

# Office Space Manager

Dokument techniczna

17 Styczeń 2022

## 1 Lista członków projektu

1. Grzegorz Nieużyła
2. Dawid Karolewski

## 2 Opis działania aplikacji

Aplikacja umożliwia:

- Możliwość rezerwacji, modyfikacji oraz usunięcia rezerwacji miejsca do pracy w wybranym słocie czasowym
- Inteligentny wybór bliskiego pomieszczenia w zależności od dokonanych rezerwacji przez ludzi w danym zespole
- Maile przy utworzeniu użytkownika i po wykonaniu rezerwacji
- Generacja raportów tygodniowych
- Gromadzenie statystyk z zarządzania powierzchnią biurową - ilość rezerwowanego sprzętu i stan miejsc
- Możliwość deklaracji dodatkowego wyposażenia na rezerwowanym stanowisku/pomieszczeniu

Użytkownik:

1. Rezerwacja miejsca - należy najpierw zalogować się za pomocą loginu i hasła wysłanego w emailu, następnie po przekierowaniu na stronę główną wypełnić formularz rezerwacji widoczny po prawej stronie
2. Zmiana/usunięcie rezerwacji - klikamy na pasku nawigacyjnym 'dashboard', a po przekierowaniu klikamy odpowiednio na 'modify reservation' albo 'cancel reservation', a następnie wypełniamy odpowiedni formularz i zatwierdzamy

Admin:

1. Blokowanie miejsca - po zalogowaniu należy na stronie głównej po prawej stronie wpisać miejsce, które chcemy zablokować oraz zatwierdzić
2. Przegląd statystyk - w celu wglądu do statystyk należy kliknąć na pasku nawigacyjnym w 'Admin panel', panel wyświetla się po prawej stronie
3. Dodawanie/usuwanie użytkownika oraz pobranie raportu - podobnie jak w przypadku przeglądu statystyk klikamy w 'Admin panel', a następnie po lewej stronie klikamy odpowiedni przycisk oraz wypełniamy odpowiednie dane w przypadku dodawania/usuwania użytkownika

### 3 Środowisko deweloperskie

Wymagania wstępnie:

- Java - wersja 11+
- Yarn - wersja 1.21+
- IDE - rekomendowane środowisko IntelliJ IDEA

Instrukcja kompilacji i uruchomienia aplikacji:

1. Pobrać kod źródłowy projektu
2. W katalogu głównym projektu w terminalu skompilować i zbudować projekt za pomocą komendy - `mvn clean install`
3. Następnie w katalogu `/web` pobrać wszystkie potrzebne dependencje poprzez komendę - `yarn install`
4. Kolejno należy skonfigurować aplikację Spring Bootową poprzez dodanie konfiguracji typu 'Spring Boot' podając jako parametr w 'Main Class' wartość: `agh.osm.api.ApiApplication`
5. Uruchomiamy aplikację backendową
6. W celu uruchomienia frontendu należy będąc w katalogu `/web` uruchomić komendę - `yarn start`
7. Przy pierwszym uruchomieniu aplikacji należy wpisać w przeglądarce url - `localhost:8080/h2-console`, a następnie w konsoli H2 uruchomić wszystkie skrypty, które znajdują się w katalogu `/utils`
8. Wchodzimy w przeglądarkę i wpisujemy url - `localhost:3000`

## 4 Wykorzystywane algorytmy

W aplikacji wykorzystano wiele standardowych algorytmów typu CRUD. Do jednego z ciekawszych algorytmów należy inteligentny wybór pomieszczenia dla osób w tym samym zespole. Algorytm ten zawiera następujące kroki:

1. Po kliknięciu przycisku do inteligentnej rezerwacji przez użytkownika, frontend wysła informację o położeniu (x, y środka wszystkich miejsc) do backendu
2. Na podstawie ID użytkownika, który jest aktualnie zalogowany, system selekcjonuje wszystkich użytkowników w danym zespole, w którym zajmują się bieżący użytkownik
3. Dla wszystkich tych użytkowników brane są wszystkie rezerwacje tych osób, a następnie są mapowane do wszystkich miejsc, które obejmują te rezerwacje
4. Następnie dla wszystkich dużych pomieszczeń jest liczona odległość do tych miejsc zajmowanych przez osoby z danego zespołu na podstawie metryki Manhattan
5. Wybierane jest pomieszczenie dla którego metryka ta ma najmniejszą wartość

