

Your Playlist

Dokument projektowy

Opis projektu

Nasz projekt – „Your Playlist” to system informatyczny, który łączy w sobie aplikację internetową jak i mobilną. Jest to aplikacja związana z muzyką. Jej celem jest umożliwienie użytkownikom wejście na wyższy poziom interaktywności w procesie tworzenia playlist i sprawienie by do tworzenia i edytowania playlist dodać aspekt pewnego rodzaju medium społecznościowego. Rdzeniem aplikacji są pokoje i playlisty. Użytkownik może stworzyć pokój, w którym rozpocznie tworzenie nowej playlisty. Może do niej dodawać utwory zarówno ze Spotify jak i z Soundcloud. Po utworzeniu pokoju może do niego zapraszać innych użytkowników, czy to poprzez udostępnienie im kodu do pokoju czy też przez wybranie ich z listy swoich znajomych i wysłanie im zaproszenia. Wszyscy użytkownicy pokoju mogą poprzez połączenie naszej aplikacji ze swoim kontem Spotify i Soundcloud dodawać utwory z tych platform do playlisty. Wszyscy członkowie danego pokoju mogą wyświetlić listę piosenek w playliście i głosować za konkretnymi piosenkami. Przy każdej piosence w playliście figurować będą dwie strzałki – jedna strzałka w górę i jedna strzałka w dół. O kolejności odtworzeń piosenek będzie decydowała suma głosów za daną piosenką. Każda nowo dodana do playlisty piosenka jak i ta która przed chwilą została odtworzona otrzymuje sumę głosów równą 0 i playlista zostaje posortowana względem liczby głosów. Każdy użytkownik może zapisać na swoim koncie daną playlistę i w przyszłości ponownie utworzyć na jej podstawie swój pokój.

Motywacja

Nasz produkt idealnie nadaje się do stosowania podczas spotkań towarzyskich i wszelkiego rodzaju wydarzeń, którym towarzyszy muzyka. Wystarczy, że do systemu nagłośnienia (n.p. głośnika bluetooth’owego) podłączy się tylko jedna osoba, utworzy pokój, do pokoju dołączą zainteresowani użytkownicy i w ten sposób każdy może z poziomu swojego telefonu modyfikować playlistę bez uzgadniania szczegółów z osobą której telefon (lub komputer) podłączony jest do nagłośnienia. W pewien sposób odciąża to tę osobę jako, że modyfikacji playlisty nie muszą poprzedzać żmudne konwersacje i dyskusje nad tym jaką piosenkę dodać do playlisty i którą odtworzyć w pierwszej kolejności. Proces ten jest znacznie przyspieszony i w pewien sposób zdemokratyzowany ponieważ o wszystkim decyduje liczba głosów za daną piosenką.

Dodatkowo playlisty mogą się składać z piosenek pochodzących ze Spotify i z Soundcloud. Te dwie platformy umożliwiają tworzenie playlist, ale tylko z piosenek będących w ich bazie danych. Nie każda piosenka istniejąca na jednej platformie istnieje na drugiej platformie. Nasz produkt rozwiązuje ten problem.

Czas realizacji projektu

Biorąc pod uwagę to, że zdecydowaliśmy się w części implementacyjnej na sprinty dwutygodniowe oraz to, że chcemy oprzeć swoją pracę na diagramie przypadków użycia chcemy zorganizować swoją pracę przypisując poszczególne przypadki użycia do konkretnych sprintów.

Przypadków użycia jest 26. W zbiorze wszystkich przypadków użycia istnieją podzbiory przypadków użycia powiązanych ze sobą. Chcemy każdy sprint przeznaczyć na cały jeden taki podzbiór.

Po wyróżnieniu tych podzbiorów można naliczyć ich 9. Czyli potrzebujemy $9 * 14$ dni co daje 126 dni. Czyli zaczynając pracę 28 lutego 2022 roku, czyli wtedy gdy zaczyna się semestr przeznaczony na implementację powinniśmy skończyć pracę na pewno nie później niż 4 lipca 2022 roku.

Spis członków projektu wraz z zakresem obowiązków

W semestrze przeznaczonym na planowanie rozkład ról wyglądał następująco:

- Aleksander Modzelewski – Lider
- Andżelina Piatrowicz – Projekt graficzny/mockupy
- Konrad Iwan – schemat bazy danych
- Szymon Werema – tester
- Piotr Lenarczyk – dokumentacja/diagramy

W semestrze przeznaczonym na implementację rozkład ról będzie wyglądał następująco

- Aleksander Modzelewski – Lider, backend engineer
- Andżelina Piatrowicz – tester
- Konrad Iwan – frontend engineer, mobile engineer
- Szymon Werema – tester
- Piotr Lenarczyk – devOps

<https://superglosnik.atlassian.net/jira/software/projects/UFG/boards/1/roadmap>

Kroki milowe i epiki

W semestrze przeznaczonym na planowanie wyróżniliśmy 3 kroki milowe: epik „Wybór stosu technologicznego” był milestonem [M1], epiki „Opracowanie dokumentacji i planowanie core functionalities” i „opracowanie dokumentacji i planowanie pozostałych functionalities” składały się na milestone [M2], a w ramach milestonu trzeciego był epik „stworzenie środowisk” [M3].

W ramach epika „Wybór stosu technologicznego” istniał jeden task o tej samej nazwie. W ramach epika „Opracowanie dokumentacji i planowanie core functionalities” każdy task dotyczył realizacji dokumentowania i planowania tych przypadków użycia, które dotyczyły core functionalities

(wszelkie diagramy, schemat bazy danych i mockupy). W ramach epiki „Opracowanie dokumentacji i planowanie pozostałych functionalities” każdy task dotyczył realizacji dokumentowania i planowania tych przypadków użycia, które dotyczyły funkcjonalności pozostałych (wszelkie diagramy, schemat bazy danych i mockupy).

Opis i motywacja wyboru danych technologii

- Node.js/express – Node.js to środowisko uruchomieniowe javascriptu świetnie nadające się do tworzenia aplikacji serwerowych, do backendu. Wraz z frameworkiem express.js tworzy doskonałe narzędzie do szybkiego i stosunkowo łatwego budowania skalowalnych i asynchronicznych aplikacji sieciowych, a razem z wbudowanymi w javascript mechanizmami wspomagającymi asynchroniczne programowanie (async/await, promesy) praca w nim jest płynna i intuicyjna.
- MongoDB – Jest to dokumentowa baza danych świetnie integrująca się z Node.js i javascriptem. Z uwagi na małą liczbę tabel jakie ma nasza aplikacja nie istnieje szczególne zapotrzebowanie na mechanizmy relacyjności, a zamiast tego na prostotę pracy i szybkość implementacji. MongoDB spełnia te wymagania.
- Figma – Figma to edytor grafiki wektorowej i narzędzie do tworzenia prototypów. Wybierając między Photoshopem a Figmą, zdecydowaliśmy się na Figmę dlatego że jest to wygodniejsze i łatwiejsze środowisko dla projektowania, tworzenia grafiki, prototypów. Łatwo udostępniać projekt klientom. Figma też jest szybka w działaniu.
- React.js – Do stworzenia frontendu aplikacji webowej użyjemy javascriptowej biblioteki React.js. Jest to biblioteka javascriptu służąca do tworzenia graficznych interfejsów użytkownika aplikacji webowych. Cechuje się ona dobrą wydajnością, skalowalnością oraz wysokim poziomem bezpieczeństwa. Ze względu na swoją popularność, istnieje wiele bibliotek współgrających z Reactem, na przykład Redux, którego wykorzystamy do zarządzania stanem aplikacji czy biblioteka Styled-Components która pozwoli na łatwe i szybkie stylowanie komponentów aplikacji.
- Flutter – Do stworzenia aplikacji mobilnej wykorzystamy zestaw narzędzi Flutter. Pozwala on na tworzenie aplikacji mobilnych na urządzenia z systemem Android oraz iOS z wykorzystaniem jednego kodu napisanego w języku Dart. Spośród wszystkich rozwiązań tego typu, Flutter okazuje się być najlepszy pod względem wydajności, bliski natywnym aplikacjom.

