1 BLa bla bla - wymagane punkty jak z deliverable-one (MODULARNOSC!!!)

2 Architektura systemu

System z założenia może skalować w dwóch wymiarach, to znaczy:

1. zwiększanie liczby serwerów wirtualnych - liczba sesji dla użytkowników
2. zwiększenie liczby nadzorców - liczba obsługiwanych klientów jednocześnie

3 Opis tworzonych modułów

3.1 Nadzorca

3.2 Serwer wirtualizacji

3.3 Aplikacja kliencka

3.4 Panel administratora

4 Opis zewnętrznie dostarczonych modułów

4.1 Broker wiadomości - RabbitMQ

4.2 Dysk sieciowy

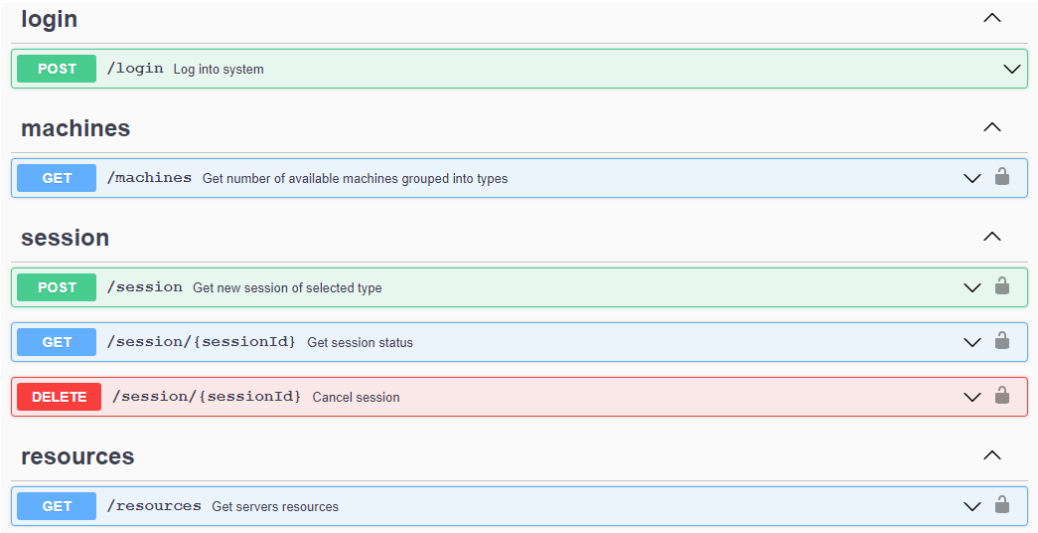
4.3 System katalogowy

5 Komunikacja

5.1 Komunikacja użytkownika z systemem - REST API

Komunikacja aplikacji klienckiej oraz panelu administratora z systemem - nadzorcą - rozwiązana jest za pomocą REST API[1]. Wiadomości wysyłane są za pomocą protokołu HTTPS[2], który zapewnia ich szyfrowanie. W tym celu wymagane jest, aby na adres, pod którym udostępniony będzie system, wystawiony był odpowiedni certyfikat[3], gwarantujący jego tożsamość. Podczas tworzenia systemu i testów możliwe jest użycie sztucznego, własnoręcznie podpisanego certyfikatu[4].

Całość specyfikacji API umieszczona jest w osobnym pliku. Poniżej znajduje się zestawienie oraz krótki opis endpointów.



Rysunek 1: Endpointy API

5.2 Komunikacja wewnątrz systemu - broker wiadomości

[1]Opis REST API
[2]Specyfikacja protokołu HTTP Over TLS
[3]Opis certyfikatu TLS/SSL
[4]Opis własnoręcznie podpisanego certyfikatu TLS/SSL