---

**Algorithm 1** Inteligentna rezerwacja dla zespołu

---

**Require:** *userId*, *date*, *placePositionList*

*placesInTeam*  $\leftarrow$  *getPlacesInTeam*(*userId*, *date*)

*freeRooms*  $\leftarrow$  *getFreeRooms*(*date*)

*chosenPlace*  $\leftarrow$  null

*minDistance*  $\leftarrow$  MAX-INTEGER

**for** *room*  $\in$  *freeRooms* **do**

*dist*  $\leftarrow$  0

**for** *place*  $\in$  *placesInTeam* **do**

*dist*  $\leftarrow$  *dist* + *manhattan*(*room*, *place*)

**end for**

**if** *dist* < *minDistance* **then**

*minDistance*  $\leftarrow$  *dist*

*chosenPlace*  $\leftarrow$  *place*

**end if**

**end for**

*reservationService.createReservation*(*userId*, *chosenPlace*, *date*)

---

## 5 Użycie bibliotek zewnętrznych

W systemie użyto wiele zależności, które dla części backendowej znajdują się w pom.xml, a dla części frontendowej w package.json. Do implementacji wykorzystano następujących bibliotek i narzędzi:

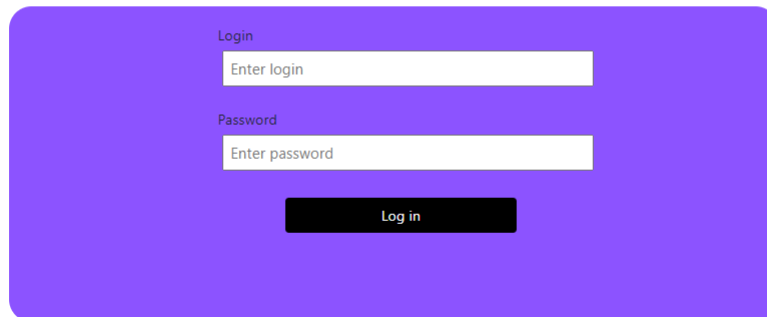
- Spring Boot - narzędzie ułatwiające tworzenia aplikacji (tutaj backend)
- H2 - lokalna baza danych
- Lombok - automatyczna generacja setterów, getterów, dodatkowe linkowanie itp.
- Spring Security - zarządzanie bezpieczeństwem aplikacji, algorytmy kryptograficzne
- Javax Mail - wysyłanie email
- Lowagie - generacja raportów
- React - budowy komponentów UI
- Redux - zarządzanie stanem (zmienne, parametry, pola) aplikacji
- Jest - testowanie elementów UI
- Axios - komunikacja z REST API
- ReCharts - rysowanie wykresów
- StyledComponents - stylowanie komponentów

## 6 Komponenty i moduły

Główne komponenty i moduły w aplikacji:

- Moduł logowanie - zabezpiecza aplikację

# Office Space Manager



A login panel with a purple background. It contains two input fields: 'Login' with the placeholder 'Enter login' and 'Password' with the placeholder 'Enter password'. Below the fields is a black button with the text 'Log in'.