Dokumentacja techniczna

Schemat bazy danych

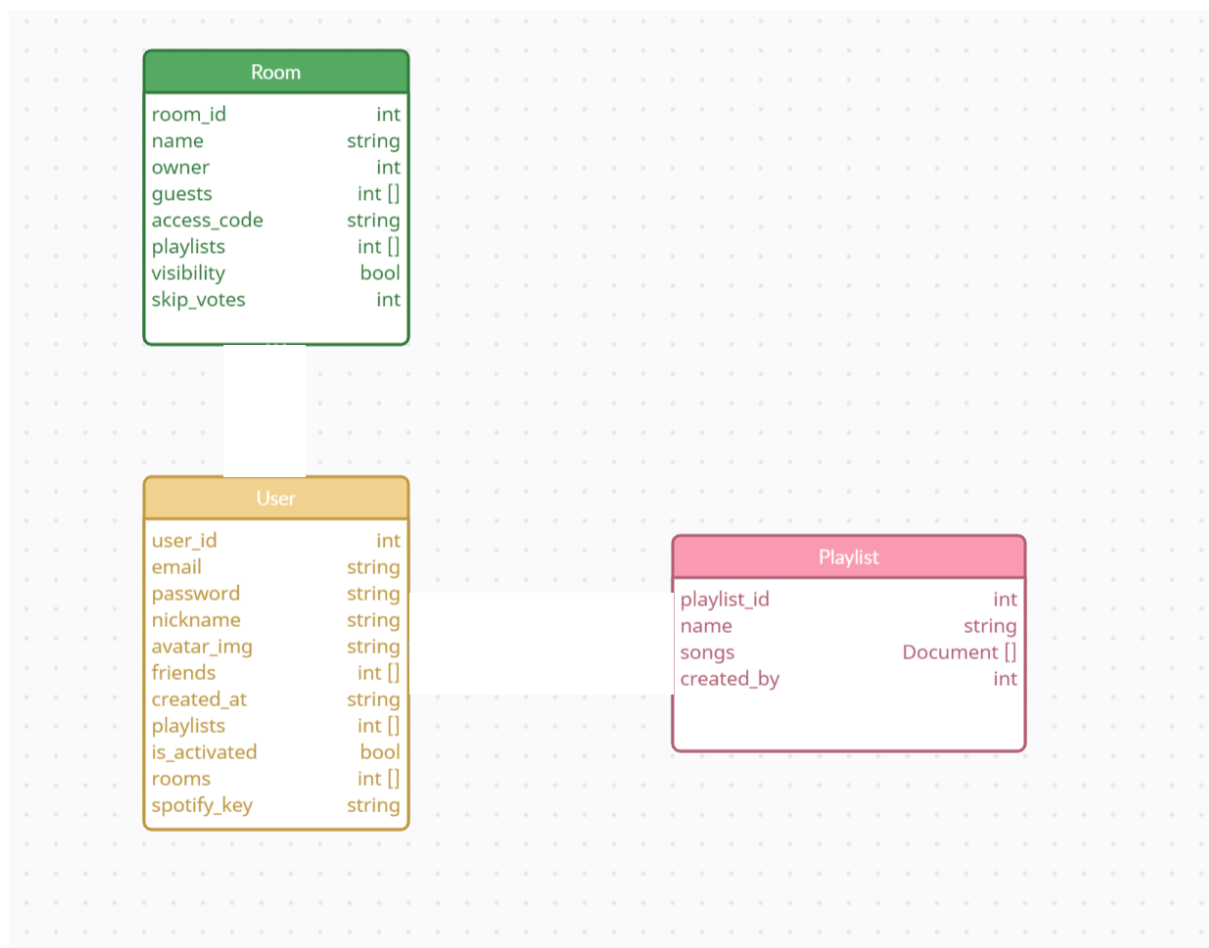
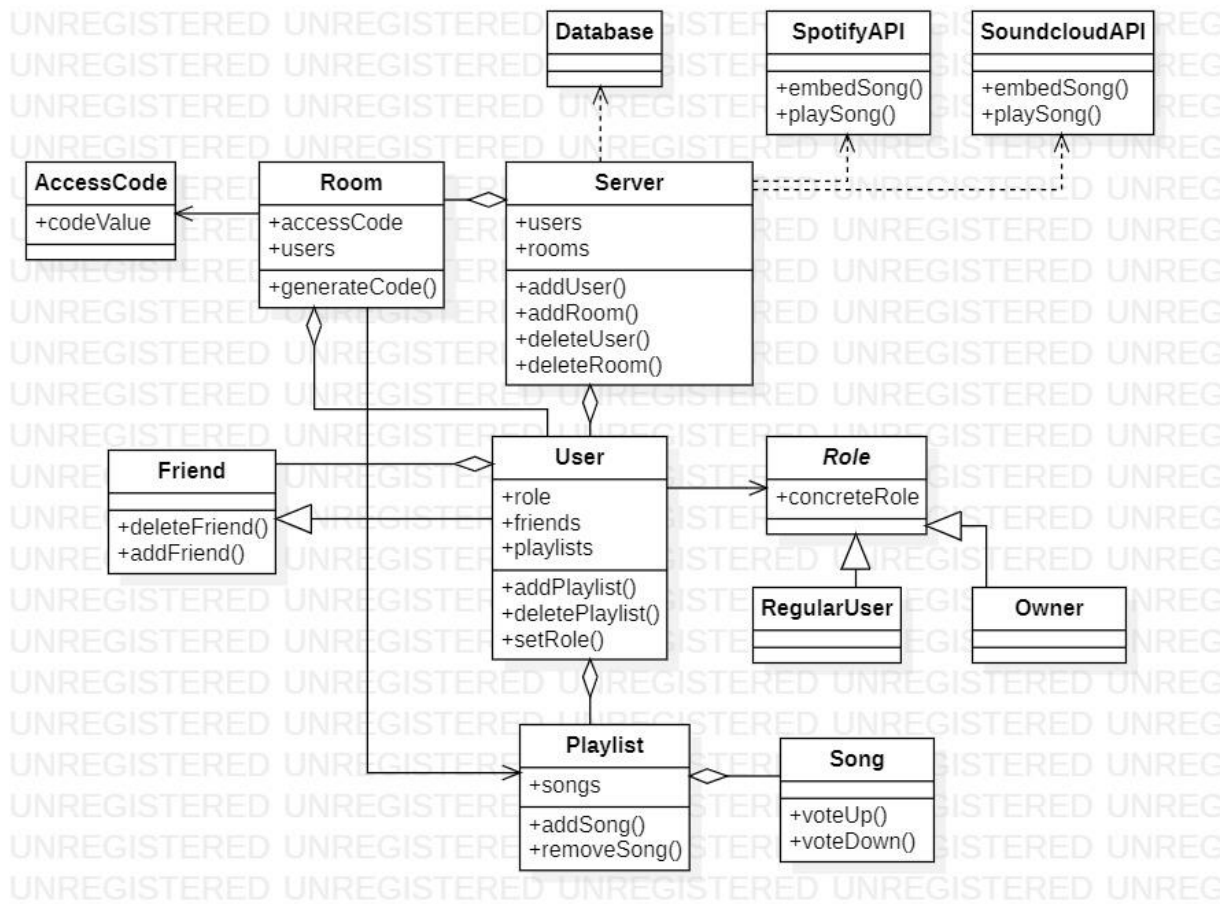
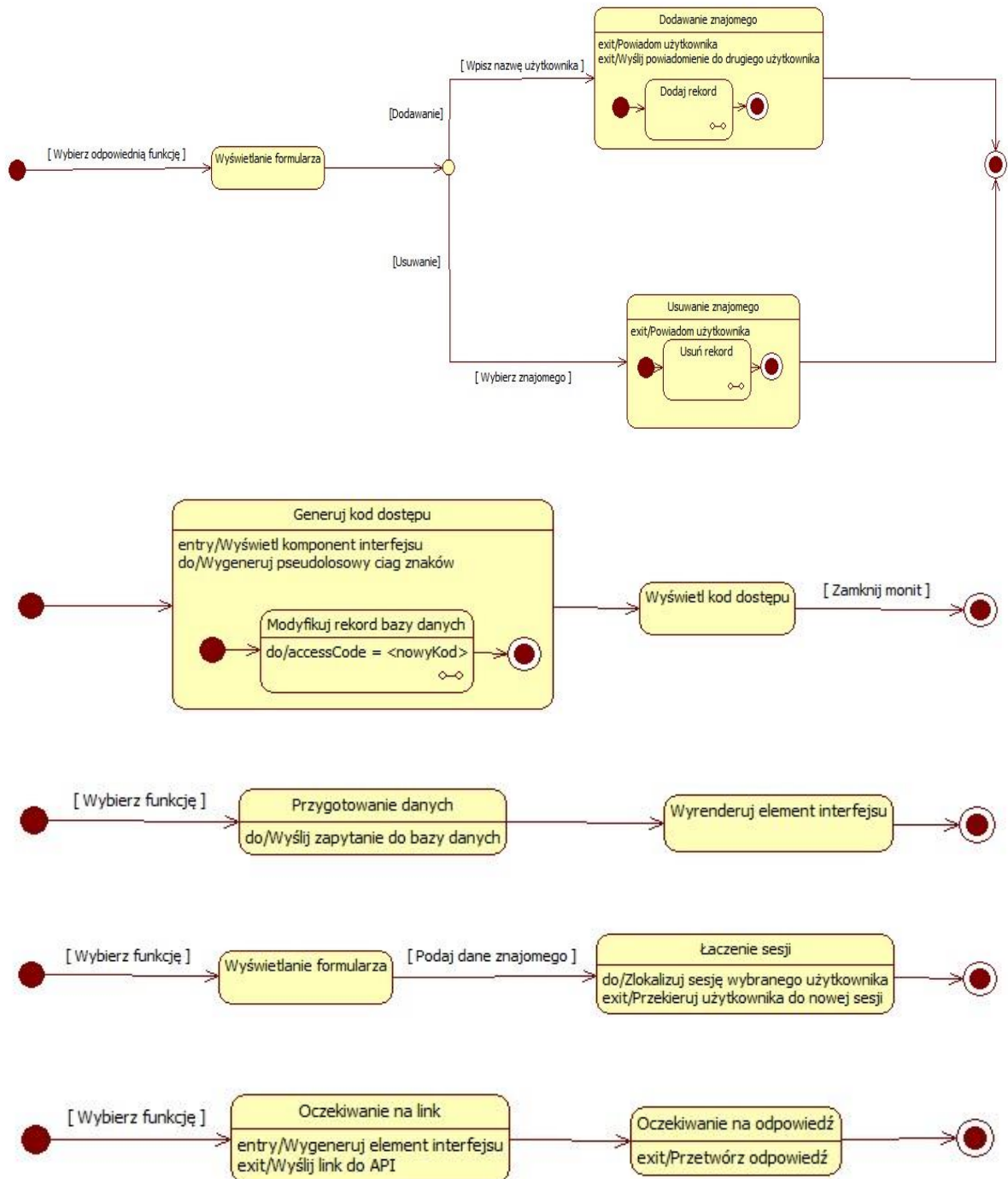
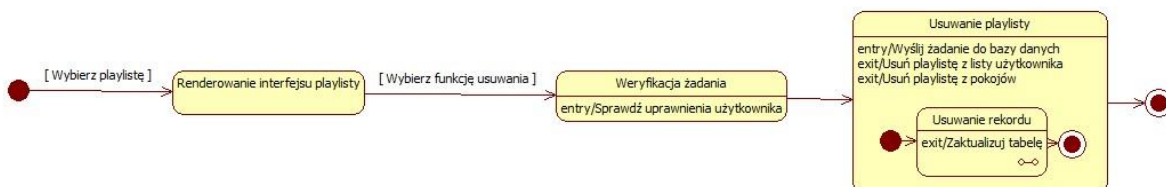
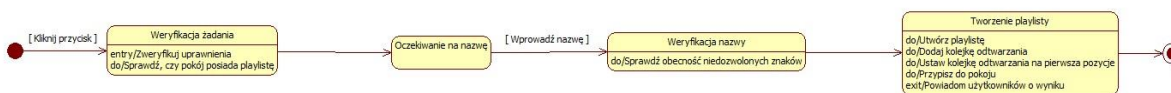
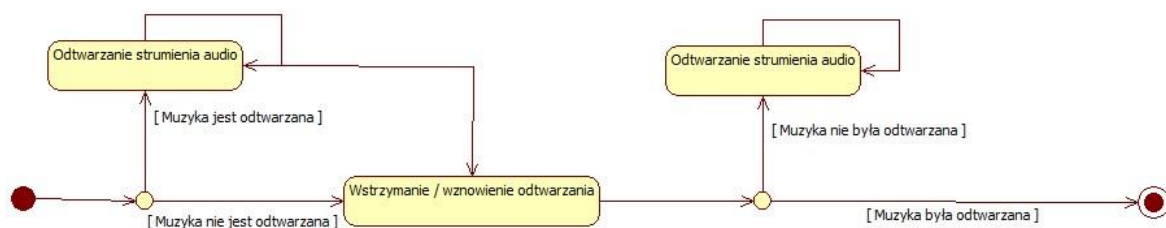
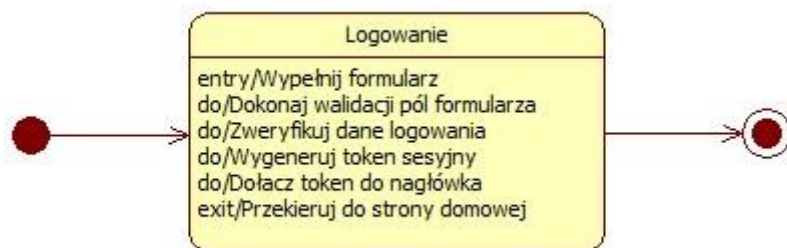