6 Diagramy

6.1 Diagramy stanów

6.2 Diagramy aktywności

6.3 Diagramy klas

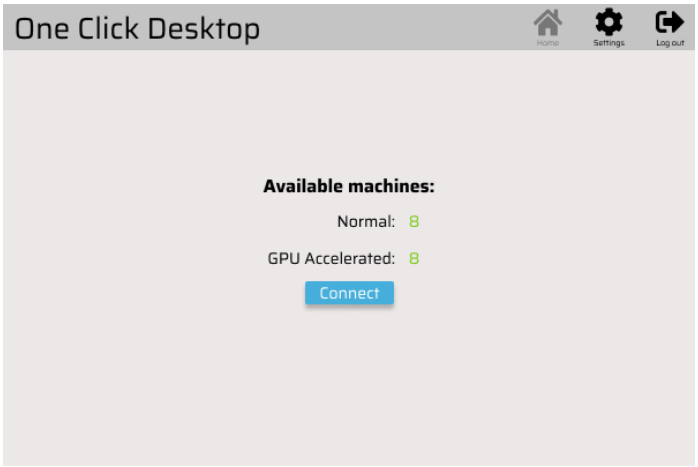
6.4 Diagramy sekwencji

7 Interfejs użytkownika

7.1 Aplikacja kliencka

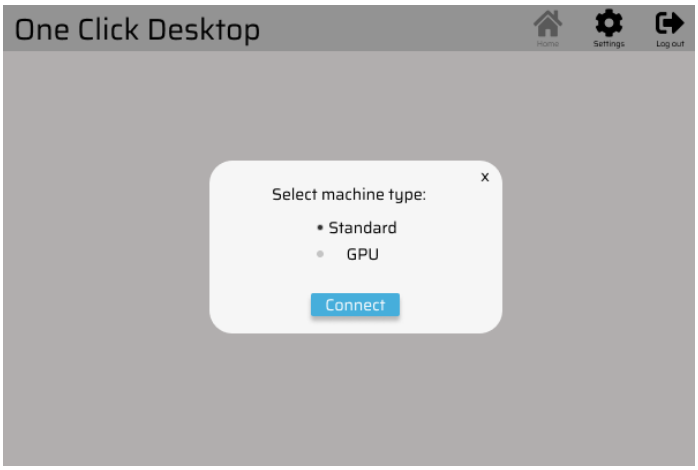
Aplikacja kliencka posiada interfejs użytkownika pozwalający na zalogowanie się oraz nawiązanie połączenia ze zdalną sesją. Użytkownikowi wyświetlany jest czynność, która aktualnie się odbywa, oraz w każdym momencie może zakończyć sesję.

Rysunek 2: Ekran logowania



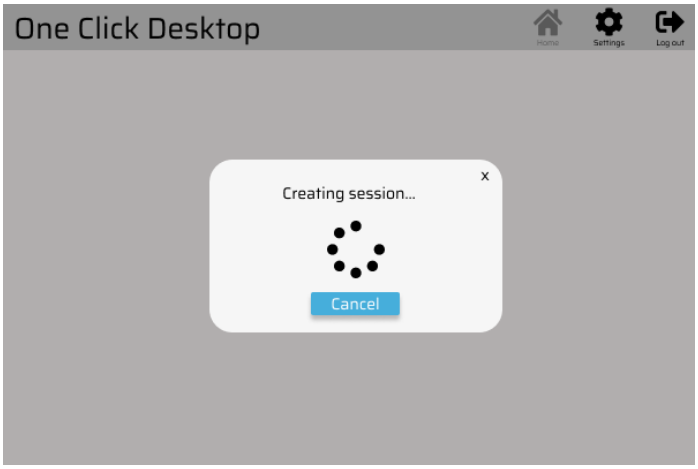
Rysunek 3: Główny widok zawierający dostępność maszyn

Na tym ekranie użytkownik może rozpocząć proces uzyskiwania sesji, przejść do ekranu ustawień lub wylogować się. Jeżeli w systemie nie ma dostępnych maszyn, lub nie uda się uzyskać informacji o ich dostępności, to przycisk połączenia jest niedostępny. Wciśnięcie tego przycisku prowadzi do kolejnego ekranu.