**Rys.1** Panel logowania

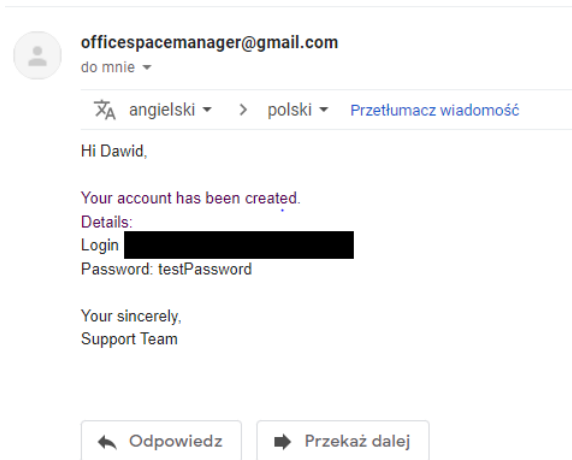
- Moduł raporty - pozwala na generowanie raportów, który jest pobierany w module administracyjnym

```
=====RESERVATION STATUS REPORT=====

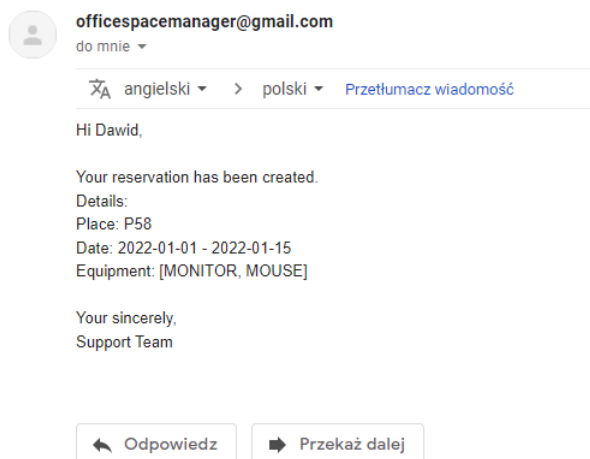
-----
Date: 2022-01-15
Free desks: 252/268
Free rooms: 12/13
-----
Date: 2022-01-16
Free desks: 252/268
Free rooms: 12/13
-----
Date: 2022-01-17
Free desks: 252/268
Free rooms: 12/13
-----
Date: 2022-01-18
Free desks: 252/268
Free rooms: 12/13
-----
Date: 2022-01-19
Free desks: 252/268
Free rooms: 12/13
-----
Date: 2022-01-20
Free desks: 253/268
Free rooms: 12/13
-----
Date: 2022-01-21
Free desks: 253/268
Free rooms: 12/13
```

**Rys.2** Wygenerowany raport rezerwacji

- Moduł email - pozwala na wysyłanie 2 rodzajów email - pierwszy ze szczegółami rezerwacji przy utworzeniu rezerwacji oraz drugi przy utworzeniu konta dla użytkownika



**Rys.3** Email wysyłany po utworzeniu konta



**Rys.4** Email wysyłany po dokonaniu rezerwacji

- Moduł rezerwacji - pozwala na zarezerwowanie miejsca dla danych dni (max 14) z dodatkowym sprzętem oraz możliwością wysłania emaila

**RESERVATION GRID**

Floor 1

01/15/2022

Reservation for Team

P101	P102	P117	P118	P133	P134	R103	
P103	P104	P119	P120	P135	P136		
R101		P121	P122	P137	P138	P153	P154
		P123	P124	P139	P140	P155	P156
P105	P106	P125	P126	P141	P142	P157	P158
P107	P108	P127	P128	P143	P144	P159	P160
P109	P110	P129	P130	P145	P146	P161	P162
P111	P112	P131	P132	P147	P148	P163	P164
P113	P114	R102		P149	P150	P165	P166
P115	P116			P151	P152	P167	P168

Place

Place number

From

mm/dd/yyyy

To

mm/dd/yyyy

Additional equipment

Select...

Send reminder email

☐

Save reservation

**Rys.5** Panel rezerwacji

**RESERVATION GRID**

Select...