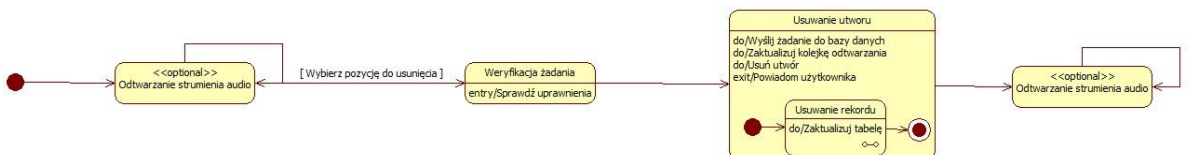
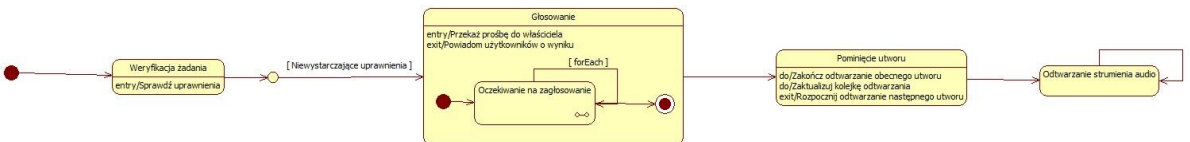
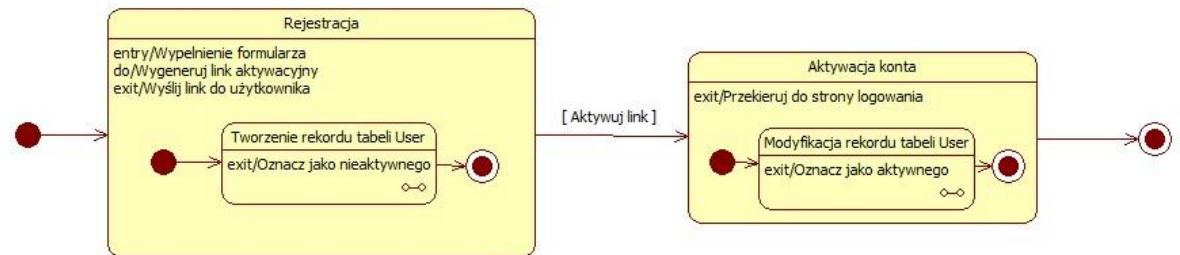
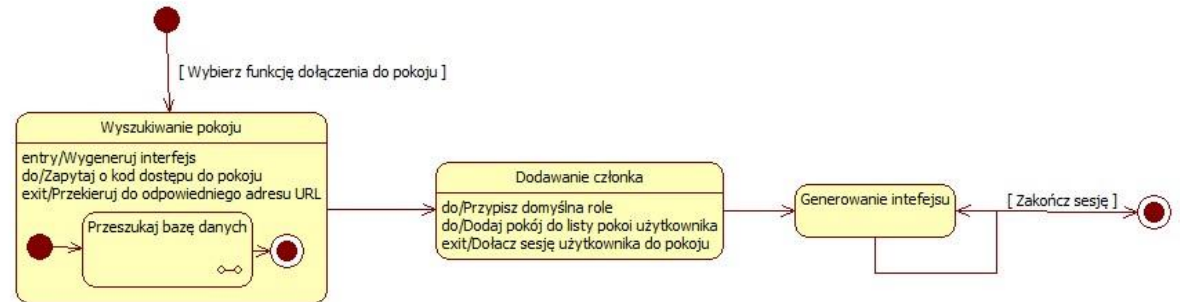
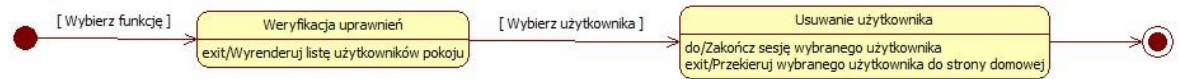
Diagram klas

