Office Space Manager

Dokument specyfikacji

8 Listopad 2021

1 Lista członków projektu

1. Grzegorz Nieużyła
2. Dawid Karolewski

2 Zakres projektu

2.1 Lista cech

- Możliwość rezerwacji, modyfikacji oraz usunięcia rezerwacji miejsca do pracy w wybranym slocie czasowym
- Inteligenty wybór bliskiego pomieszczenia w zależności od dokonanych rezerwacji przez ludzi w danym zespole
- Powiadomienia mailowe o stanie rezerwacji
- Zapis do pliku istotnych informacji o rezerwacji, bądź raportu z nich danego dnia
- Gromadzenie statystyk z zarządzania powierzchnia biurową
- Dodawanie komentarzy do danej rezerwacji w celu dyskusji pewnych szczegółów związanych z miejscem
- Możliwość deklaracji dodatkowego wyposażenia na rezerwowanym stanowisku/pomieszczeniu

2.2 Lista celów produktu

- Brak konieczności zatrudniania osób do zarządzania powierzchnią biurową
- Redukcja kolizji rezerwowanych pomieszczeń
- Powiadomienia użytkowników o stanie rezerwacji
- Gromadzenie statystyk do dalszych analiz

2.3 Lista produktów

- Aplikacja zarządzająca powierzchnią biurową
- Panel administracyjny aplikacji
- Dokumentacja
- Szkolenia z użytkowania i administracji systemu

2.4 Harmonogram realizacji projektu

Realizacja projektu zakłada dostarczenie poszczególnych części w danych ramach czasowych:

- 01.11 - Dokumentu wizyjnego
- 08.11 - Dokumentu specyfikacji
- 15.11 - Proof of concept aplikacji
- 22.11 - Dokumentacja systemu
- 6.12 - Komponent logowania
- 20.12 - Rozwój komponentu rezerwacji
- 3.01 - Panel administracyjny
- 10.01 - Komponenty notyfikujący oraz raportujący
- 17.01 - Testy jednostkowe i manualne aplikacji
- 23.01 - Finalnej końcowej wersji implementacji systemu

Rys.1 Diagram Gantta

2.5 Opis kosztów projektu

Koszty, które trzeba ponieść:

- Koszty czasu programistów - 80h * 100zł/h = 8000zł
- Media (prąd) - 20zł
- Zużycie komputera - 100zł
- Licencja oprogramowania (Intellij IDEA) - 2000zł * 2 = 4000zł
- Koszty serwerów - 2000zł rocznie