01/15/2022

R1		P13	P14	P33	P34	P49	P50
		P15	P16	P35	P36	P51	P52
P1	P2	P17	P18	R3		P53	P54
P3	P4	P19	P20			P55	P56
P5	P6	P21	P22	P37	P38	P57	P58
P7	P8	P23	P24	P39	P40	P59	P60
R2		P25	P26	P41	P42	P61	P62
		P27	P28	P43	P44	P63	P64
P9	P10	P29	P30	P45	P46	R4	
P11	P12	P31	P32	P47	P48		

Place

Place number

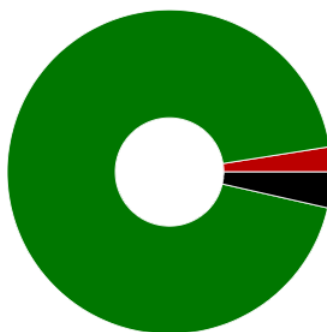
Change state

**Rys.6** Panel blokowania miejsca przez admina

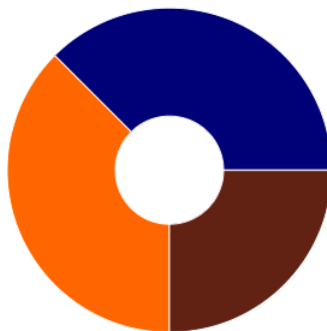
- Moduł statystyk - wyświetla 2 wykresy dla obecnego dnia - ilość miejsc zarezerwowanych, wolnych, niedostępnych oraz ilości zarezerwowanego sprzętu danego dnia

### Today's booking status

■ Reserved ■ Free ■ Not available



■ Monitor ■ Keyboard ■ Mouse



**Rys.7** Panel statystyk

- Moduł administracyjny - pozwala na usuwanie, dodawanie użytkowników, wyłączanie i włączenie danego miejsca do rezerwacji oraz pobranie raportu dla danego tygodnia



## ADMIN PANEL

This is admin panel where you can track statistics on how the system is working and generate reports. You can also manage your system from this panel here.

User ID	First name	Last name	Email
1	John	df	fsd
34	firstName	lastName	email@domain.com
66	anotherFirstName	anotherLastName	anotherEmail@domain.com
67	admin	admin	admin@email.com
130	Dawid	xxx	
194	Trevor	Hans	trevor@email.com
195	Tim	Hans	tim@email.com

Add new user

Remove user

Download reservation report

### Rys.8 Panel administracyjny

- Moduł dashboard - pozwala na podgląd obecnych rezerwacji oraz modyfikowanie i usuwanie danych rezerwacji

## DASHBOARD

Here you can view your current reservations and manage them.

### Reservations details:

Reservation ID	From	To	Place	Email notifications
195	2021-12-01	2021-12-15	P34	No
227	2022-01-01	2022-01-15	P36	No
323	2022-01-01	2022-01-15	P49	No
387	2022-01-13	2022-01-27	P56	No
388	2022-01-13	2022-01-27	P14	No
389	2022-01-14	2022-01-28	P17	No
390	2022-01-13	2022-01-20	P49	Yes

Modify reservation

Cancel reservation

### Rys.9 Panel dashboard

- Strona kontaktowa - zawiera podstawowe informacje kontaktowe do supportu

## CONTACT

If you have any questions or problems with application feel free to send us email via email below.

[officespacemeneger@gmail.com](mailto:officespacemeneger@gmail.com)



**Rys.10** Panel kontaktowy

## 7 REST API

Aplikacja udostępnia następujące endpointy:

- GET `/api/v1/users` - zwraca wszystkich użytkowników
- POST `/api/v1/users` - zapisuje nowego użytkownika
- DELETE `/api/v1/users/userId` - usuwa użytkownika o podanym ID
- GET `/api/v1/teams` - zwraca wszystkie zespoły
- POST `/api/v1/auth/login` - umożliwia proces logowania, w przypadku sukcesu zwraca informację o obecnie zalogowanym użytkowniku
- GET `/api/v1/sectors` - zwraca wszystkie sektory
- GET `/api/v1/roles` - zwraca wszystkie dostępne role
- GET `/api/v1/reservations/reservationId` - zwraca rezerwację o danym ID
- GET `/api/v1/reservations/reservationId/details` - zwraca szczegółowe informacje dla danej rezerwacji
- POST `/api/v1/reservations` - zapisuje nową rezerwację

