

System przedłużania wizyt w akademiku

Analiza i projektowanie systemów informacyjnych

Spis treści

1 Cel projektu	2
2 Zakres projektu	2
3 Stos technologiczny	2
4 Model bazy danych	3
4.1 Relacyjny	3
4.2 Logiczny	4
5 Wymagania projektu	4
5.1 Wymagania funkcjonalne	4
6 Przypadki użycia	5
6.1 Aktorzy	5
6.2 Specyfikacja przypadków użycia	5
6.2.1 Moduł rejestracji i logowania użytkowników	5
6.2.2 Moduł zarządzania pobytami	7
6.2.3 Moduł raportowania	13
7 Scenariusze Testowe	14
7.1 Przeprowadzanie scenariuszy	14
7.2 Scenariusz testowy 1: Rejestracja mieszkańców	15
7.3 Scenariusz testowy 2: Umówienie wizyty przez mieszkańca	15
7.4 Scenariusz testowy 3: Anulowanie wizyty przez mieszkańca	16
7.5 Scenariusz testowy 4: Przedłużenie wizyty przez mieszkańca	16
7.6 Scenariusz testowy 5: Akceptacja bądź odrzucenie wniosku o przedłużenie wizyty przez Radę Mieszkańców	17
7.7 Scenariusz testowy 6: Wyświetlanie listy aktualnych gości przez recepcjonistę	17
7.8 Scenariusz testowy 7: Zakończenie wizyty w momencie dyscyplinarnego wyrzucenia gościa z turnu akademika przez recepcjonistę	18
7.9 Scenariusz testowy 8: Generowanie i pobieranie raportów przez uprawnionych użytkowników	18
7.10 Scenariusz testowy 9: Filtrowanie i sortowanie wizyt przez recepcjonistę	19
7.11 Scenariusz testowy 10: Dostęp do funkcji generowania raportów przez nieuprawnionych użytkowników	20
8 Dokumentacja Administratorska	21
8.1 Dodawanie nowego akademika	21
8.2 Dodawanie nowego recepcjonisty lub członka Rady Mieszkańców	22
8.3 Zarządzanie wizytami przez pracownika recepcji	23
8.4 Zarządzanie wizytami przez Członka Rady Mieszkańców	24
8.5 Generowanie i zarządzanie raportami	24
8.6 Nadanie i usunięcie uprawnień członka Rady Mieszkańców	24

1 Cel projektu

Projekt dotyczy stworzenia aplikacji umożliwiającej obsługę gości akademika. Ma zapewnić zwiększenie efektywności procesów przyjmowania i przedłużania wizyt, zapewnienie łatwiejszego dostępu do informacji dla gości, dla personelu recepcji i Rady mieszkańców akademika.

System składał się będzie z trzech modułów:

1. moduł rejestracji i logowania użytkowników
2. moduł zarządzania pobytami i wizytami
3. moduł raportowania

2 Zakres projektu

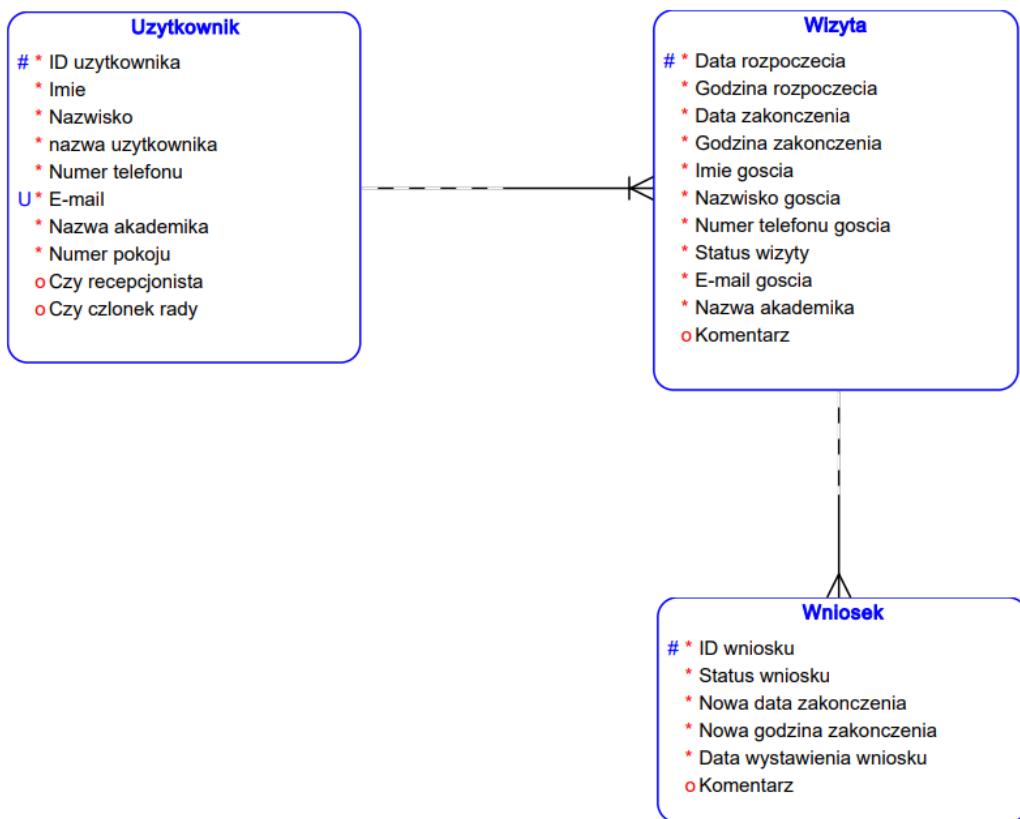
Celem projektu jest stworzenie systemu umożliwiającego kontrolę nad wizytami w akademiku. System składa się z dwóch modułów odpowiedzialnych za rejestrację użytkownika i zarządzanie wizytami oraz pobytom. Głównym zadaniem narzędzia jest możliwość umówienia wizyty gościa przez mieszkańców w danym terminie, oraz umożliwienie kontroli nad pobytami różnych gości przez personel administracyjny. Zakres projektu obejmuje również wyświetlanie listy gości przez recepcję oraz wyświetlanie historii pobytów.