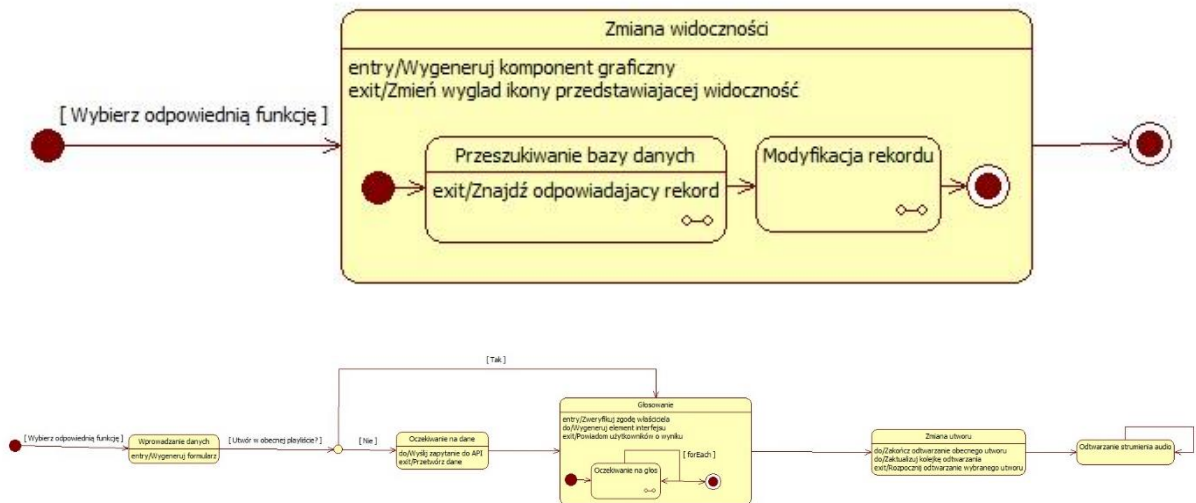


Diagramy stanów

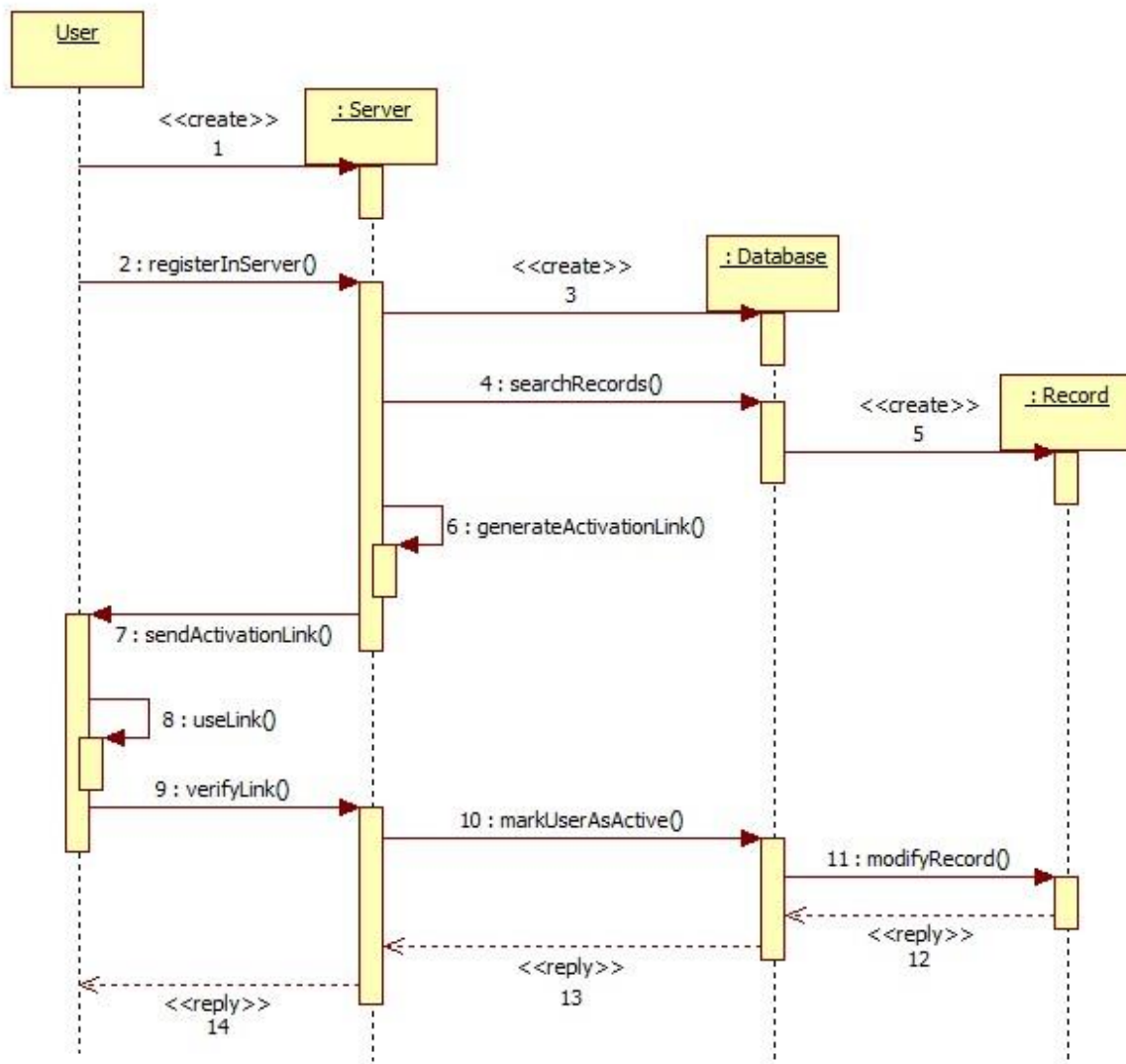


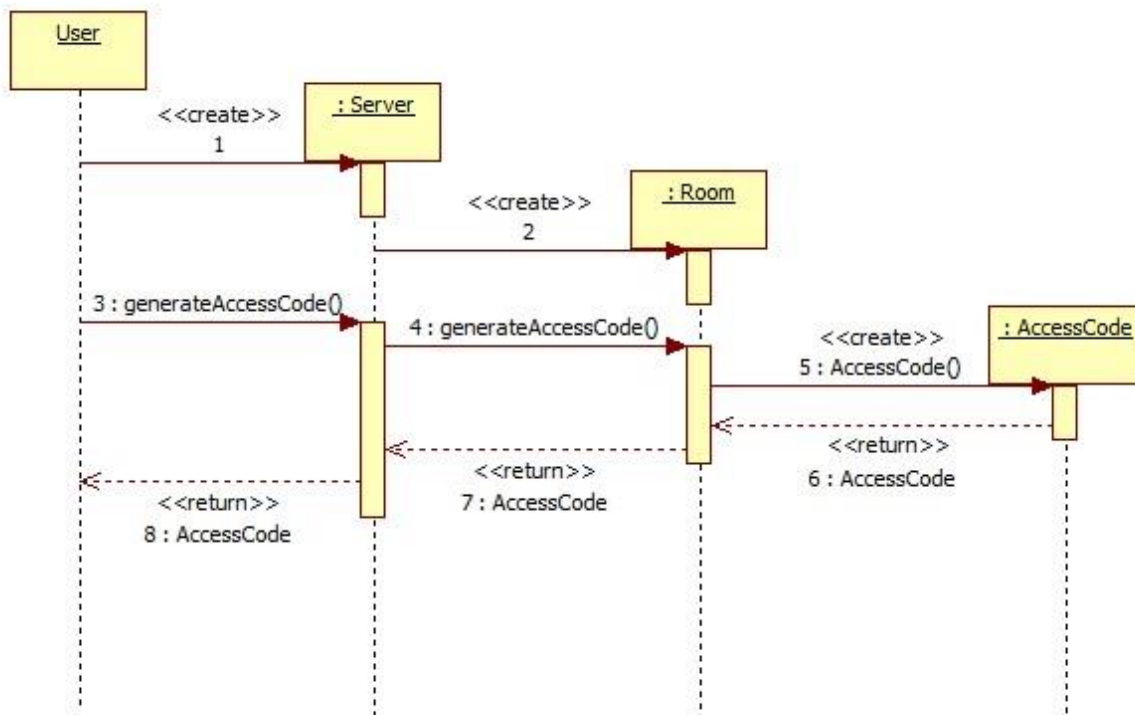
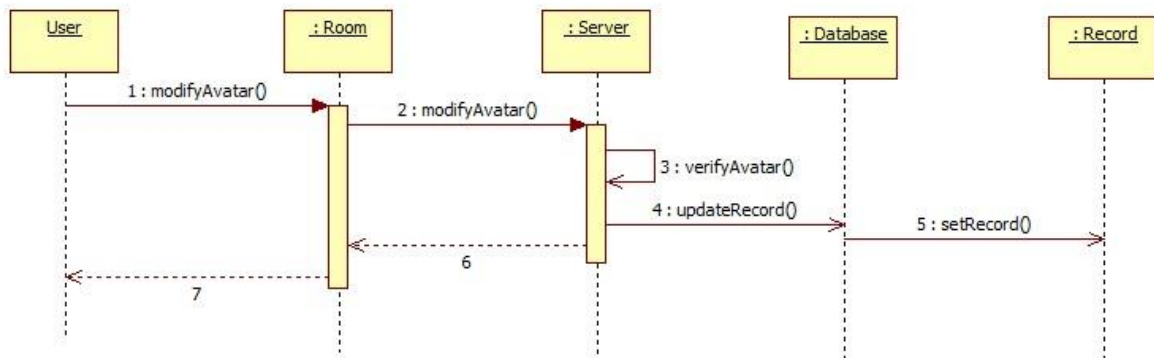
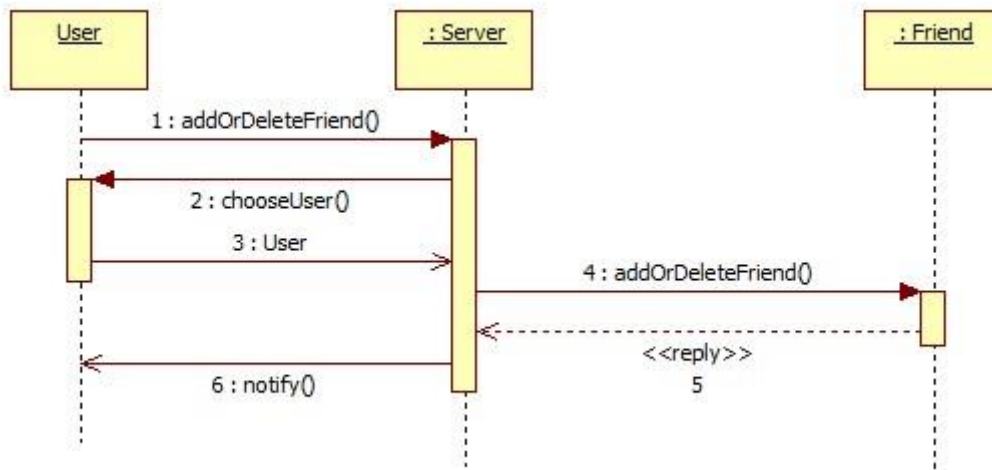