- PATCH /api/v1/reservations/reservationId - pozwala na modyfikację istniejącej rezerwacji
- DELETE /api/v1/reservations/reservationId - usuwa daną rezerwację
- GET /api/v1/reports/generate - zwraca wygenerowany raport
- GET /api/v1/places - zwraca listę miejsc razem z obecnym ich stanem dla obecnego dnia
- GET /api/v1/places/in/date - zwraca listę miejsc z obecnym stanie dla danego dnia
- PATCH /api/v1/places/place - pozwala na modyfikację stanu danego miejsca
- POST /api/v1/places/intelligent-reservation - inteligentna rezerwacja miejsca dla zespołu
- GET /api/v1/equipment/ - zwraca listę dodatkowego wyposażenia
- GET /api/v1/equipment/now - zwraca listę dodatkowego wyposażenia dla obecnego dnia

## 8 Kluczowe fragmenty kodu, sygnatury funkcji, interfejsy

W tej sekcji przedstawiano wszystkie istotne elementy kodu i logiki biznesowej:

- UserService.addUser() - pozwala na dodanie nowego użytkownika; przyjmuje jako argument UserDto
- UserService.removeUser() - pozwala na usunięcie obecnego użytkownika; przyjmuje jako argument userId
- AuthenticationService.login() - pozwala na autentyfikację użytkownika podczas logowania; przyjmuje jako argument LoginDto; zwraca UserDto
- ReservationService.createReservation() - pozwala na rezerwację miejsca; przyjmuje jako parametr ReservationDto
- ReservationService.updateReservation() - pozwala na modyfikację rezerwacji; przyjmuje jako parametr ReservationDto oraz reservationId
- ReservationService.deleteReservation() - pozwala na usunięcie rezerwacji; przyjmuje jako parametr reservationId
- ReservationService.checkIfDateRangeOverlapsWithAnyReservation() - sprawdza czy dane rezerwacja nakłada się z inną rezerwacją w danym przedziale dat; przyjmuje jako parametr placeId, daty fromDate oraz toDate

- ReportGeneratorService.export() - pozwala na generację raportu w formacie .pdf
- PlaceService.getPlacesInDate() - zwraca aktualną listę miejsc w danym dniu; przyjmuje datę
- PlaceService.updatePlace() - pozwala na modyfikację stanu miejsca - wolne/zajęte/niedostępne
- PlaceService.intelligentReservation() - pozwala na automatyczną rezerwację miejsca najbliższego pokoju w stosunku do miejsc zarezerwowanych przez dany zespół
- EmailService.sendEmail() - interfejs udostępniający metodę pozwalającą na wysyłanie email
- EmailContentGenerator.generateCreateReservationEmailMessage() - pozwala na generację szablonu wysyłanego podczas utworzenia rezerwacji do użytkownika
- EmailContentGenerator.generateCreateAccountEmailMessage() - pozwala na generację szablonu wysyłanego po utworzeniu konta do użytkownika

## 9 Możliwe rozszerzenia

Obecnie prezentowana wersja systemu zawiera podstawowe funkcjonalności pozwalające na zarządzanie powierzchnią biurową. Do możliwych rozszerzeń mogą należeć:

- Email przypominający przed pierwszym dniem rezerwacji
- Raporty o konfigurowalnej dacie z jakiej można go wygenerować
- Dodatkowe statystyki (ilość użytkowników) oraz ich wybór w zależności od daty
- Moduł monitorujący zbierający statystyki o niepokojących zdarzeniach (dużo nieudanych prób logowania)
- Granulacja rezerwacji do godzin (obecnie najmniejszą jednostką jest dzień)
- Dodawanie komentarzy do rezerwacji
- Modyfikacja danych użytkownika

## 10 Wykryte błędy i metody ich usunięcia

Podczas testowania manualnego jak i automatycznego wykryto następujące błędy:

- Dostęp do każdej strony aplikacji przez każdą osobę - wprowadzenie logowania i menadżera zasobów w zależności od użytkownika i jego roli (niezalogowany - dostęp tylko do ekranu logowania, zalogowany użytkownika - dostęp do aplikacji bez admin panelu)
- Utworzony użytkownik ma nieograniczony czasowo dostęp do systemu - wprowadzanie nowego pola (expiration time) w tabeli user, który jest sprawdzany podczas logowania