3 Stos technologiczny

1. Baza danych - SQLite
2. Języki programowania - Python, JavaScript, CSS
3. Frameworki i biblioteki - Django, React, Bootstrap
4. Usługa chmurowa - TBA
5. System kontroli wersji - Git + GitLab
6. Backlog - Jira

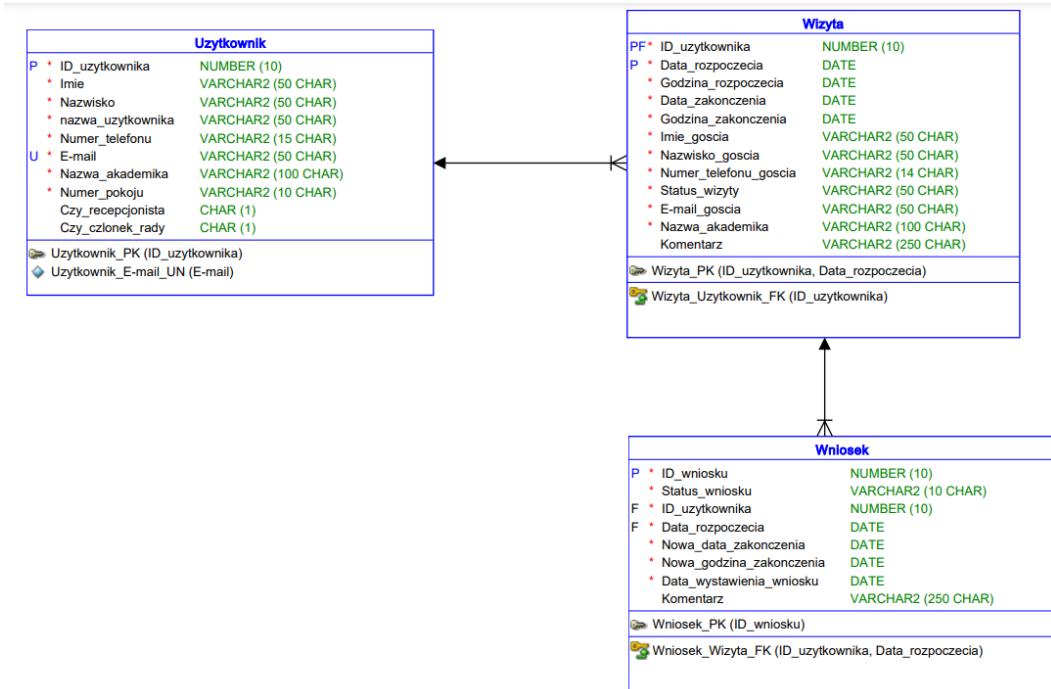
4 Model bazy danych

4.1 Relacyjny



Rysunek 1: Diagram ERD bazy danych

4.2 Logiczny



Rysunek 2: Diagram ERD bazy danych

5 Wymagania projektu

5.1 Wymagania funkcjonalne

1. Rejestracja mieszkańca
 - wprowadzenie danych mieszkańca - imię, nazwisko, numer telefonu, akademik, numer pokoju, email
 - weryfikacja konta za pomocą wiadomości email
2. Umówienie wizyty przez mieszkańca
 - wypełnienie formularza z danymi gościa
 - wygenerowanie kodu QR wysłanie go na maila gościa, by przy wejściu potwierdzić wizytę przy kamerze
 - ograniczenie wizyty do godziny 23
3. Anulowanie wizyty przez mieszkańca
 - anulowanie wizyty mieszkańca w aplikacji
 - wiadomość email dla gościa z informacją o odwołanej wizycie
4. Wgląd do aktywnych, historycznych i nadchodzących wizyt

5. Przedłużenie wizyty przez mieszkańców
 - umożliwienie przedłużenia wizyty w panelu użytkownika
 - akceptacja bądź odrzucenie wniosku przez członka Rady mieszkańców
 - wysyłanie informacji o akceptacji albo odrzuceniu zgłoszenia przez Radę Mieszkańców za pomocą wiadomość email
6. Zakończenie wizyty
 - pokazanie kodu QR przy wyjściu
 - zamknięcie wizyty w systemie
7. Wyświetlenie listy aktualnych gości przez recepcję
8. Wyświetlenie historii gości przez recepcję
9. Wyświetlenie alertu w momencie gdy gość nie opuści terenu akademika po określonym czasie
10. W przypadku wyrzucenia gościa administrator z recepcji kończy jego wizytę w systemie
11. Generowanie raportów przez administratora.

6 Przypadki użycia

6.1 Aktorzy

- Mieszkaniec - ma możliwość umówienia wizyty swojego gościa, ma wgląd do swoich aktywnych i nadchodzących wizyt, może składać wniosek o przedłużenie wizyty.
- **Administrator** z podziałem na:
 1. Pracownika recepcji, który ma wgląd do listy gości przebywających na terenie akademika, może przeglądać historię wizyt, jest informowany alertem w momencie gdy po wyznaczonym czasie gość nie opuścił terenu akademika.
 2. Radę Mieszkańców, którzy poza wszystkimi funkcjami mieszkańca mają również wgląd do listy gości przebywających na terenie akademika, mogą przeglądać historię wizyt, mają możliwość akceptacji bądź odrzucenia prośby o przedłużenie wizyty.
- Gość - dostaje powiadomienie o planowanej wizycie wraz z kodem QR, który umożliwia wejście do akademika, dostaje powiadomienie o przedłużeniu wizyty.

6.2 Specyfikacja przypadków użycia

6.2.1 Moduł rejestrowania i logowania użytkowników

1. Nazwa: Rejestracja mieszkańca

Krótki opis: Po podaniu podstawowych danych takich jak: imię, nazwisko, numer telefonu, adres e-mail, akademik, numer pokoju oraz hasło, tworzone jest konto w systemie. Wysyłany jest kod weryfikacyjny aktywujący konto.

Aktorzy: Mieszkaniec.

Warunek początkowy: Brak.

Przebiegi:

Przebieg główny - rejestracja użytkownika:

1. Mieszkaniec wchodzi w zakładkę "Rejestracja".
2. System wyświetla okno z formularzem do wypełnienia.
3. Wprowadzanie danych: imię, nazwisko, akademik, numer telefonu, adres e-mail, numer pokoju oraz hasło.
4. System wysyła link aktywacyjny na podany adres email
5. Mieszkaniec aktywuje konto.
6. System tworzy konto na podstawie wcześniej wprowadzonych danych.

Przebieg alternatywny - anulacja w trakcie rejestracji:

1. Mieszkaniec wchodzi w zakładkę "Rejestracja".
2. System wyświetla okno z formularzem do wypełnienia.
3. Mieszkaniec decyduje się anulować rejestrację i kliką w przycisk "anuluj".
4. System wychodzi z okna rejestracji i wraca na stronę główną.

Przebieg wyjątkowy - konto nie zostało zweryfikowane:

1. Mieszkaniec wchodzi w zakładkę "Rejestracja".
2. System wyświetla okno z formularzem do wypełnienia.
3. Wprowadzanie danych: imię, nazwisko, numer telefonu, adres e-mail, numer pokoju oraz hasło.
4. System wysyła link aktywacyjny na adres e-mail.
5. Mieszkaniec nie aktywuje konta.
6. System nie tworzy konta na podstawie podanych danych.

Warunek końcowy: Nowy użytkownik zapisywany jest w bazie danych i umożliwia późniejsze logowanie do systemu za pomocą podanego adresu email oraz hasła. Alternatywnie - konto nie jest tworzone, jeśli konto nie zostanie zweryfikowane.

2. Nazwa: Logowanie mieszkańca

Krótki opis: Mieszkaniec podaje adres e-mail i hasło w celu zalogowania się do systemu. Jeśli dane są poprawne, mieszkaniec uzyskuje dostęp do swojego konta. W przeciwnym razie system wyświetla komunikat o błędzie.

Aktorzy: Mieszkaniec.

Warunek początkowy: Mieszkaniec posiada aktywne konto w systemie.

Przebiegi:

Przebieg główny - logowanie użytkownika:

1. Mieszkaniec wchodzi w zakładkę "Logowanie".

2. System wyświetla okno z formularzem logowania.
3. Wprowadzanie danych: adres e-mail oraz hasło.
4. System weryfikuje podane dane.
5. Jeśli dane są poprawne, system loguje mieszkańców i wyświetla jego konto.