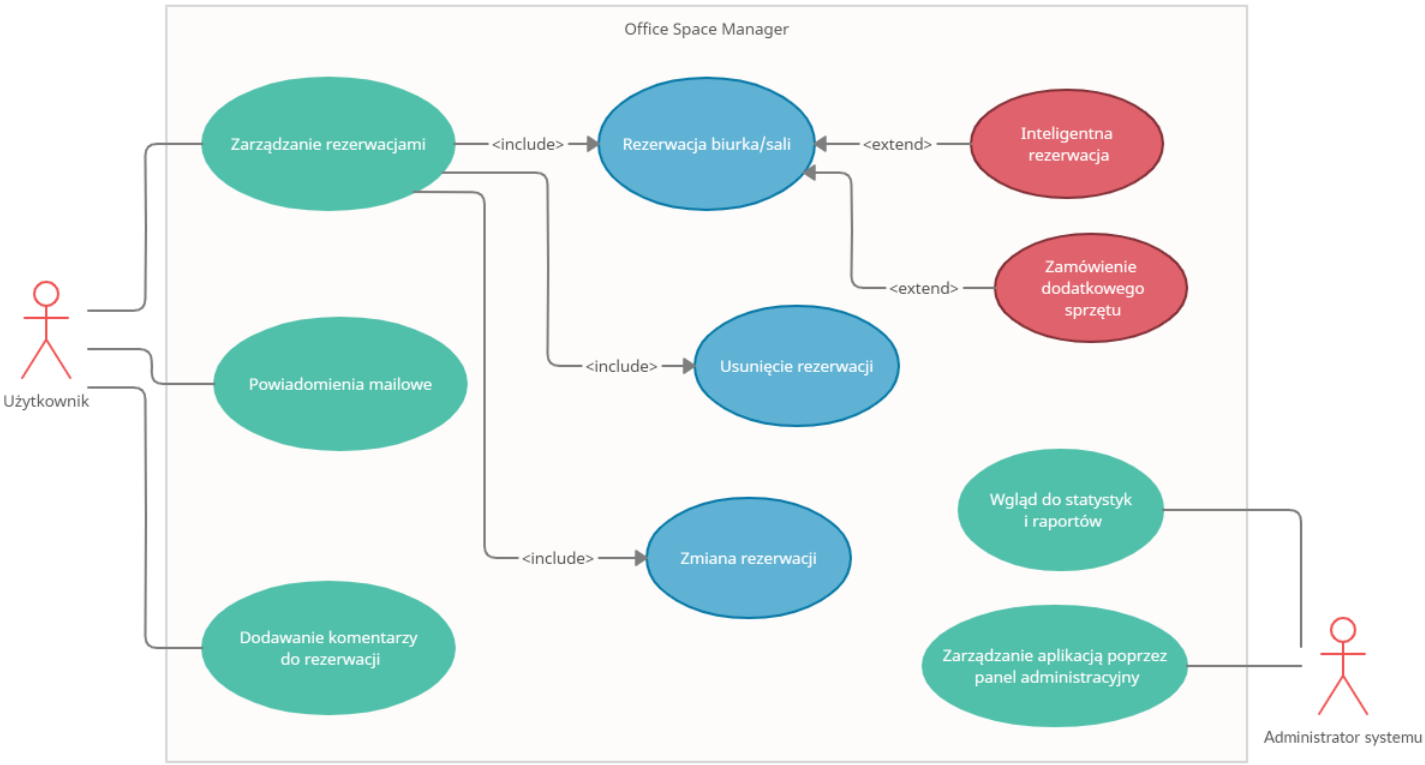
Koszty, które udało się zredukować to:

- Użyte API oraz frameworki są darmowe

3 Przypadki użycia

Przypadki użycia systemu prezentuje **Rys.2**. Występuje w nim dwóch głównych aktorów. Użytkownik ma do dyspozycji 3 główne funkcjonalności: zarządzanie

rezerwacjami, otrzymywanie powiadomień mailowych oraz dodawanie komentarzy do rezerwacji w aplikacji. Funkcjonalność rezerwacji składa się z 3 podstawowych składowych: rezerwacji, anulacji oraz jej zmiany. Podczas składania rezerwacji użytkownik może zgłosić konieczność użycia dodatkowego sprzętu. Rezerwacja większych pomieszczeń może też być zostać dokonana w sposób inteligentny/automatyczny na podstawie miejsc rezerwacji osób z danego zespołu.