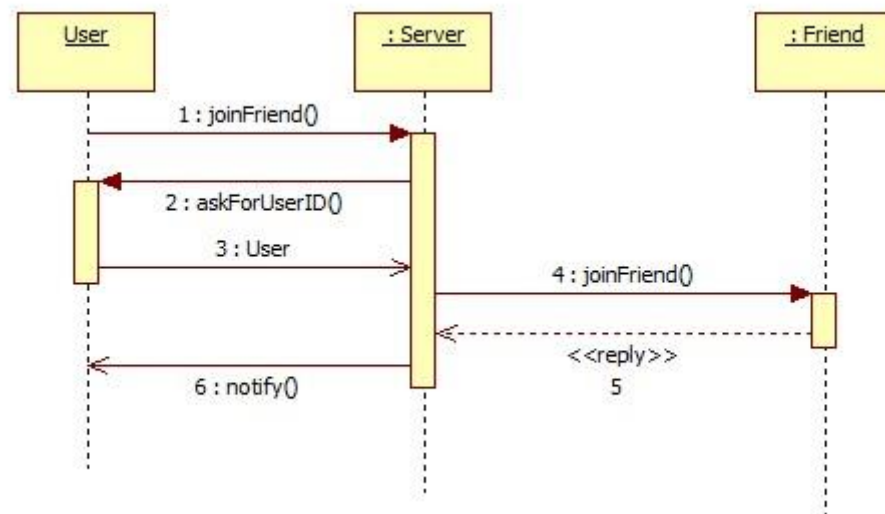
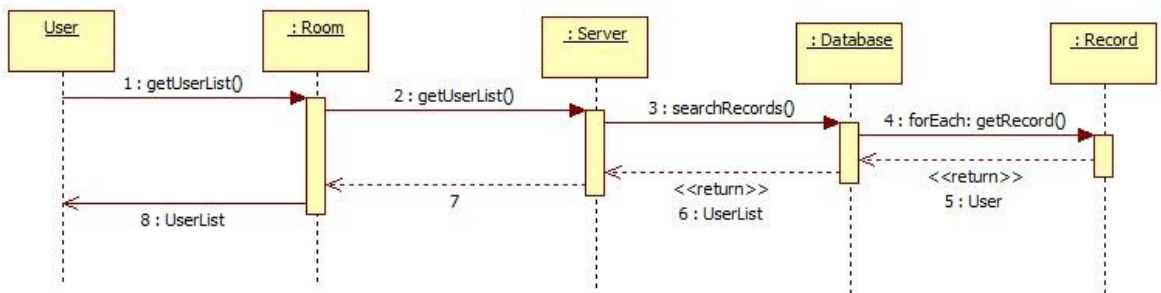
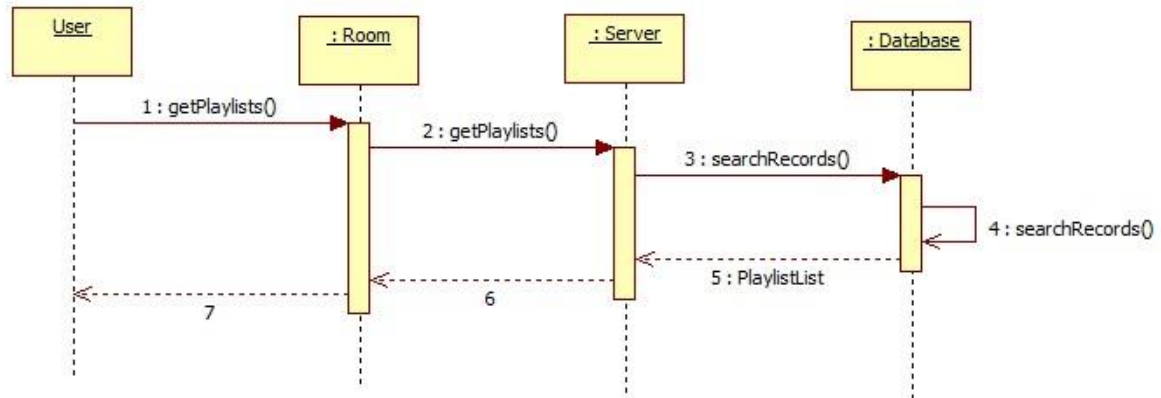


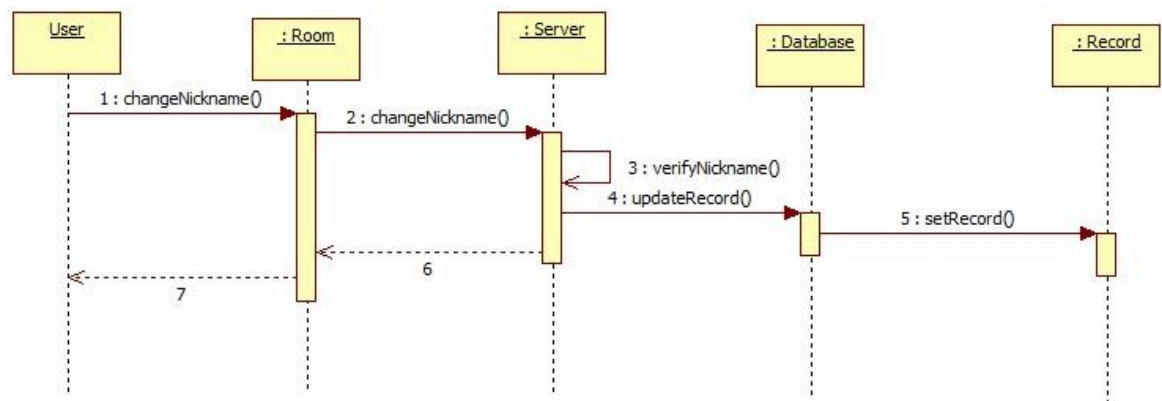
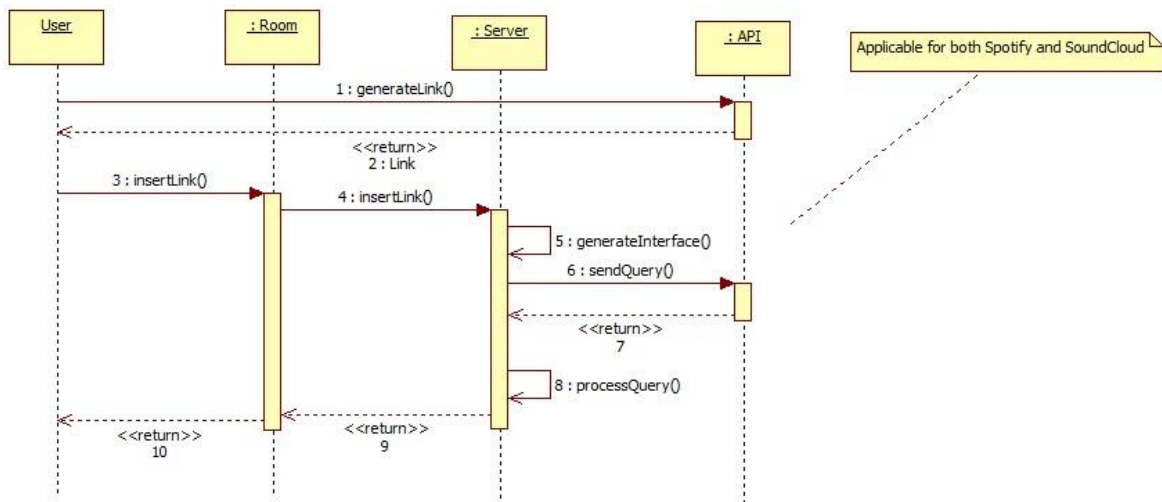
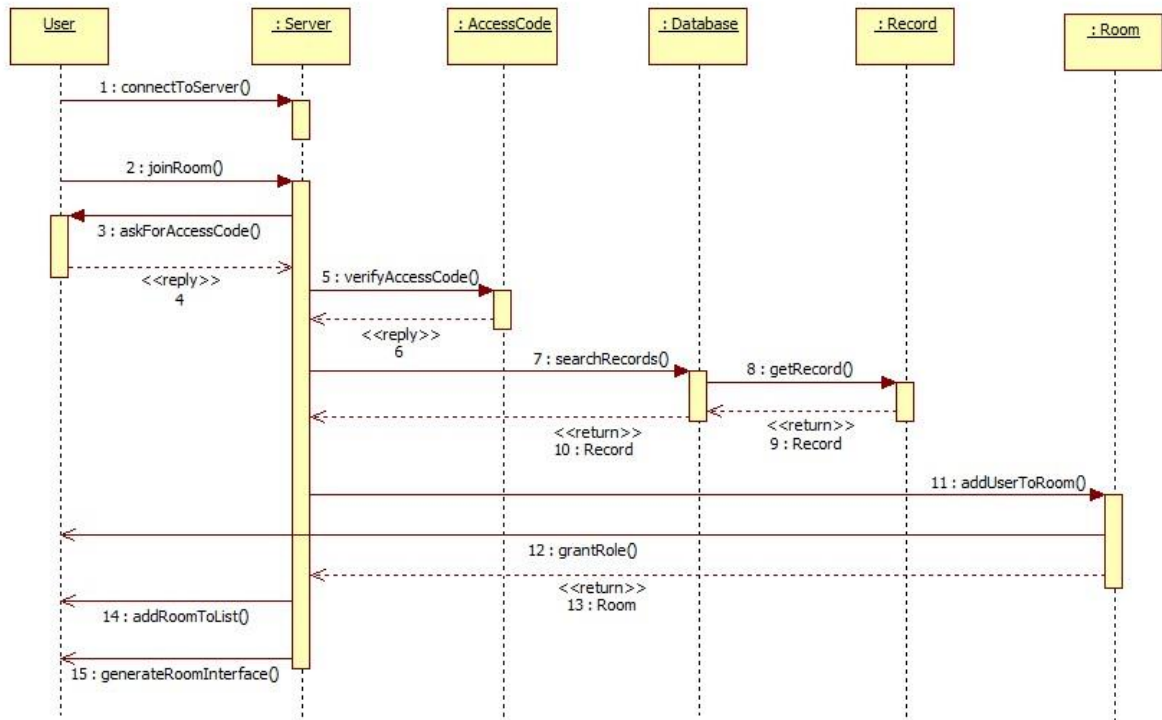