Przebieg wyjątkowy - błędne dane logowania:

1. Mieszkaniec wchodzi w zakładkę "Logowanie".
2. System wyświetla okno z formularzem logowania.
3. Wprowadzanie danych: adres e-mail oraz hasło.
4. System weryfikuje podane dane.
5. Jeśli dane są niepoprawne, system wyświetla komunikat o błędnych danych logowania i prosi o ponowne ich wprowadzenie.

Warunek końcowy: Mieszkaniec jest zalogowany do systemu i ma dostęp do swojego konta. Alternatywnie, mieszkaniec nie jest zalogowany, jeśli wprowadzone dane są niepoprawne.

6.2.2 Moduł zarządzania pobytom i wizytami

1. Nazwa: Umówienie wizyty (mieszkaniec)

Krótki opis: Mieszkaniec wprowadza termin wizyty oraz dane gościa do systemu. Gość dostaje powiadomienie wraz z kodem QR uprawniającym do wejścia na teren akademika w wyznaczonym terminie.

Aktorzy: Mieszkaniec, gość.

Warunek początkowy: Mieszkaniec jest zalogowany w systemie.

Przebiegi:

Przebieg główny - umówienie wizyty:

1. Mieszkaniec wchodzi w zakładkę "Zaplanuj wizytę"
2. System wyświetla okno z formularzem do wypełnienia.
3. Wprowadzanie danych: imię, nazwisko, numer telefonu, adres e-mail gościa.
4. System wysyła kod QR służący do rozpoczęcia wizyty na adres e-mail.
5. System wpisuje do bazy danych planowaną wizytę wraz z danymi gościa.

Przebieg główny - anulowanie rejestracji wizyty:

1. Mieszkaniec wchodzi w zakładkę "Zaplanuj wizytę"
2. System wyświetla okno z formularzem do wypełnienia.
3. Mieszkaniec decyduje się anulować rejestrację wizyty i kliknie w przycisk "anuluj".
4. System wychodzi z okna umawiania wizyt i wraca na stronę główną.

Warunek końcowy: Umówiona wizyta pojawia się w bazie danych systemu. Możliwy jest wgląd do planowanej wizyty z konta mieszkańców.

2. Nazwa: Anulowanie wizyty (mieszkaniec)

Krótki opis: Mieszkaniec anuluje zaplanowaną wizytę swojego gościa.

Aktorzy: Mieszkaniec, gość

Warunek początkowy: Mieszkaniec jest zalogowany w systemie. Wizyta gościa jest zaplanowana.

Przebiegi:

Przebieg główny - anulowanie wizyty:

1. Mieszkaniec wchodzi w zakładkę "Zarządzaj Wizytami".
2. System wyświetla listę zaplanowanych wizyt.
3. Mieszkaniec wybiera wizytę, którą chce anulować i potwierdza swoją decyzję.
4. System anuluje wizytę i aktualizuje listę wizyt, wysyła mail informacyjny do gościa.

Warunek końcowy: Wizyta zostaje anulowana, a informacje o anulowaniu są zapisane w systemie.

3. Nazwa: Przeglądanie aktywnych i nadchodzących wizyt (mieszkaniec)

Krótki opis: Mieszkaniec może przeglądać status oraz szczegóły swoich aktywnych i nadchodzących wizyt.

Aktorzy: Mieszkaniec

Warunek początkowy: Mieszkaniec jest zalogowany w systemie.

Przebiegi:

Przebieg główny - przeglądanie aktywnych i nadchodzących wizyt:

1. Mieszkaniec wchodzi w zakładkę "Strona główna".
2. System wyświetla nadchodzące wizyty w formie pozycji w kalendarzu.
3. Mieszkaniec przegląda szczegóły wizyt, w tym dane gości i planowane czasy wizyt poprzez najechanie kursem myszy na pozycje w kalendarzu.

Warunek końcowy: Mieszkaniec ma pełny wgląd w swoje aktywne i nadchodzące wizyty, co umożliwia lepsze zarządzanie swoim czasem i planowaniem przestrzeni.

4. Nazwa: Przedłużenie wizyty (mieszkaniec)

Krótki opis: Mieszkaniec składa wniosek o przedłużenie czasu wizyty swojego gościa.

Aktorzy: Mieszkaniec, gość

Warunek początkowy: Mieszkaniec jest zalogowany w systemie. Wizyta gościa jest aktywna.

Przebiegi:

Przebieg główny - pomyślne przedłużenie wizyty:

1. Mieszkaniec wchodzi w zakładkę "Zarządzaj wizytami".
2. System wyświetla listę aktywnych i zaplanowanych wizyt.
3. Mieszkaniec wybiera wizytę, którą chce przedłużyć.
4. Mieszkaniec kliką przycisk "Przedłuż wizytę".

5. System wysyła wniosek o przedłużenie wizyty do rady mieszkańców oraz mail informacyjny do gościa.

6. Członek rady zatwierdza wniosek o przedłużenie wizyty.

7. Gość otrzymuje maila o zaakceptowaniu wniosku.

Przebieg alternatywny- wniosek o przedłużenie wizyty odrzucony:

1. Mieszkaniec wykonuje kroki (1-4) w poprzednim punkcie.

2. System wysyła wniosek o przedłużenie wizyty do rady mieszkańców oraz mail informacyjny do gościa.

3. Członek rady odrzuca prośbę o przedłużenie wizyty.

4. Gość otrzymuje maila o odrzuceniu wniosku.

Warunek końcowy: Wniosek o przedłużenie wizyty zostaje wysłany do Rady Mieszkańców. Mieszkaniec oczekuje na decyzję.

5. Nazwa: Rozpoczęcie wizyty

Krótki opis: Gość przybywa na teren akademika w wyznaczonym terminie i skanuje kod QR przy pomocy kamery podłączonej do komputera recepcji. System zmienia status wizyty na rozpoczętą, otwiera drzwi wejściowe oraz wysyła mieszkańcowi kod QR do zakończenia wizyty.

Aktorzy: Gość, recepcjonista, mieszkaniec.

Warunek początkowy: Gość posiada kod QR otrzymany wcześniej na adres e-mail, recepcjonista jest zalogowany w systemie.

Przebiegi:

Przebieg główny - rozpoczęcie wizyty:

1. Gość przychodzi do recepcji akademika.

2. Recepcjonista prosi gościa o pokazanie kodu QR.

3. Gość pokazuje kod QR recepcjonistce.

4. Recepcjonista skanuje kod QR za pomocą kamery podłączonej do komputera.

5. System weryfikuje kod QR:

a. Jeśli kod QR jest poprawny:

i. System zmienia status wizyty na "rozpoczęta".

ii. System rejestruje godzinę rozpoczęcia wizyty.

iii. System otwiera drzwi wejściowe.

iv. System wysyła mieszkańcowi kod QR do zakończenia wizyty.

v. System wyświetla potwierdzenie na ekranie recepcji.

b. Jeśli kod QR jest niepoprawny:

i. System wyświetla komunikat o błędzie na ekranie recepcji.

ii. Recepcjonista informuje gościa o nieprawidłowym kodzie QR.

iii. Gość otrzymuje instrukcje co do dalszych kroków (np. kontakt z mieszkańcem lub administracją).

