Endpoint umożliwia potwierdzenie nowozałożonego konta przez przekazanie kodu z wiadomości.

Parametry:

- id (string) - id organizatora, wymagany
- code (string) - kod potwierdzający otrzymany w mailu, wymagany

Request body: *brak*

Odpowiedzi:

- Kod 201 - Poprawnie utworzono konto organizatora, odpowiedź zawiera obiekt Organizer (application/json).
- Kod 400 - Błędny kod

Metoda: DELETE, ścieżka: /organizer/{id}

Endpoint umożliwia organizatorowi usunięcie swojego konta. Usunięcie powoduje, że wszystkie wydarzenia związane z organizatorem zostają anulowane, a konto zarchiwizowane (pozbowione danych osobowych).

Parametry:

- id (string) - id organizatora, wymagany, w ścieżce

Request body: *brak*

Autoryzacja: wymaga aktywanego tokenu sesji

Odpowiedzi:

- Kod 204 - Usunięto
- Kod 404 - Błędne id

Metoda: PATCH, ścieżka: /organizer/{id}

Endpoint umożliwia modyfikację wybranych personaliów i danych konta. Dane które podlegają edycji: name, email i password.

Parametry:

- id (string) - id organizatora, wymagany, w ścieżce

Request body: Obiekt Organizer, wartość nie nullową mają pola do aktualizacji.

Autoryzacja: wymaga aktywanego tokenu sesji

Odpowiedzi:

- Kod 202 - Zaktualizowano
- Kod 404 - Błędne id

Metoda: GET, ścieżka: /organizer/login

Endpoint służący do logowania się do konta organizatora. Zwraca token sesji ważny przez określony interwał czasu.

Parametry:

- email (string) - email organizatora, wymagany
- password (string) - hasło do konta, wymagany

Request body: *brak*

Odpowiedzi:

- Kod 200 - Sukces, zwrócony zostaje token sesji (string)
- Kod 400 - Błędna kombinacja email/hasło.

7.2.2 Dot. Eventu

Metoda: GET, ścieżka: /events

Endpoint umożliwia pobranie listy wszystkich eventów.

Parametry: *brak*

Request body: *brak*

Odpowiedzi:

- Kod 200 - Zwrócona została lista wszystkich Eventów (application/json)

Metoda: POST, ścieżka: /events

Endpoint umożliwia utworzenie nowego wydarzenia. Wiąże wydarzenie z organizatorem na podstawie tokenu sesji.

Parametry:

- title (string) - krótki opis eventu, wymagany
- name (string) - długi opis eventu, wymagany
- freePlace (integer) - liczba wolnych miejsc, wymagany
- placeSchema (string) - zserializowany schemat miejsc
- startTime (integer) - unixowy timestamp początku eventu, wymagany
- endTime (integer) - unixowy timestamp końca eventu, wymagany
- categories (array[integer]) - lista kategorii, wymagany

Request body: *brak*

Autoryzacja: wymaga aktywanego tokenu sesji

Odpowiedzi:

- Kod 201 - Event został utworzony, odpowiedź zawiera obiekt Event (application/json).
- Kod 400 - Event nie mógł zostać utworzony.

Metoda: GET, ścieżka: /events/{id}

Endpoint umożliwia pobranie pojedynczego eventu o wskazanym ID.

Parametry:

- id (integer) - ID eventu, wymagany, w ścieżce

Request body: *brak*

Odpowiedzi:

- Kod 200 - operacja zakończona sukcesem, odpowiedz zawiera obiekt Event (application/json).
- Kod 400 - Zostało dostarczone niepoprawne ID
- Kod 404 - Event nie został znaleziony

Metoda: DELETE, ścieżka: /events/{id}

Endpoint umożliwia usunięcie eventu należącego do organizatora. Usunięcie oznacza zmianę statusu wydarzenia na anulowane.

Parametry:

- id (integer) - ID eventu, wymagany, w ścieżce

Request body: *brak*

Autoryzacja: wymaga aktywanego tokenu sesji

Odpowiedzi:

- Kod 204 - operacja zakończona sukcesem, event został usunięty
- Kod 404 - Event nie został znaleziony

Metoda: PATCH, ścieżka: /events/{id}

Endpoint umożliwia zmianę danych wydarzenia. Edytować można jedynie wydarzenie nie-rozpoczęte i nieanulowane (o statusie inFuture). Edytować można pola: freePlace, startTime, title, name, endTime, placeSchema i categories.

Parametry:

- id (integer) - ID eventu, wymagany, w ścieżce

Request body:

- Event z wprowadzonymi nowymi wartościami (application/json)

Autoryzacja: wymaga aktywanego tokenu sesji

Odpowiedzi:

- Kod 202 - operacja zakończona sukcesem, event został zpatchowany
- Kod 404 - Event nie został znaleziony

Metoda: GET, ścieżka: /events/getByCategory

Endpoint umożliwia pobranie wszystkich eventów danej kategorii.

Parametry:

- id (integer) - ID eventu, wymagany

Request body: *brak*

Odpowiedzi:

- Kod 200 - operacja zakończona sukcesem, zwrócona lista Eventów (application/json)
- Kod 400 - invalid category ID supplied

7.2.3 Dot. kategorii

Metoda: GET, ścieżka: /categories

Endpoint umożliwia pobranie wszystkich istniejących kategorii.

Parametry: *brak*

Request body: *brak*

Odpowiedzi:

- Kod 200 - Zwrócona została lista wszystkich obiektów Category

Metoda: POST, ścieżka: /categories

Endpoint umożliwia dodanie nowej kategorii.

Parametry:

- categoryName (string) - nazwa kategorii (wymagany)

Request body: *brak*

Autoryzacja: wymaga aktywanego tokenu sesji

Odpowiedzi:

- Kod 201 - Kategoria została utworzona, odpowiedz zawiera obiekt Category (application/json).
- Kod 400 - Kategoria nie mogła zostać utworzona.

7.2.4 Dot. Rezerwacji

Metoda: POST, ścieżka: /reservation

Endpoint umożliwia złożenie rezerwacji. Zwraca pełne informacje o rezerwacji w postaci obiektu ReservationDTO.

Parametry:

- eventId (int) - Id eventu, wymagany
- placeId (int) - Id miejsca, wymagany

Request body: *brak*

Odpowiedzi:

- Kod 201 - Poprawnie utworzono rezerwację, odpowiedź zawiera obiekt ReservationDTO (application/json).
- Kod 400 - Brak wolnego miejsca.
- Kod 404 - Event nie istnieje lub się zakończył

Metoda: DELETE, ścieżka: /reservation

Endpoint umożliwia usunięcie rezerwacji o danym tokenie

Parametry:

- reservationToken (string) - token rezerwacji, wymagany

Request body: *brak*

Odpowiedzi:

- Kod 204 - Poprawnie usunięto rezerwację.
- Kod 404 - rezerwacja o podanym tokenie nie istnieje

7.3 API serwera w standardzie OpenAPI 3.0

W wersji 1.0 listing został usunięty. Najnowsze API znajdują się w dołączonym pliku JSON.