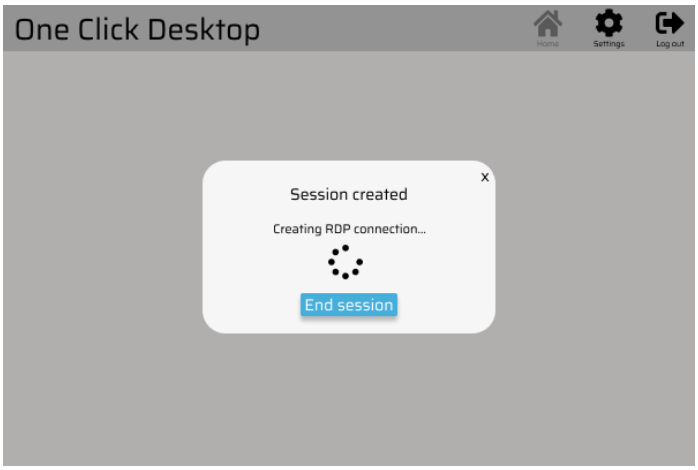


Rysunek 4: Wybór typu sesji

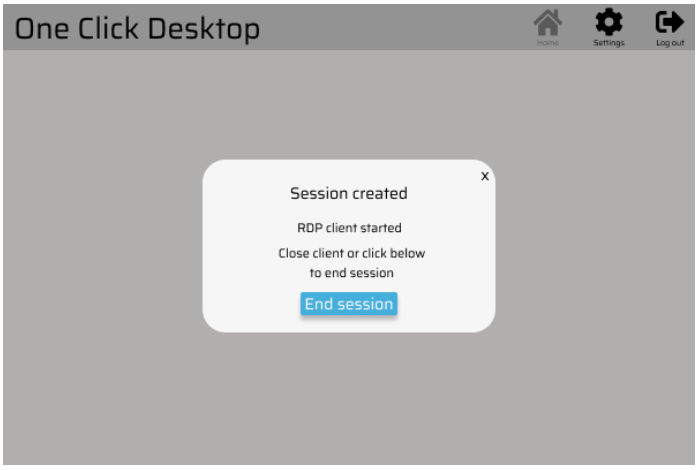
Do wyboru dostępne są jedynie typy sesji, które system określi jako dostępne. Wybranie typu sesji kliknięcie przycisku prowadzi do kolejnego ekranu. Następne ekrany przechodzą automatycznie do kolejnych bez interwencji użytkownika, aż do informacji o nawiązaniu połączenia lub błędu.