Warunek końcowy: Status wizyty w systemie zmienia się na "rozpoczeta", drzwi wejściowe zostają otwarte, gość wchodzi na teren akademika, a mieszkaniec otrzymuje kod QR do zakończenia wizyty.

6. Nazwa: Zakończenie wizyty

Krótki opis: Gość opuszcza teren akademika po zeskanowaniu kodu QR, który otrzymał po rozpoczęciu wizyty. Jeśli kod QR nie działa, gość informuje recepcję, która kończy wizytę ręcznie.

Aktorzy: Gość, recepcjonista.

Warunek początkowy: Wizyta jest w trakcie, gość znajduje się na terenie akademika, gość posiada kod QR do zakończenia wizyty.

Przebiegi:

Przebieg główny - zakończenie wizyty:

1. Gość przychodzi do recepcji akademika.
2. Gość pokazuje kod QR recepcjonistce.
3. Recepcjonista skanuje kod QR za pomocą kamery podłączonej do komputera.
4. System weryfikuje kod QR:
 - a. Jeśli kod QR jest poprawny:
 - i. System zmienia status wizyty na "zakończona".
 - ii. System rejestruje godzinę zakończenia wizyty.
 - iii. System wyświetla potwierdzenie na ekranie recepcji.
 - b. Jeśli kod QR jest niepoprawny:
 - i. System wyświetla komunikat o błędzie na ekranie recepcji.
 - ii. Recepcjonista informuje gościa o nieprawidłowym kodzie QR.
 - iii. Recepcjonista ręcznie zmienia status wizyty na "zakończona" po potwierdzeniu z mieszkańcem.

Warunek końcowy: Status wizyty w systemie zmienia się na "zakończona", gość opuszcza teren akademika.

7. Nazwa: Akceptacja bądź odrzucenie wniosku o przedłużenie (Rada Mieszkańców)

Krótki opis: Rada Mieszkańców decyduje o akceptacji lub odrzuceniu wniosków o przedłużenie wizyty gości.

Aktorzy: Rada Mieszkańców

Warunek początkowy: Wnioski o przedłużenie wizyty są złożone przez mieszkańców.

Przebieg główny - przeglądanie i decydowanie o wnioskach:

1. Członek Rady Mieszkańców loguje się do systemu.
2. Wchodzi w zakładkę "Lista wszystkich wizyt".
3. Użytkownik Rady mieszkańców może odfiltrować wnioski za pomocą pola "Status wniosku o przedłużenie" wybierając opcję "Oczekujące".
4. Członek Rady Mieszkańców przegląda wnioski i na podstawie dostępnych informacji decyduje o ich akceptacji lub odrzuceniu.

5. Dla każdego wniosku, członek Rady wybiera opcję "Akceptuj" lub "Odrzuć". W przypadku wybrania opcji "Odrzuć" pokazywane jest okno dialogowe wymagające podania powodu odrzucenia.

6. Po podjęciu decyzji, system wysyła informację o wyniku do mieszkańców, który złożył wniosek.

Warunek końcowy: Decyzja o wniosku jest podjęta i skomunikowana mieszkańcowi. Wnioski są aktualizowane w systemie zgodnie z podjętymi decyzjami.

8. Nazwa: Wyświetlanie listy wizyt (administrator)

Krótki opis: Administrator może wyświetlać wszystkie wizyty, wraz z czasem wejścia, planowanego wyjścia i statusem. W tym widoku administrator może sortować, filtrować i wyszukiwać wizyty, oraz dyscyplinarnie wyrzucić gościa z akademika.

Aktorzy: Pracownik recepcji

Warunek początkowy: Pracownik recepcji jest zalogowany

Przebiegi:

Przebieg główny - wyświetlenie listy gości:

1. Po zalogowaniu administrator znajduje się na stronie głównej.
2. System wyświetla listę wizyt w danym akademiku.

Przebieg alternatywny - wyszukanie danej osoby:

1. Po zalogowaniu administrator znajduje się na stronie głównej.
2. System wyświetla listę wizyt w danym akademiku.
3. Administrator wybiera czy szuka gościa, czy mieszkańca akademika zaznaczając odpowiednie pole.
4. Administrator wpisuje dane osoby do okna wyszukiwania.
5. Jeżeli dana osoba znajduje się w bazie danych, zostaje odfiltrowana, a wszystkie wizyty z nią związane, wyświetlane.

Przebieg alternatywny - filtrowanie po czasie wizyty:

1. Po zalogowaniu administrator znajduje się na stronie głównej.
2. System wyświetla listę wizyt w danym akademiku.
3. Administrator wybiera filtr, który pozwala na wyświetlenie gości, których wizyta odbywała się, odbywa się, lub będzie się odbywać w wybranym oknie czasowym.
4. System filtryuje listę osób zgodnie z wybranym kryterium.

Przebieg alternatywny - filtrowanie po statusie wizyty:

1. Po zalogowaniu administrator znajduje się na stronie głównej.
2. System wyświetla listę wizyt w danym akademiku.
3. Administrator wybiera filtr, który pozwala na wyświetlenie gości z określonym statusem wizyty: Zaplanowana, Trwa, Zakończona, Gość wyrzucony, Anulowana.
4. System filtryuje listę osób zgodnie z wybranym kryterium.

Przebieg alternatywny - filtrowanie po przedłużeniu:

1. Po zalogowaniu administrator znajduje się na stronie głównej.
2. System wyświetla listę wizyt w danym akademiku.
3. Administrator wybiera filtr, który pozwala na wyświetlenie gości z określonym statusem przedłużenia wizyty: Brak, Oczekujący, Zatwierdzony, Odrzucony.
4. System filtryuje listę osób zgodnie z wybranym kryterium.

Przebieg alternatywny - sortowanie:

1. Po zalogowaniu administrator znajduje się na stronie głównej.
2. System wyświetla listę wizyt w danym akademiku.
3. Administrator wybiera sposób sortowania od najnowszych lub od najstarszych wizyt (po dacie rozpoczęcia wizyty).
4. System sortuje listę osób zgodnie z wybranym kryterium.

Warunek końcowy: System wyświetla listę wizyt zgodnie z kryteriami wprowadzonymi przez administratora.

9. Nazwa: Wyświetlenie informacji o przekroczeniu czasu wizyty (administrator)

Krótki opis: System zaznacza wizytę, której czas został przekroczony.

Aktorzy:

- Pracownik recepcji
- Gość

Warunek początkowy: Gość przekroczył planowany czas trwania wizyty.

Przebiegi:

1. Administrator wyświetla panel ze wszystkimi wizytami.
2. Jeżeli wizyta trwa i czas jej trwania został przekroczony, system podświetla daną wizytę na czerwono informując administratora o tym fakcie.

Warunek końcowy: Gość przedłuża wizytę wykorzystując system przedłużek, lub jest dyscyplinarnie usuwany z akademika.

10. Nazwa: Zakończenie wizyty w momencie dyscyplinarnego wyrzucenia gościa z terenu akademika (administrator)

Krótki opis: Proces zakończenia wizyty gościa, który naruszył regulamin akademika, co skutkuje dyscyplinarnym wyrzuceniem z obiektu przez administratora.