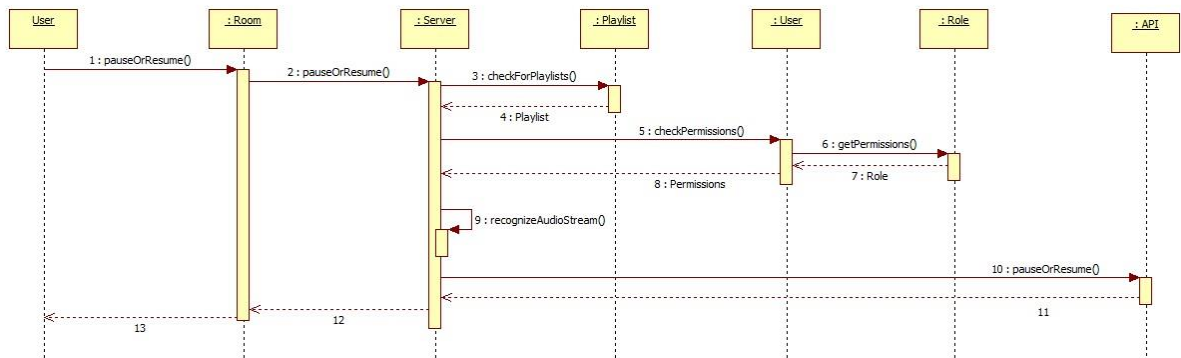
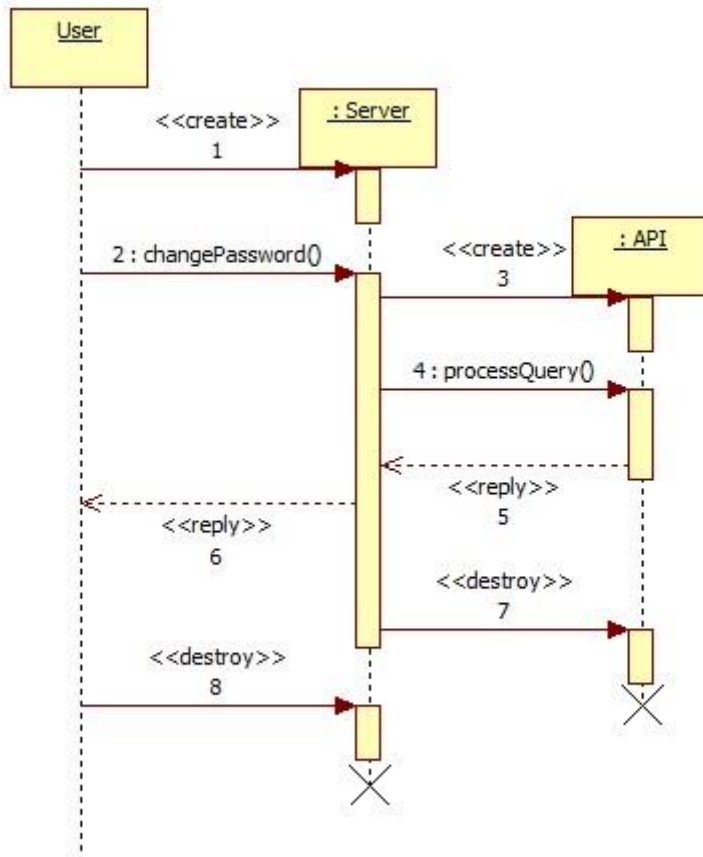
Diagramy sekwencji



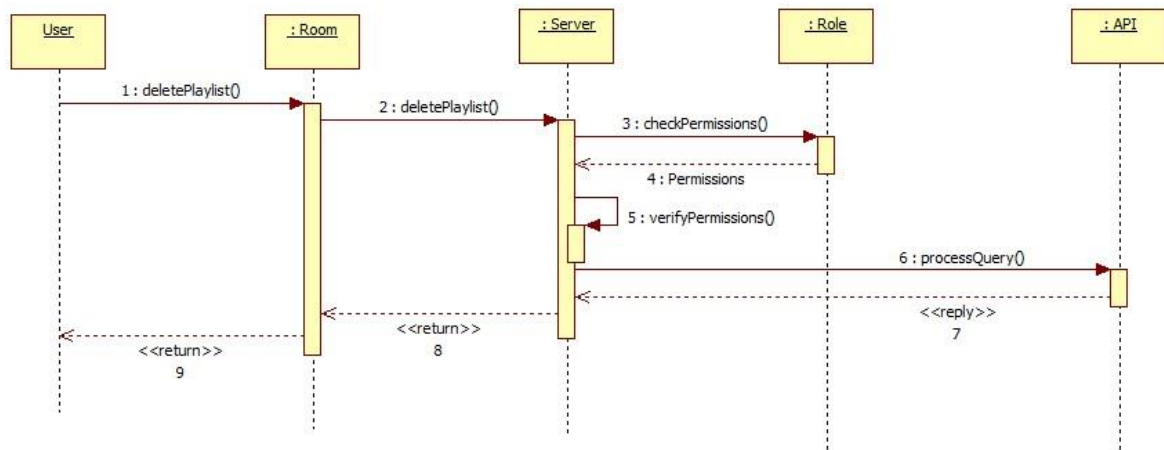
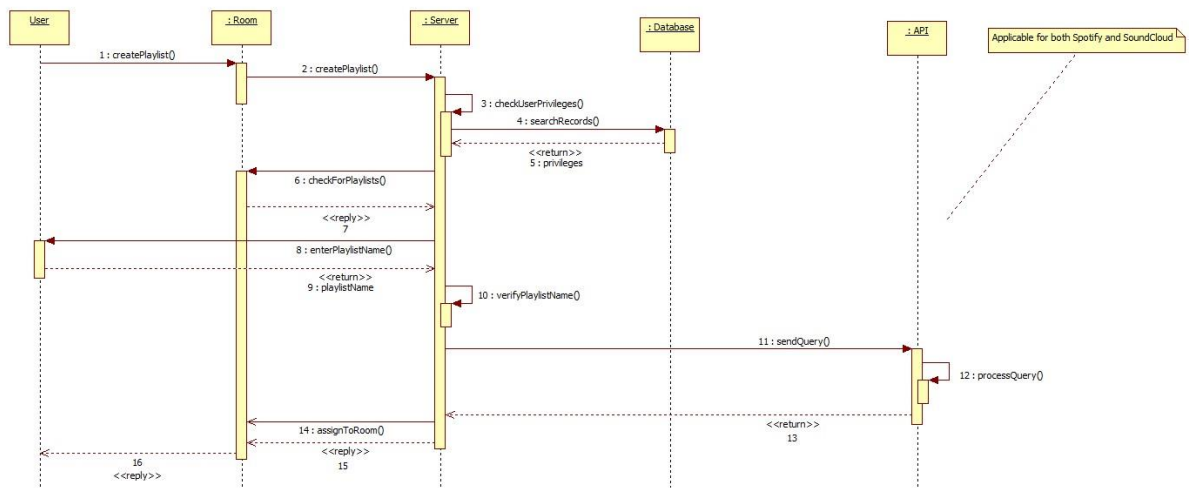
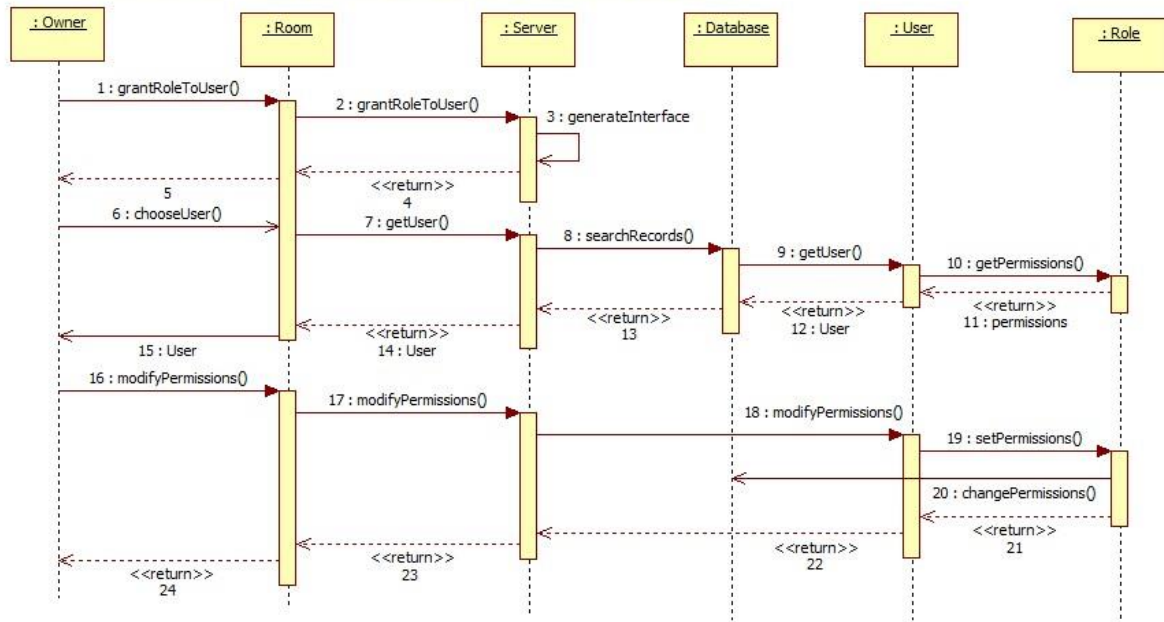