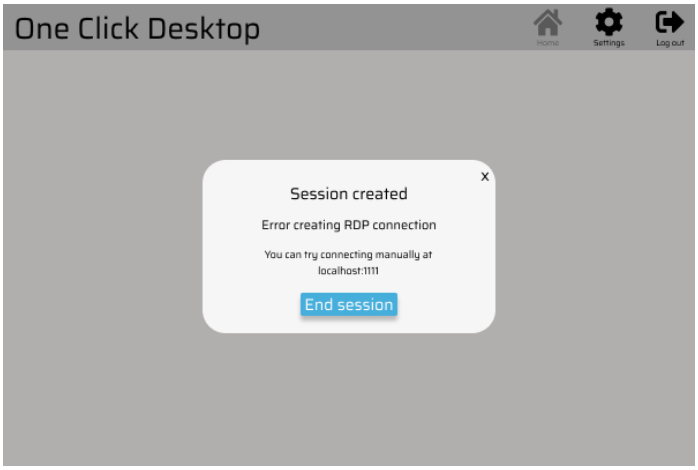
Rysunek 5: Tworzenie sesji



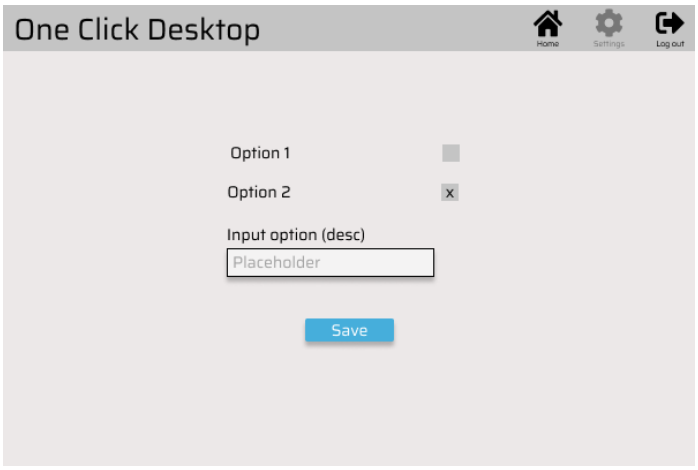
Rysunek 6: Nawiązywanie połączenia RDP



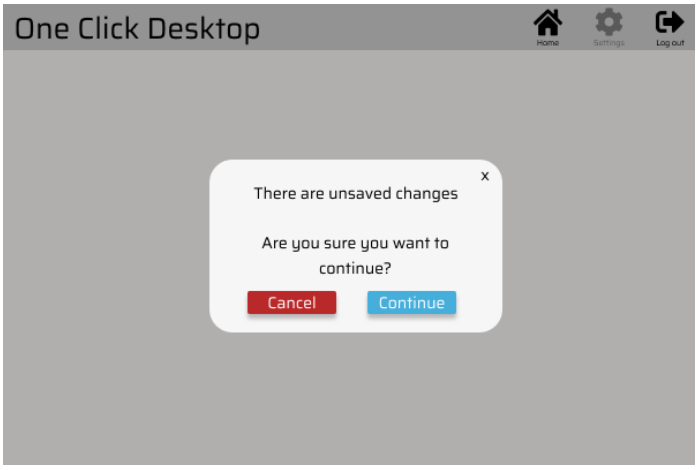
Rysunek 7: Połączenie nawiązane



Rysunek 8: Błąd przy nawiązywaniu połączenia



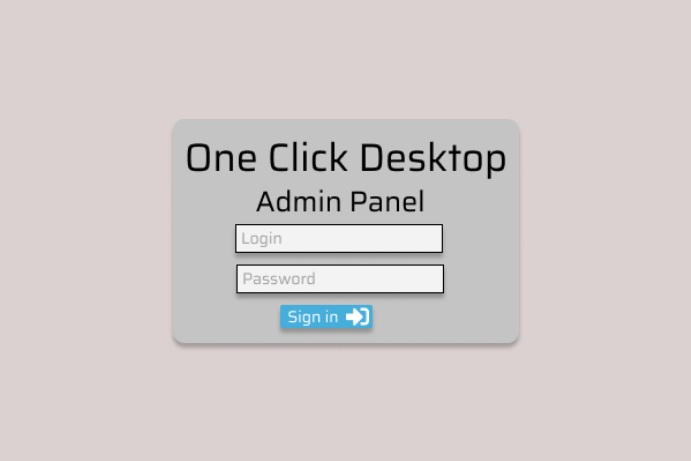
Rysunek 9: Ustawienia



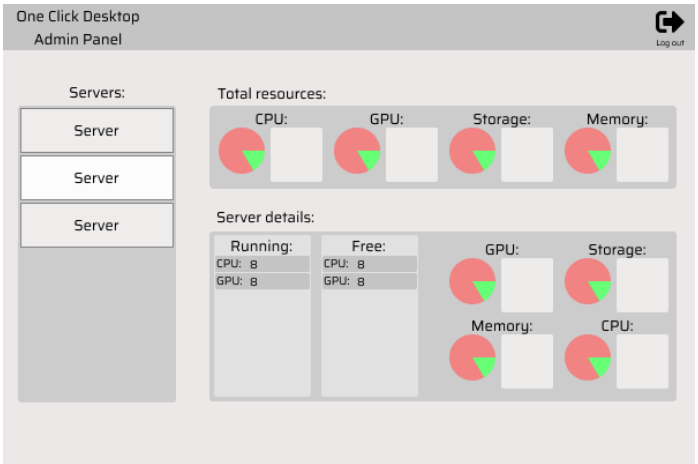
Rysunek 10: Powiadomienie przy wyjściu z ekranu ustawień z niezapisanymi zmianami

7.2 Panel administratora

Panel administratora posiada skromny interfejs umożliwiający zalogowanie się oraz podgląd zużycia zasobów.



Rysunek 11: Ekran logowania



Rysunek 12: Widok zużycia zasobów

Na tym widoku możemy zobaczyć zużycie zasobów globalne oraz dla każdego z serwerów wirtualizacji. Dodatkowo dla serwera wyświetlona jest również ilość działających i możliwych do uruchomienia maszyn każdego z typów.

8 Zewnętrzne narzędzia

8.1 Ansible

8.2 Vagrant

8.3 libvirt

9 Wybrana technologia

- Aplikacja kliencka
  - Typescript[5] /Javascript[6]
  - Node.js[7] - środowisko uruchomieniowe używane do integracji z systemem użytkownika
  - Angular[8] - renderowanie widoków
  - Electron[9] - platforma programistyczna
  - Jest[10] - testy jednostkowe
  - Cypress[11] - testy integracyjne
  - GNU/Linux oraz Windows - wspierane systemy operacyjne
- Panel administratora
  - Typescript/Javascript
  - Angular - platforma aplikacji WWW
  - Jest - testy jednostkowe
  - Cypress - testy integracyjne
- Nadzorca i serwer wirtualizacji
  - C#[12]
  - RabbitMQ[13] - broker asynchronicznych wiadomości
  - Ansible[14] - zarządzanie maszynami wirtualnymi
  - Vagrant[15] - tworzenie obrazów maszyn wirtualnych
  - libvirt[16] - uruchamianie maszyn wirtualnych
  - OpenLDAP[17] - dostępu do systemu katalogowego
  - NFS[18] - dostęp do katalogów domowych z maszyny wirtualnej

[5]Strona projektu Typescript
[6]Obecny standard języka Javascript
[7]Strona projektu Node.js
[8]Strona projektu Angular
[9]Strona projektu Electron
[10]Strona projektu Jest
[11]Strona projektu Cypress
[12]Dokumentacja języka C#
[13]Strona projektu RabbitMQ
[14]Strona projektu Ansible
[15]Strona projektu Vagrant
[16]Strona projektu libvirt
[17]Strona projektu libvirt
[18]Opis na stronie firmy Microfost

  - Arch Linux[19] - system operacyjny uruchamiany przez maszyny wirtualne
  - GNU/Linux - wspierany system operacyjny

- Różne
  - Swagger Codegen[20] - automatyczna generacja API na podstawie specyfikacji
  - RDP[21] - łączenie ze zdalnymi sesjami

[19]Strona systemu operacyjnego Arch Linux
[20]Opis narzędzia na stronie firmy Swagger
[21]Dokumnetacja protokołu RDP od Microsoft