Aktorzy:

- Pracownik recepcji
- Gość (wyrzucany)

Warunek początkowy: Gość został przyłapany na naruszeniu istotnych zasad akademika, co skutkuje koniecznością natychmiastowego usunięcia go z terenu.

Przebiegi:

Przebieg główny - gość jest wydalany z akademika:

1. Pracownik recepcji akademika otrzymuje zgłoszenie o naruszeniu regulaminu przez gościa.
2. Pracownik recepcji weryfikuje tożsamość gościa oraz dowody naruszenia zasad.
3. Pracownik recepcji informuje gościa o decyzji dyscyplinarnego usunięcia i konieczności natychmiastowego opuszczenia akademika.
4. Pracownik recepcji monitoruje opuszczenie terenu przez gościa oraz finalizuje zakończenie wizyty w systemie poprzez naciśnięcie przycisku wyrzuć, przy trwającej wizyty.
5. Pracownik recepcji uzupełnia formularz w którym wpisuje powód wydalenia.
6. Po przesłaniu formularza, status wizyty zmienia się na "Gość wyrzucony", system przesyła informację o wyrzuceniu na adres e-mail gościa.

Przebieg alternatywny - gość odmawia opuszczenia akademika:

1. Pracownik recepcji akademika informuje gościa o decyzji dyscyplinarnego usunięcia i konieczności natychmiastowego opuszczenia akademika.
2. Gość odmawia opuszczenia obiektu i ignoruje polecenia administratora.
3. Pracownik recepcji powiadamia straż akademicką o sytuacji, przedstawiając szczegóły incydentu i identyfikację gościa.
4. Straż akademicka przybywa na miejsce, aby zainterweniować.
5. Po krótkiej rozmowie z gościem, straż akademicka fizycznie eskortuje gościa poza teren akademika.
6. Pracownik recepcji monitoruje opuszczenie terenu przez gościa oraz finalizuje zakończenie wizyty w systemie poprzez naciśnięcie przycisku wyrzuć, przy trwającej wizyty.
7. Pracownik recepcji uzupełnia formularz w którym wpisuje powód wydalenia.
8. Po przesłaniu formularza, status wizyty zmienia się na "Gość wyrzucony", system przesyła informację o wyrzuceniu na adres e-mail gościa.

Warunek końcowy: Wizyta gościa jest zakończona, a wszelkie dokumenty i systemy są aktualizowane o przyczynę zakończenia wizyty.

6.2.3 Moduł raportowania

1. Nazwa: Wyświetlenie raportu

Krótki opis: Pracownik recepcji przechodzi do zakładki raporty, wybiera interesujący go raport i ustawia odpowiednie parametry do jego wygenerowania.

Aktorzy: Pracownik recepcji

Warunek początkowy: Pracownik recepcji jest zalogowany do systemu.

Przebiegi:

Przebieg główny - wygenerowanie raportu:

1. Administrator loguje się do systemu.
2. Administrator wchodzi w zakładkę "Raporty"
3. System umożliwia użytkownikowi wybór rodzaju raportu
4. Administrator wybiera rodzaj raportu.
5. Administrator wypełnia wymagane do wygenerowania raportu parametry.
 - Lista gości dla hosta - Administrator wyszukuje danego hosta przez wprowadzenie odpowiednich danych w pole wyszukiwania: imię i nazwisko lub adres e-mail lub numer pokoju.
 - Statystyki aktywności - Administrator wybiera okno czasowe z którego chce wygenerować raport zawierający statystyki
 - Wizyty w danym okresie - Administrator wybiera okno czasowe z którego chce wygenerować raport zawierający liczbę odbytych wizyt w danym dniu.
6. Administrator naciska przycisk "Generuj raport" po czym jest on generowany i wyświetlany

Warunek końcowy: Raport został wyświetlony.

2. Nazwa: Pobieranie raportu

Krótki opis: Pracownik recepcji przechodzi do zakładki raporty, generuje raport po czym pobiera go na swój dysk.

Aktorzy: Pracownik recepcji

Warunek początkowy: Użytkownik jest recepcjonistą.

Przebiegi:

Przebieg główny - wygenerowanie raportu:

1. Administrator loguje się do systemu.
2. Administrator wchodzi w zakładkę "Raporty"
3. Administrator generuje interesujący go raport
4. Administrator naciska przycisk "Pobierz raport" po czym jest on pobierany na jego dysk w formacie .csv.

Warunek końcowy: Raport został pobrany.

7 Scenariusze Testowe

7.1 Przeprowadzanie scenariuszy

Przeprowadzanie testów manualnych mających na celu zweryfikowanie poprawnej realizacji przypadków użycia opisanych w poprzednich sekcjach oraz dodatkowych funkcji systemu. Poniższe scenariusze prezentują działania wysokiego poziomu, wskazujące testowane aspekty potrzebne do osiągnięcia celu przypadku użycia. Scenariusze zorganizowano w formie Akcja → Sukces.

7.2 Scenariusz testowy 1: Rejestracja mieszkańców

Nazwa: Rejestracja mieszkańców

ID: TC001

Aktor główny: Mieszkaniec

Priorytet: Wysoki

Cel: Upewnienie się, że mieszkaniec może poprawnie zarejestrować konto w systemie.

Przebieg testu:

1. Mieszkaniec wchodzi w zakładkę "Rejestracja".
2. **Oczekiwany wynik:** System wyświetla okno z formularzem do wypełnienia.
3. Mieszkaniec wprowadza dane: Adres e-mail, Nazwę użytkownika, Hasło, Imię, Nazwisko, Numer telefonu, Akademik oraz Numer pokoju.
4. **Oczekiwany wynik:** Dane są wprowadzone do formularza.
5. Mieszkaniec kliką "Zarejestruj".
6. **Oczekiwany wynik:** System tworzy konto na podstawie wcześniej wprowadzonych danych.
7. **Przebieg alternatywny - Anulowanie rejestracji:** Mieszkaniec w dowolnym momencie kliką "Anuluj" i system wraca do strony głównej bez tworzenia konta.

7.3 Scenariusz testowy 2: Umówienie wizyty przez mieszkańców

Nazwa: Umówienie wizyty przez mieszkańców

ID: TC002

Aktor główny: Mieszkaniec

Priorytet: Wysoki

Cel: Upewnienie się, że mieszkaniec może poprawnie umówić wizytę swojego gościa.

Przebieg testu:

1. Mieszkaniec loguje się do systemu.
2. **Oczekiwany wynik:** Mieszkaniec jest zalogowany.
3. Mieszkaniec wchodzi w zakładkę "Umów wizytę".
4. **Oczekiwany wynik:** System wyświetla okno z formularzem do wypełnienia.
5. Mieszkaniec wprowadza dane: Datę rozpoczęcia wizyty, Czas rozpoczęcia wizyty, Datę zakończenia wizyty, Czas zakończenia wizyty, Imię gościa, Nazwisko gościa, numer telefonu gościa, adres e-mail gościa.
6. **Oczekiwany wynik:** Dane są wprowadzone do formularza.
7. Mieszkaniec kliką przycisk "Zaplanuj wizytę".
8. **Oczekiwany wynik:** System waliduje wprowadzone dane:
 - Jeśli data i czas rozpoczęcia są wcześniejsze niż obecna data i czas, system wyświetla komunikat: "Data i czas rozpoczęcia nie mogą być wcześniejsze niż obecna data i czas".