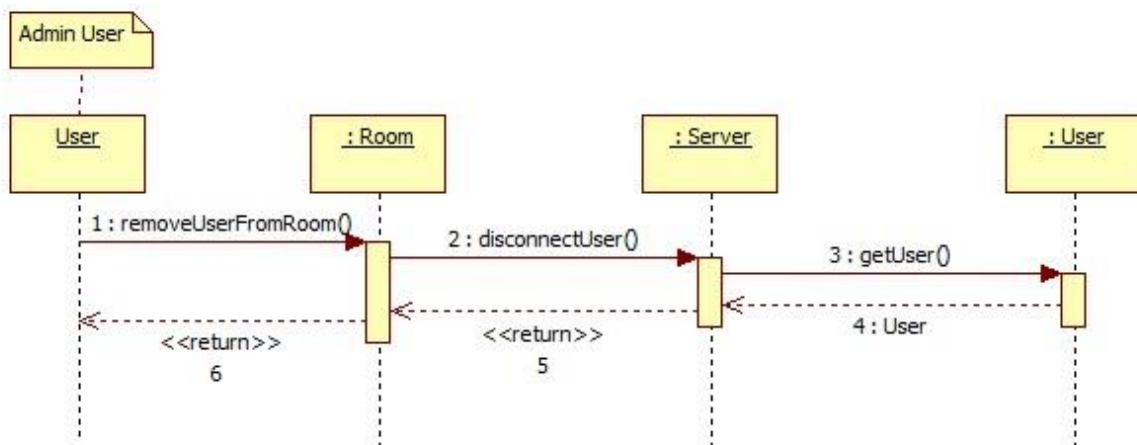
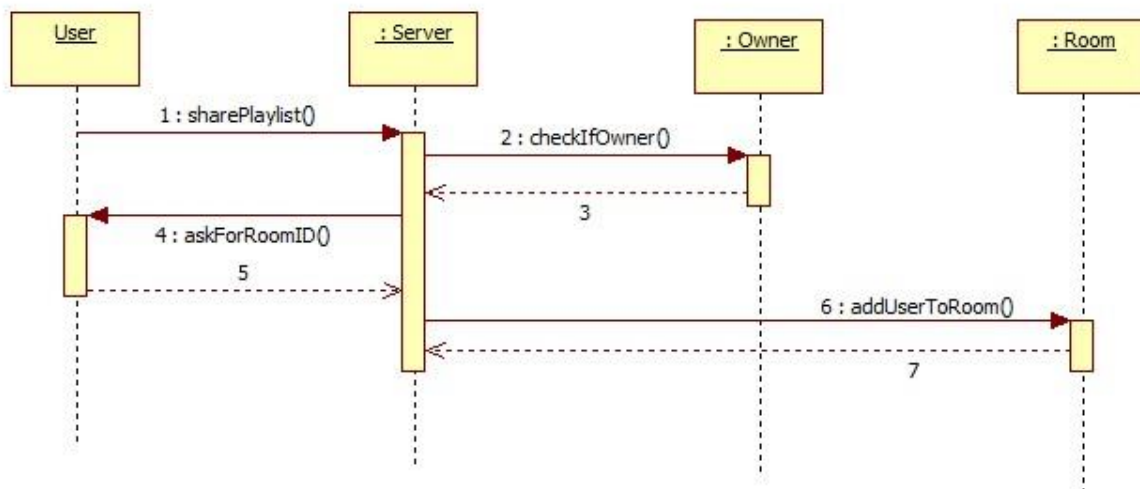
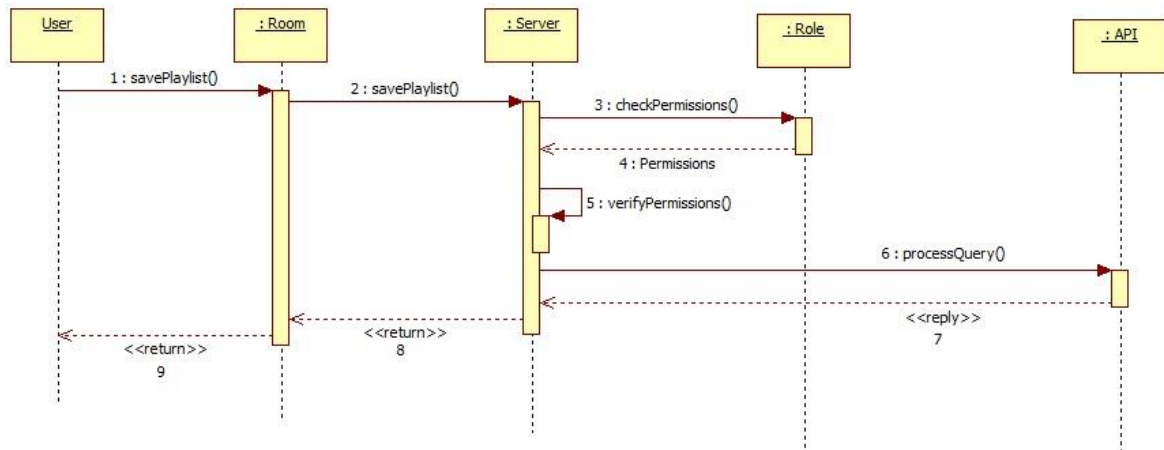


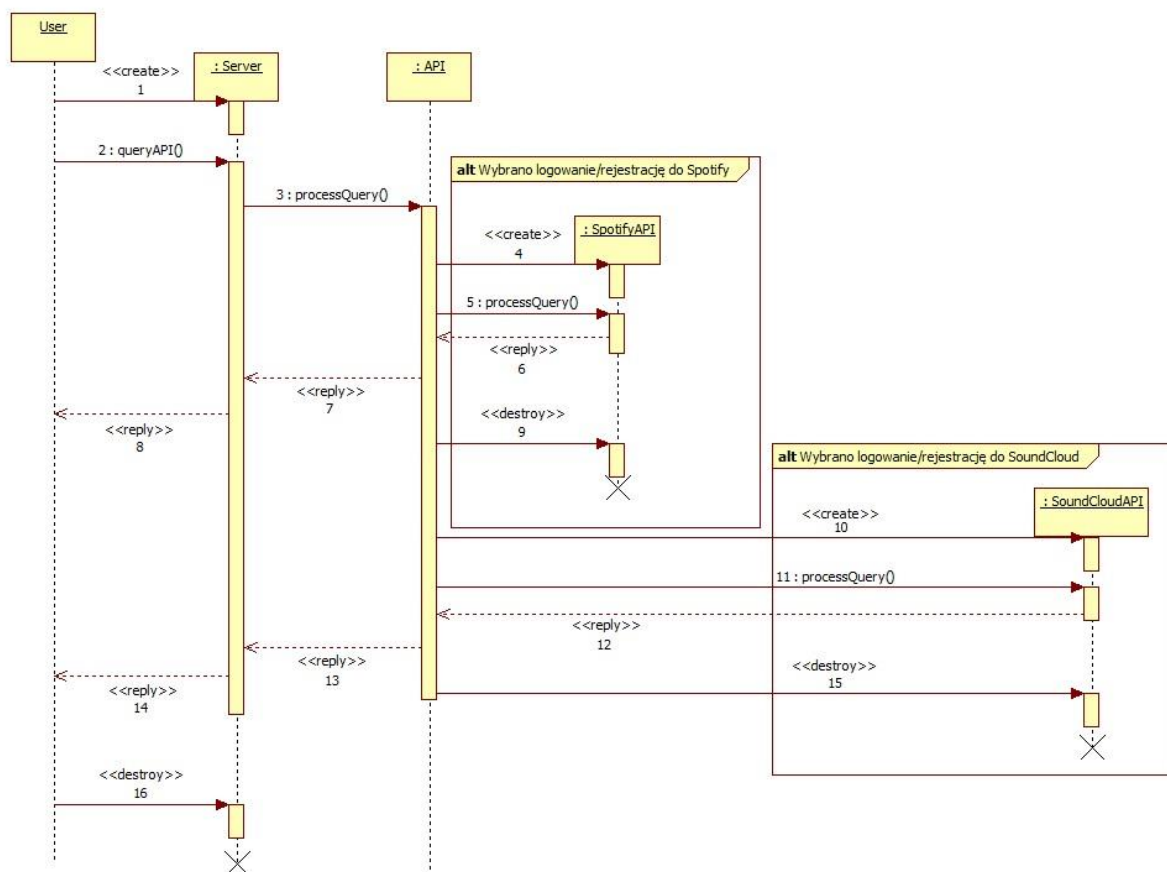
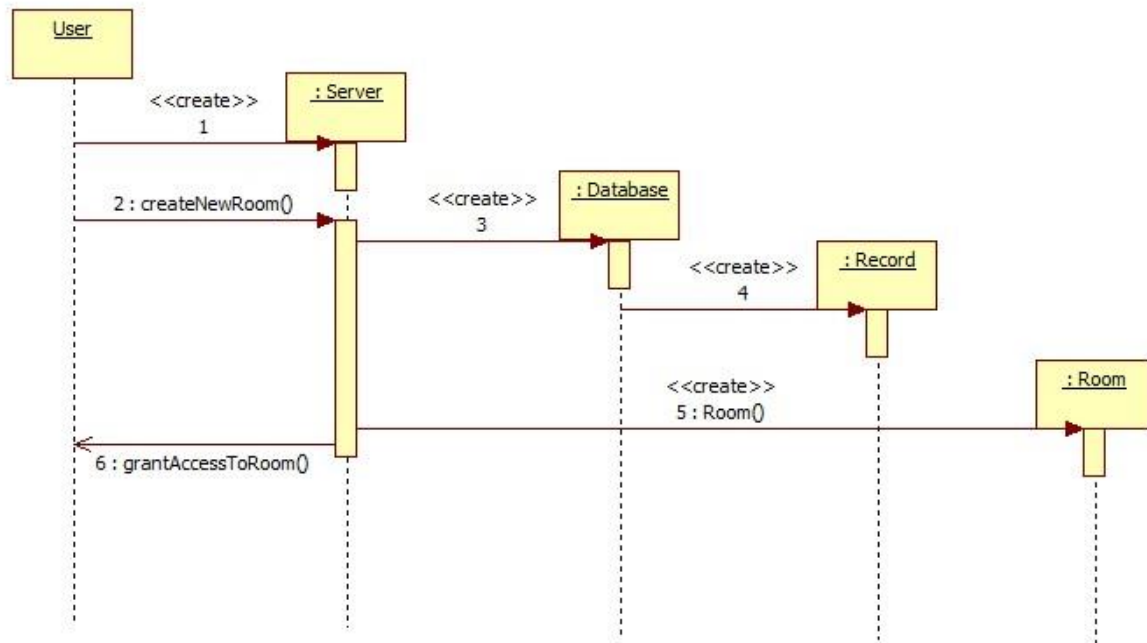