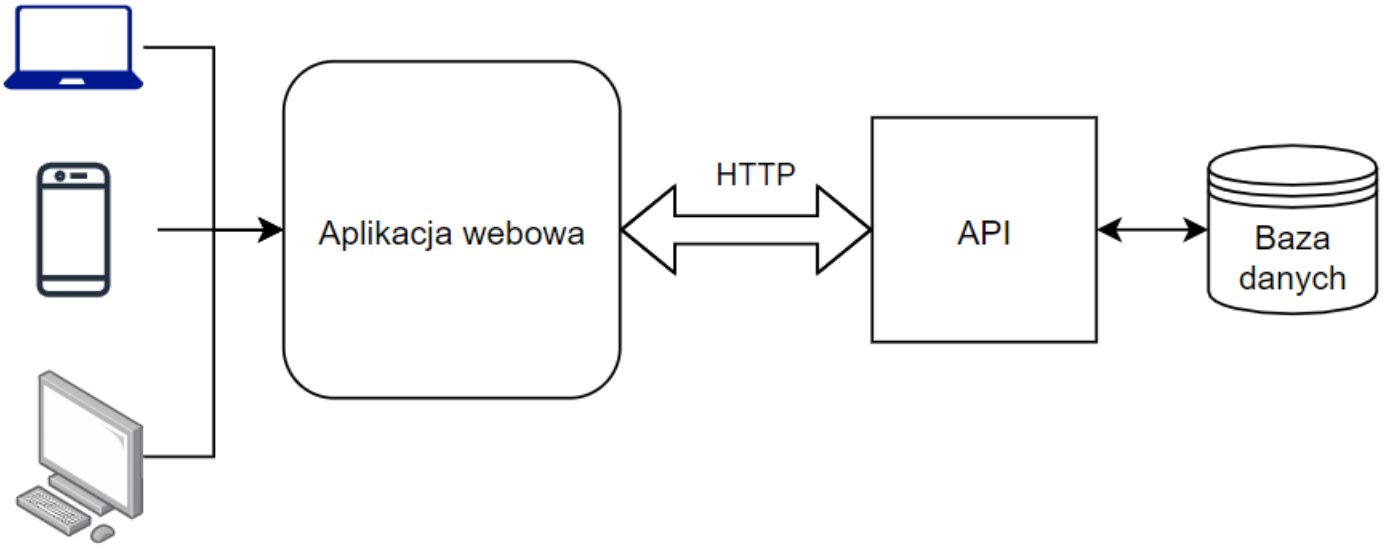


Rys.2 Diagram UML Use Case

Drugim aktorem jest administrator, który za pomocą panelu administracyjnego może zarządzać aplikacją. Ma on też wgląd do statystyk i wygenerowanych raportów systemowych.

4 Architektura

Architektura systemu została przedstawiona na **Rys.3**. Dowolne urządzenie może otworzyć przeglądarkę internetową i po zalogowaniu korzystać z aplikacji.



Rys.3 Architektura systemu

Użytkownik poprzez frontend (przyciski, formularze itp.) aplikacji w przeglądarce wyzwala akcje, które poprzez ruch HTTP powodują wysłanie żądań do backendu aplikacji (API). Następnie API przetwarza żądanie często manipulując przy tym bazą danych i po przygotowaniu odpowiedzi wysyła ją z powrotem, która jest wyświetlana użytkownikowi w odpowiedni sposób przez UI aplikacji webowej.

5 Ryzyko i plan awaryjny

- Kradzież kodu źródłowego
- Zmiana wymagań podczas trwania implementacji
- Pojawienie się konkurencyjnych rozwiązań
- Zwolnienia lekarskie programistów
- Awarie sprzętowe - serwer hostujący przestaje działać
- Ryzyko powstania luk w bezpieczeństwie systemu
- Wyciek wrażliwych danych użytkowników z bazy danych
- Nie ukończenie projektu w ustalonych ramach czasowych

W celu zabezpieczenia się przed w/w ryzykami przyjęto następujące, które stanowią plan awaryjny:

- Zabezpieczenie repozytorium oraz nadawanie praw dostępu tylko developerom
- Obniżenie cen systemu oraz wprowadzenie nowych obiecujących feature'ów, które przyciągną klientów
- Wydłużenie czasu pracy drugiego developera
- Rozważenie hostingu na przynajmniej dwóch serwerach hostujących, aby w przypadku awarii jednego przekierować ruch sieciowy na drugi serwer
- Przeprowadzenie dodatkowych skanów bezpieczeństwa przez zewnętrzne serwisy np. Veracode
- Rezygnacja z mniej istotnych funkcjonalności w celu ukończenia projektu w założonych ramach czasowych

6 Opis konfiguracji i planu zarządzania produkcją oraz testowaniem

6.1 Wymagania systemowe aplikacji i konfiguracja

W związku z tym, że projektowany system to aplikacja webowa, wyklucza to konieczność stosowania konkretnego systemu operacyjnego. Jedynym wymaganiem koniecznym do używania aplikacji jest stabilne połączenie internetowe oraz zainstalowana dowolna przeglądarka internetowa.

6.2 Testowanie i kontrola wersji

System będzie testowany manualnie na maszynie spełniającej wymagania aplikacji zawarte w poprzednim sekcji oraz przez testy jednostkowe. W przypadku dalszego rozwoju aplikacji, jeśli wystąpi konieczność komunikacji z innymi serwisami, dodane zostaną również testy integracyjne.Kod źródłowy aplikacji będzie przechowywany w serwisie GitHub, w prywatnym repozytorium zarządzany przez system Git.