- Jeśli data i czas zakończenia są późniejsze niż 23:00 tego samego dnia, system wyświetla komunikat: "Koniec wizyty nie może przekraczać 23:00 tego samego dnia".
- 9. Oczekiwany wynik:** System wyświetla odpowiednie komunikaty o błędach.
 10. Mieszkaniec poprawia datę i czas rozpoczęcia oraz zakończenia wizyty, jeśli to konieczne.
 11. **Oczekiwany wynik:** Dane są poprawione w formularzu.
 12. Mieszkaniec kliką przycisk "Zaplanuj wizytę" ponownie.
 13. **Oczekiwany wynik:** System zapisuje wizytę, wysyła powiadomienie do gościa i wyświetla ją w kalendarzu.
 14. **Przebieg alternatywny - Anulowanie umawiania wizyty:** Mieszkaniec w dowolnym momencie kliką "Anuluj" i system wraca do strony głównej bez zaplanowania wizyty.

7.4 Scenariusz testowy 3: Anulowanie wizyty przez mieszkańców

Nazwa: Anulowanie wizyty przez mieszkańców

ID: TC003

Aktor główny: Mieszkaniec

Priorytet: Średni

Cel: Upewnienie się, że mieszkaniec może poprawnie anulować zaplanowaną wizytę swojego gościa.

Przebieg testu:

1. Mieszkaniec loguje się do systemu.
2. **Oczekiwany wynik:** Mieszkaniec jest zalogowany.
3. Mieszkaniec wchodzi w zakładkę "Zarządzaj wizytami".
4. **Oczekiwany wynik:** System wyświetla listę wizyt.
5. Mieszkaniec wybiera wizytę, którą chce anulować i potwierdza swoją decyzję.
6. **Oczekiwany wynik:** System zmienia status wizyty na "anulowana" i aktualizuje listę wizyt. Gość otrzymuje powiadomienie o anulowaniu wizyty.
7. **Przebieg alternatywny - Przerwanie procesu anulowania wizyty:** Mieszkaniec w dowolnym momencie kliką "Przerwij" i system wraca do widoku listy wizyt bez anulowania wybranej wizyty.

7.5 Scenariusz testowy 4: Przedłużenie wizyty przez mieszkańców

Nazwa: Przedłużenie wizyty przez mieszkańców

ID: TC004

Aktor główny: Mieszkaniec

Priorytet: Średni

Cel: Upewnienie się, że mieszkaniec może poprawnie złożyć wniosek o przedłużenie wizyty swojego gościa.

Przebieg testu:

1. Mieszkaniec loguje się do systemu.

2. **Oczekiwany wynik:** Mieszkaniec jest zalogowany.
3. Mieszkaniec wchodzi w zakładkę "Zarządzaj wizytami".
4. **Oczekiwany wynik:** System wyświetla listę wizyt.
5. Mieszkaniec wybiera wizytę, którą chce przedłużyć oraz kliką "Przedłuż wizytę".
6. **Oczekiwany wynik:** System wysyła wniosek o przedłużenie wizyty do Rady Mieszkańców.
7. **Przebieg alternatywny - Anulowanie wniosku:** Mieszkaniec w dowolnym momencie kliką "Anuluj" i system wraca do listy wizyt bez wysłania wniosku.

7.6 Scenariusz testowy 5: Akceptacja bądź odrzucenie wniosku o przedłużenie wizyty przez Radę Mieszkańców

Nazwa: Akceptacja bądź odrzucenie wniosku o przedłużenie wizyty przez Radę Mieszkańców

ID: TC005

Aktor główny: Rada Mieszkańców

Priorytet: Wysoki

Cel: Upewnienie się, że Rada Mieszkańców może poprawnie akceptować lub odrzucać wnioski o przedłużenie wizyt.

Przebieg testu:

1. Członek Rady Mieszkańców loguje się do systemu.
2. **Oczekiwany wynik:** Członek Rady Mieszkańców jest zalogowany.
3. Członek Rady Mieszkańców przechodzi do sekcji "Strona główna" na której znajduje się panel zarządzania.
4. **Oczekiwany wynik:** System wyświetla listę wszystkich wizyt.
5. Członek Rady Mieszkańców przegląda wnioski o przedłużenie wizyt.
6. **Oczekiwany wynik:** System wyświetla wnioski o przedłużenie wizyt.
7. Członek Rady Mieszkańców wybiera wniosek do akceptacji lub odrzucenia.
8. **Oczekiwany wynik:** System wyświetla opcje akceptacji lub odrzucenia wniosku.
9. Członek Rady Mieszkańców akceptuje lub odrzuca wniosek.
10. **Oczekiwany wynik:** System aktualizuje status wniosku i wysyła powiadomienie do gościa.

7.7 Scenariusz testowy 6: Wyświetlanie listy aktualnych gości przez recepcjonistę

Nazwa: Wyświetlanie listy aktualnych gości przez recepcjonistę

ID: TC006

Aktor główny: Pracownik recepcji

Priorytet: Wysoki

Cel: Upewnienie się, że pracownik recepcji może poprawnie wyświetlić listę wszystkich aktualnych gości w akademiku.

Przebieg testu:

1. Pracownik recepcji loguje się do systemu.
2. **Oczekiwany wynik:** Pracownik recepcji jest zalogowany.
3. Pracownik recepcji przechodzi na Stronę główną na której znajduje się cały panel.
4. **Oczekiwany wynik:** System wyświetla listę gości.

7.8 Scenariusz testowy 7: Zakończenie wizyty w momencie dyscyplinarnego wyrzucenia gościa z terenu akademika przez recepcjonistę

Nazwa: Zakończenie wizyty w momencie dyscyplinarnego wyrzucenia gościa z terenu akademika przez recepcjonistę

ID: TC007

Aktor główny: Pracownik recepcji

Priorytet: Wysoki

Cel: Upewnienie się, że pracownik recepcji może poprawnie zakończyć wizytę gościa, który naruszył regulamin akademika.

Przebieg testu:

1. Pracownik recepcji otrzymuje zgłoszenie o naruszeniu regulaminu przez gościa.
2. **Oczekiwany wynik:** Pracownik recepcji weryfikuje tożsamość gościa oraz dowody naruszenia zasad.
3. Pracownik recepcji informuje gościa o decyzji dyscyplinarnego usunięcia i konieczności natychmiastowego opuszczenia akademika.
4. **Oczekiwany wynik:** Gość jest poinformowany o decyzji.
5. Pracownik recepcji loguje się do systemu.
6. **Oczekiwany wynik:** Pracownik recepcji jest zalogowany.
7. Pracownik recepcji wybiera z panelu przycisk "Wyrzuć" przy odpowiednim gościu.
8. **Oczekiwany wynik:** System wyświetla formularz z powodem anulowania wizyty w trakcie jej trwania.
9. Pracownik recepcji wpisuje powód i kliką przycisk "Anuluj wizytę".
10. **Oczekiwany wynik:** System zmienia status wizyty na "Gość wyrzucony".
11. **Przebieg alternatywny - Gość odmawia opuszczenia akademika:** Pracownik recepcji wzywa straż akademicką, która fizycznie eskortuje gościa poza teren akademika.