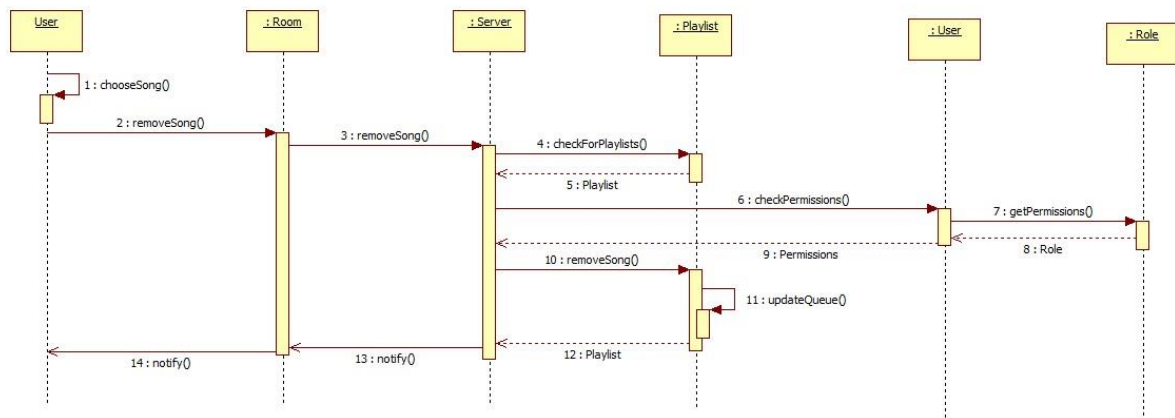
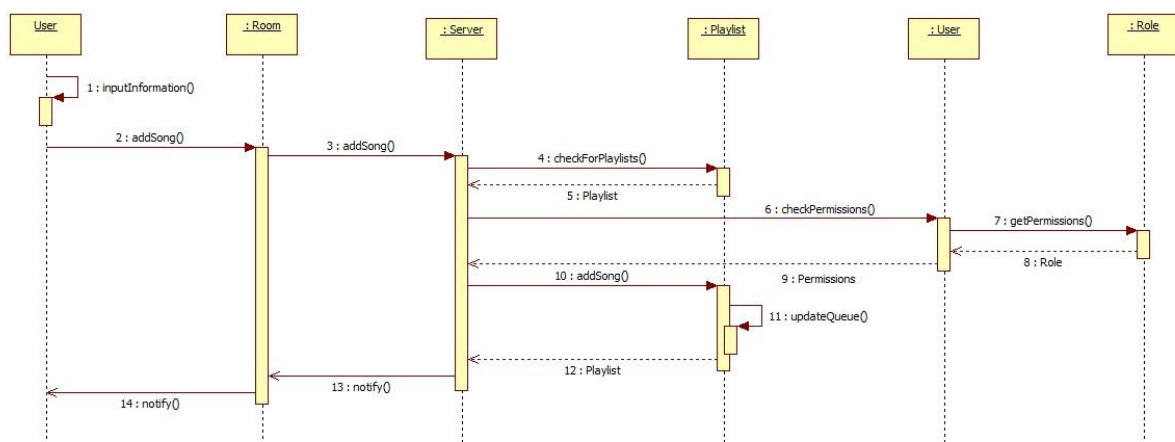
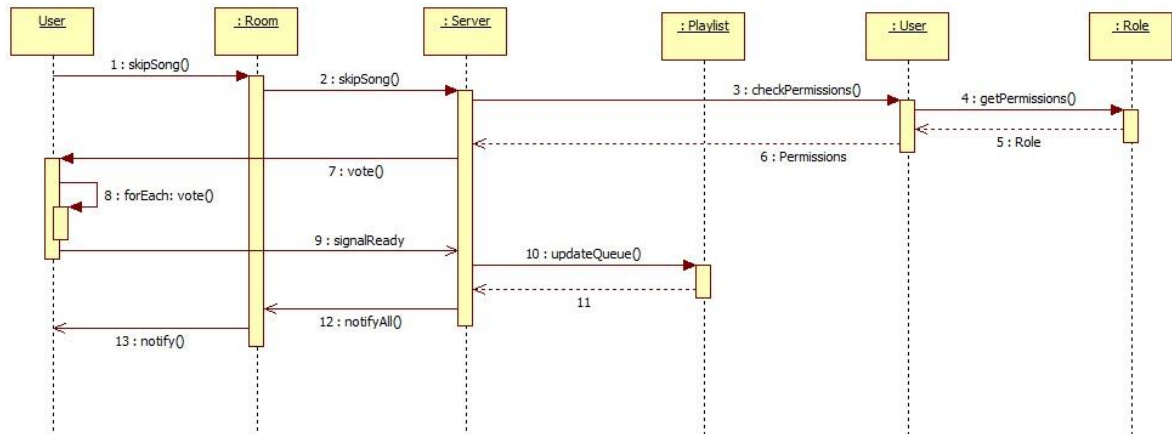


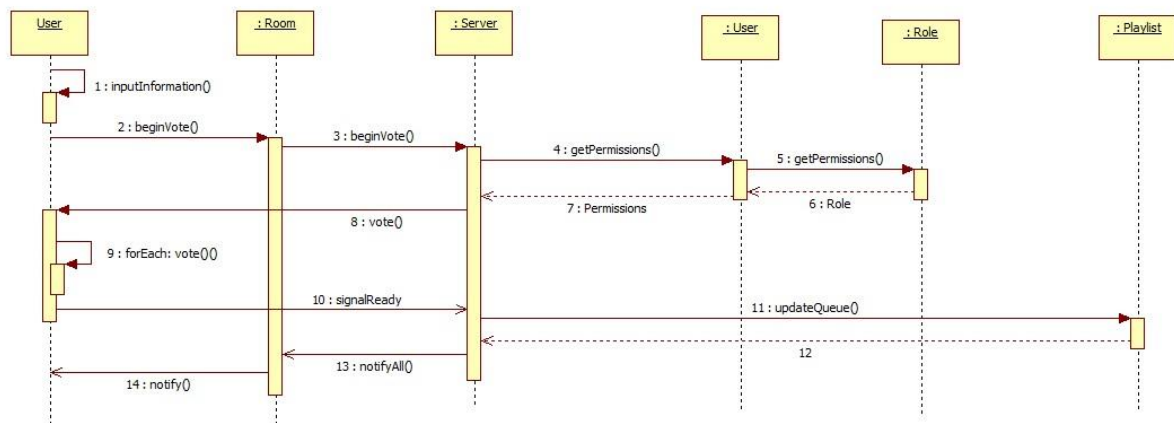
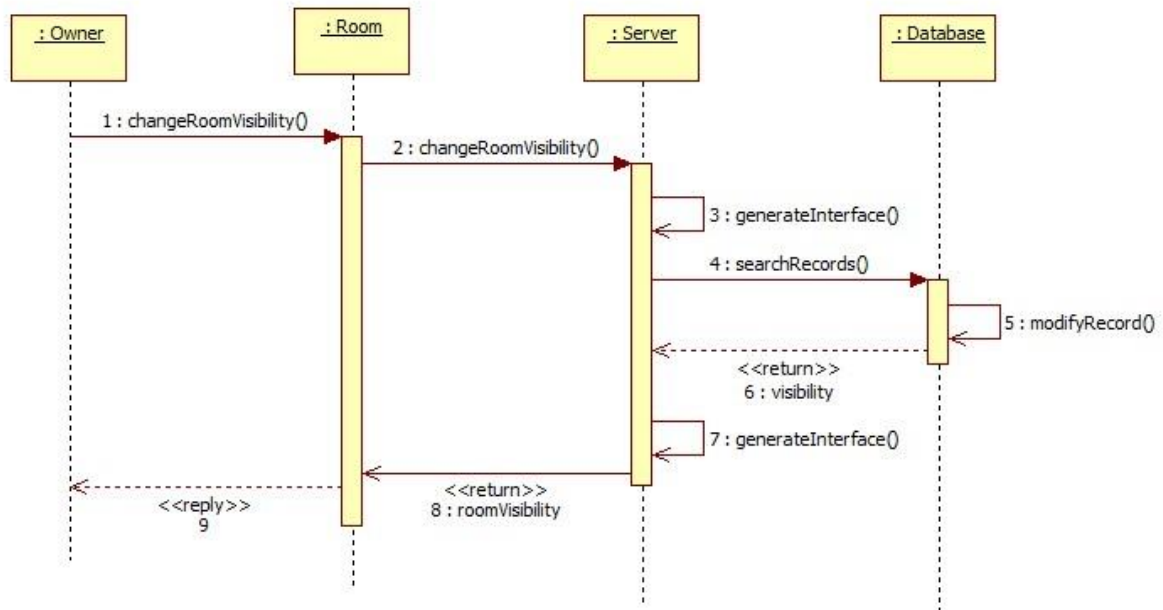
Initial conditions:
 Owner is already connected to the server as well as the room he owns
 User whose permission are being modified is already connected to the server and the room of the Owner











Scenariusze testowe

Scenariusze testowe są dostępne na bitbucket w sekcji test

<https://bitbucket.org/ToTenOn/ultra-fajny-g-o-nik/src/master/>

GUI

Link do Figma z przedstawieniem workflow interfejsu graficznego:

<https://www.figma.com/proto/7o5igVOu5hZz4wQXF9e138/app?node-id=202%3A1737&scaling=scale-down&page-id=0%3A1&starting-point-node-id=202%3A1737>

API

API jest dostępne na bitbucket w formacie JSON

https://bitbucket.org/ToTenOn/ultra-fajny-g-o-nik/src/master/api/yourplaylist.postman_collection.json

Zasoby

Do realizacji projektu potrzebna będzie odpowiednia domena sieciowa, dostęp do API Spotify i Soundcloud oraz darmowy hosting na serwisie Heroku. Jako, że baza danych naszego projektu nie jest szczególnie rozbudowana, a przechowywane dane są w większości plikami tekstowymi, a więc zajmują mało miejsca, a na dodatek core naszej aplikacji nie jest wymagające obliczeniowo w zupełności(przynajmniej w początkowym okresie istnienia aplikacji) wystarczy rozwiązanie darmowe jakie oferuje Heroku.

Środowiska

Użycie konkretnych środowisk(DEV, Test, PROD) zależy od stadium workflow, w którym znajduje się projekt. Opierając się na workflow opisanym w jira naszego projektu:

- DEV: To Do, In Progress
- TEST: Ready for test, In testing, Ready for done
- PROD: DONE