7.9 Scenariusz testowy 8: Generowanie i pobieranie raportów przez uprawnionych użytkowników

Nazwa: Generowanie i pobieranie raportów przez uprawnionych użytkowników

ID: TC008

Aktor główny: Pracownik recepcji

Priorytet: Wysoki

Cel: Upewnienie się, że tylko uprawnieni użytkownicy mogą generować i pobierać raporty oraz że system odpowiednio obsługuje próby dostępu do raportów przez nieuprawnionych użytkowników.

Przebieg testu:

1. Pracownik recepcji loguje się do systemu.
2. **Oczekiwany wynik:** Pracownik recepcji jest zalogowany.
3. Pracownik recepcji przechodzi do zakładki "Raporty".
4. **Oczekiwany wynik:** System wyświetla dostępne opcje raportów:
 - Lista gości dla hosta.
 - Statystyki aktywności recepcji.
 - Raport wizyt w wybranym okresie.
5. Pracownik recepcji wybiera raport do wygenerowania:
 - Lista gości dla hosta.
 - Statystyki aktywności recepcji.
 - Raport wizyt w wybranym okresie.
6. **Oczekiwany wynik:** System wyświetla odpowiednie pola do wprowadzenia dodatkowych informacji (np. ID/nazwę/email/nr pokoju hosta, datę początkową i końcową) i przycisk "Generuj raport".
7. Pracownik recepcji wprowadza wymagane informacje i kliką "Generuj raport".
8. **Oczekiwany wynik:** System wyświetla odpowiedni raport wizyt oraz przycisk "Pobierz raport".
9. Pracownik recepcji kliką przycisk "Pobierz raport".
10. **Oczekiwany wynik:** System umożliwia pobranie raportu w formacie .csv.

7.10 Scenariusz testowy 9: Filtrowanie i sortowanie wizyt przez recepcjonistę

Nazwa: Filtrowanie i sortowanie wizyt przez recepcjonistę

ID: TC009

Aktor główny: Pracownik recepcji

Priorytet: Wysoki

Cel: Upewnienie się, że Pracownik recepcji może poprawnie filtrować i sortować wizyty według różnych kryteriów.

Przebieg testu:

1. Pracownik recepcji loguje się do systemu.
2. **Oczekiwany wynik:** Pracownik recepcji jest zalogowany.
3. Pracownik recepcji przechodzi do strony głównej, gdzie dostępne są opcje filtrowania wizyt.

4. **Oczekiwany wynik:** System wyświetla opcje filtrowania: według osoby (Gość/Host), według wyszukiwania po imieniu/nazwisku/telefonie/emailu wybranego użytkownika, według danych wizyty (data rozpoczęcia, czas rozpoczęcia, data zakończenia, czas zakończenia), status wizyty, status wniosku o przedłużenie, sortowanie.
5. Pracownik recepcji wybiera opcję filtrowania według osoby (Gość/Host).
6. **Oczekiwany wynik:** System umożliwia wpisanie imienia, nazwiska, telefonu lub email wybranego użytkownika.
7. Pracownik recepcji wprowadza dane.
8. **Oczekiwany wynik:** System automatycznie aktualizuje listę wizyt zgodnie z wprowadzonymi kryteriami.
9. Pracownik recepcji wybiera opcję filtrowania według danych wizyty.
10. **Oczekiwany wynik:** System umożliwia wybór daty i czasu rozpoczęcia oraz zakończenia wizyty.
11. Pracownik recepcji wprowadza odpowiednie daty i czasy.
12. **Oczekiwany wynik:** System automatycznie aktualizuje listę wizyt zgodnie z wprowadzonymi kryteriami.
13. Pracownik recepcji wybiera status wizyty oraz status wniosku o przedłużenie.
14. **Oczekiwany wynik:** System automatycznie aktualizuje listę wizyt zgodnie z wybranymi statusami.
15. Pracownik recepcji wybiera opcję sortowania wizyt (od najnowszych lub od najstarszych).
16. **Oczekiwany wynik:** System sortuje listę wizyt zgodnie z wybranym kryterium.
17. Pracownik recepcji kliką przycisk "Resetuj filtrowanie".
18. **Oczekiwany wynik:** System resetuje wszystkie filtry (według osoby, według danych wizyty, status wizyty, status wniosku o przedłużenie) i wyświetla pełną listę wizyt.

7.11 Scenariusz testowy 10: Dostęp do funkcji generowania raportów przez nieuprawnionych użytkowników

Nazwa: Dostęp do funkcji generowania raportów przez nieuprawnionych użytkowników

ID: TC010

Aktor główny: Nieuprawniony użytkownik (np. mieszkaniec)

Priorytet: Wysoki

Cel: Upewnienie się, że nieuprawnieni użytkownicy nie mają dostępu do funkcji generowania raportów.

Przebieg testu:

1. Nieuprawniony użytkownik (np. mieszkaniec) loguje się do systemu.
2. **Oczekiwany wynik:** Nieuprawniony użytkownik jest zalogowany.
3. Nieuprawniony użytkownik przechodzi do zakładki "Raporty".
4. **Oczekiwany wynik:** System wyświetla komunikat o braku uprawnień do dostępu do tej funkcji.

5. Nieuprawniony użytkownik próbuje uzyskać dostęp do raportów poprzez bezpośredni link.
6. **Oczekiwany wynik:** System wyświetla komunikat o braku uprawnień do dostępu do tej funkcji i przekierowuje użytkownika na stronę główną.

8 Dokumentacja Administratorska

8.1 Dodawanie nowego akademika

Cel: Dodanie nowego akademika do systemu.

Kroki:

1. Otwórz plik konfiguracyjny:

- Otwórz plik zawierający definicję listy akademików. Na przykład `registrationForm.js`.

2. Znajdź sekcję kodu z listą akademików:

- Zlokalizuj tablicę `dormitories` w kodzie.

3. Dodaj nowy akademik:

- Dodaj nazwę nowego akademika do tablicy `dormitories`. Przykład:

```
1 const dormitories = [
2   'DS Akademik',
3   'DS Babilon',
4   'DS Bratniak-Muszelka',
5   'DS Mikrus',
6   'DS Pineska-Tulipan',
7   'DS Riviera',
8   'DS Tatrzańska',
9   'DS Ustronia',
10  'DS Wczesniak',
11  'DS Zaczek',
12  'DS Nowy' // Dodano nowy akademik
13 ];
```

4. Zapisz zmiany:

- Zapisz plik po dokonaniu modyfikacji.

5. Przetestuj zmiany:

- Uruchom aplikację i przejdź do formularza rejestracji, aby upewnić się, że nowy akademik jest dostępny w liście rozwijanej.

Wynik końcowy: Nowy akademik jest dodany do systemu i dostępny dla użytkowników podczas rejestracji.

8.2 Dodawanie nowego recepcjonisty lub członka Rady Mieszkańców

Cel: Dodanie nowego recepcjonisty lub członka Rady Mieszkańców do systemu za pomocą funkcji Django w backendzie.

Kroki:

1. Stwórz konto użytkownika:

- Skorzystaj z funkcji `create_superuser` w terminalu backendu Django.

```
python manage.py createsuperuser
```

2. Zdefiniuj parametry funkcji:

- Uzupełnij parametry funkcji odpowiednimi danymi użytkownika, takimi jak `email`, `username`, `first_name`, `last_name`, `phone_number`, `dormitory`, `room_number` oraz `password`.

3. Utwórz nowego użytkownika:

- Aby utworzyć recepcjonistę, ustaw parametr `is_receptionist` na `True`:

```
1 def create_superuser(
2     self,
3     email,
4     username,
5     first_name,
6     last_name,
7     phone_number,
8     dormitory,
9     room_number,
10    password=None,
11    is_receptionist=True,
12):
13     user = self.create_user(
14         email,
15         username,
16         first_name,
17         last_name,
18         phone_number,
19         dormitory,
20         room_number,
21         password,
22     )
23     user.is_superuser = True
24     user.is_receptionist = is_receptionist
25     user.is_community_member = not is_receptionist
26     user.save()
27     return user
```

4. Przykład dodania recepcjonisty:

```
1 create_superuser(
2     email='receptionist@example.com',
3     username='receptionist123',
4     first_name='Jan',
```

```
5     last_name='Kowalski',
6     phone_number='123456789',
7     dormitory='DS Akademik',
8     room_number='101',
9     password='password',
10    is_receptionist=True
11 )
```

5. Przykład dodania członka Rady Mieszkańców:

```
1 create_superuser(
2     email='communitymember@example.com',
3     username='community123',
4     first_name='Anna',
5     last_name='Nowak',
6     phone_number='987654321',
7     dormitory='DS Zaczek',
8     room_number='202',
9     password='password',
10    is_receptionist=False
11 )
```

6. Przetestuj zmiany:

- Zaloguj się do systemu jako pracownik recepcji lub członek Rady Mieszkańców i sprawdź, czy masz dostęp do odpowiednich funkcji.

Wynik końcowy: Nowy pracownik recepcji lub członek Rady Mieszkańców jest dodany do systemu i ma odpowiednie uprawnienia.

8.3 Zarządzanie wizytami przez pracownika recepcji

Cel: Zarządzanie wizytami gości przez pracownika recepcji.

Kroki:

1. Przeglądanie wizyt:

- Pracownik recepcji loguje się do systemu.
- Pracownik recepcji znajduje się na stronie głównej z panelem administratora.
- **Oczekiwany wynik:** System wyświetla listę zaplanowanych wizyt.

2. Wyrzucanie gości:

- Pracownik recepcji wybiera osobę do wyrzucenia.
- Pracownik recepcji kliką przycisk "Wyrzuć".
- Pracownik recepcji uzupełnia formularz z powodem wyrzucenia.
- **Oczekiwany wynik:** System anuluje wizytę i aktualizuje listę wizyt.

8.4 Zarządzanie wizytami przez Członka Rady Mieszkańców

Cel: Zarządzanie wizytami gości przez Członka Rady Mieszkańców.

Kroki:

1. Przeglądanie wizyt:

- Członek Rady loguje się do systemu.
- Członek Rady znajduje się na stronie głównej z panelem administratora.
- **Oczekiwany wynik:** System wyświetla listę zaplanowanych wizyt.

2. Przedłużanie wizyt:

- Członek Rady wybiera wizytę do przedłużenia.
- Członek Rady kliką przycisk "Zaakceptuj" bądź "Odrzuć".
- **Oczekiwany wynik:** System przedłuża wizytę bądź ją odrzuca i aktualizuje listę wizyt.

8.5 Generowanie i zarządzanie raportami

Cel: Generowanie i zarządzanie raportami przez pracownika recepcji.

Kroki:

1. Generowanie raportów:

- Pracownik recepcji loguje się do systemu.
- Pracownik recepcji przechodzi do zakładki "Raporty".
- Pracownik recepcji wybiera rodzaj raportu do wygenerowania.
- Pracownik recepcji wprowadza wymagane dane do wygenerowania raportu.
- Pracownik recepcji kliką przycisk "Generuj raport".
- **Oczekiwany wynik:** System generuje raport i wyświetla go na ekranie.

2. Pobieranie raportów:

- Pracownik recepcji kliką przycisk "Pobierz raport".
- **Oczekiwany wynik:** System umożliwia pobranie raportu w formacie .csv.

8.6 Nadanie i usunięcie uprawnień członka Rady Mieszkańców

Cel: Zarządzanie uprawnieniami członka Rady Mieszkańców w systemie poprzez aktualizację bazy danych.

Nadanie uprawnień:

Kroki:

1. Otwórz narzędzie do zarządzania bazą danych:

- Użyj preferowanego narzędzia do zarządzania bazą danych, takiego jak pgAdmin, MySQL Workbench lub komenda SQL w terminalu.

2. Zlokalizuj tabelę użytkowników:

- Otwórz tabelę `user_api_appuser`, która przechowuje informacje o użytkownikach.
3. **Znajdź użytkownika, którego uprawnienia mają zostać zmienione:**
- Wykonaj zapytanie SQL, aby znaleźć użytkownika na podstawie jego ID, nazwy użytkownika lub adresu e-mail.
- ```
SELECT * FROM user_api_appuser WHERE email='example@example.com';
```
4. **Zaktualizuj uprawnienia użytkownika:**
- Ustaw wartość kolumny `is_community_member` na 1, aby nadać uprawnienia członka Rady Mieszkańców.
- ```
UPDATE user_api_appuser SET is_community_member = 1 WHERE email='example@example.com';
```
5. **Zapisz zmiany:**
- Zapisz zmiany w bazie danych.
6. **Przetestuj zmiany:**
- Zaloguj się do systemu jako zaktualizowany użytkownik i sprawdź, czy posiada uprawnienia członka Rady Mieszkańców.
- Usunięcie uprawnień:**
- Kroki:**
- Otwórz narzędzie do zarządzania bazą danych:**
 - Użyj preferowanego narzędzia do zarządzania bazą danych, takiego jak pgAdmin, MySQL Workbench lub komenda SQL w terminalu. - Zlokalizuj tabelę użytkowników:**
 - Otwórz tabelę `user_api_appuser`, która przechowuje informacje o użytkownikach. - Znajdź użytkownika, którego uprawnienia mają zostać zmienione:**
 - Wykonaj zapytanie SQL, aby znaleźć użytkownika na podstawie jego ID, nazwy użytkownika lub adresu e-mail.

```
SELECT * FROM user_api_appuser WHERE email='example@example.com';
```

 - Zaktualizuj uprawnienia użytkownika:**
 - Ustaw wartość kolumny `is_community_member` na 0, aby usunąć uprawnienia członka Rady Mieszkańców.

```
UPDATE user_api_appuser SET is_community_member = 0 WHERE email='example@example.com';
```

5. Zapisz zmiany:

- Zapisz zmiany w bazie danych.

6. Przetestuj zmiany:

- Zaloguj się do systemu jako zaktualizowany użytkownik i sprawdź, czy uprawnienia członka Rady Mieszkańców zostały usunięte.

Wynik końcowy: Użytkownik ma nadane lub usunięte uprawnienia członka Rady Mieszkańców i może korzystać z odpowiednich funkcji w systemie lub nie ma do nich dostępu.