**W A R S Z A W S K A  
W Y Ż S Z A S Z K O Ł A I N F O R M A T Y K I**

PROJEKT ZESPOŁOWY

AUTORZY: Marcin Szlassa, Kamil Rojek, Rafał Borycki, Mariusz Grzybowski

System wspierający wystawianie/rezerwacje apartamentów pensjonatu "Sleep Well".

Grupa 4

Prowadzący projekt:

Mgr Szymon Smaga

*Projekt spełnia wymagania stawiane projektom zespołowym na studiach pierwszego stopnia. :)*

W A R S Z A W A 2018

Spis treści

[1 Analiza systemu 4](#_Toc530762111)

[1.1 Przestawienie koncepcji sytemu 4](#_Toc530762112)

[1.1.1 WSTĘP 4](#_Toc530762113)

[1.1.2 Wprowadzenie do problemu. 4](#_Toc530762114)

[1.1.3 Ogólna charakterystyka działania pensjonatu. 4](#_Toc530762115)

[1.1.4 Założenia ogólne systemu. 7](#_Toc530762116)

[1.2 Specyfikacja funkcjonalna 9](#_Toc530762117)

[1.2.1 Diagram hierarchii funkcji (FHD) 9](#_Toc530762118)

[1.2.2 Lista funkcji z opisem ich zastosowania 10](#_Toc530762119)

[1.2.3 Diagramy przypadków użycia 13](#_Toc530762120)

[1.3 Model danych 15](#_Toc530762121)

[1.3.1 Tabela z opisem encji oraz związków 15](#_Toc530762122)

[1.3.2 Diagram związków encji (ERD) 17](#_Toc530762123)

[1.4 Model architektury systemu 17](#_Toc530762124)

[1.4.1 Opis diagramu architektury systemu w postaci tabeli 18](#_Toc530762125)

[1.4.2 Uzasadnienie wybranego modelu 18](#_Toc530762126)

[1.4.3 Diagram architektury systemu 20](#_Toc530762127)

[2 Projekt systemu 20](#_Toc530762128)

[2.1 Projekt architektury systemu 20](#_Toc530762129)

[2.1.1 Schemat i diagram warstw 21](#_Toc530762130)

[2.1.2 Diagram klas 22](#_Toc530762131)

[2.2 Projekt bazy danych 24](#_Toc530762132)

[2.2.1 Tabele bazy danych 24](#_Toc530762133)

[2.2.2 Pola bazy danych 25](#_Toc530762134)

[2.2.3 Projekt schematu relacji bazy danych. 30](#_Toc530762135)

[2.2.4 Mechanizmy bazy danych. 31](#_Toc530762136)

[2.3 Projekt algorytmów 31](#_Toc530762137)

[2.3.1 Diagramy czynności 31](#_Toc530762138)

[2.3.2 Diagramy sekwencji 40](#_Toc530762139)

[2.3.3 Diagramy współpracy 41](#_Toc530762140)

[2.4 Projekt interfejsu użytkownika 42](#_Toc530762141)

[2.4.1 Interfejs Klienta 42](#_Toc530762142)

[2.4.2 Interfejs Pracownika Recepcji 46](#_Toc530762143)

[2.4.3 Interfejs Managera 46](#_Toc530762144)

[2.5 Projekt testów kontrolnych 47](#_Toc530762145)

[2.5.1 Testy jednostkowe 47](#_Toc530762146)

[2.5.2 Testy bezpieczeństwa 47](#_Toc530762147)

[2.5.3 Testy zgodności 48](#_Toc530762148)

# Analiza systemu

## Przestawienie koncepcji sytemu

### WSTĘP

**Tematem pracy zespołowej jest stworzenie systemu, który będzie wspierał zarządzanie procesem rezerwacji pokoi pensjonatu "Sleep Well".**

### Wprowadzenie do problemu.

Prowadzenie nowoczesnego pensjonatu w dzisiejszych czasach wymaga od pracowników konieczności łączenia wiedzy niezbędnej do obsługi klientów, jak również podstawowej wiedzy informatycznej związanej ze znajomością obsługi komputera. Pozwala to na znacznie szybsze wyszukiwanie dostępnych pokoi, skrócenie czasu obsługi, możliwość lepszego doradzania klientowi, a przede wszystkim ograniczenie kosztów związanych z prowadzeniem takiego biznesu i co bardzo ważne oszczędności czasu. Ograniczenie czasochłonności wykonywanych operacji dzięki wykorzystaniu systemu informatycznego, który wspiera działanie pensjonatu pozwala znacznie ograniczyć koszty związane z obsługą klientów, szybciej reagować na pojawienie się ewentualnych braków materiałów eksploatacyjnych tj. środki czystości, kosmetyki dla klientów, żywności dla kucharza pensjonatu. Prowadzi to również do zmniejszenia liczby pracowników potrzebnych w firmie. W tym projekcie przestawimy jeden ze sposobów poprawienia działania pensjonatu dzięki zastosowaniu systemu informatycznego. Na podstawie zlecenia otrzymanego od właściciela pensjonatu "Sleep Well", który chce poprawić efektywność działania swojego pensjonatu, zwiększyć zyski z wynajmu, jak również dostosować swój biznes do standardów wykorzystywanych w XXI wieku. "Gang czworga" wysłał Marcina Szlassę, specjalistę od "wywiadu w terenie i umiejętności miękkich", w celu pozyskania kluczowych informacji potrzebnych do budowy takiego systemu.

### **Ogólna charakterystyka działania pensjonatu.**

Na podstawie ankiety przeprowadzonej z pracownikami pensjonatu pozyskaliśmy informacje o sposobie jego funkcjonowania, które później zostaną wykorzystane do zaprojektowania bazy danych, jak również całego systemu wpierającego działanie pensjonatu.

Jednym z pracowników pensjonatu jest recepcjonistka, która zajmuje się bezpośrednią obsługą klienta. Recepcjonistka musi udzielać potencjalnym klientom pensjonatu informacji na temat terminów i dostępności pokoi, jak również ich standardzie, obsługę klienta realizuje też telefonicznie. Z informacji otrzymanych od recepcjonistki dowiedzieliśmy się, że pensjonat ma w standardzie trzy rodzaje pokoi, jak również opcję wyżywienia, która jest uzależniona od wybranego standardu przez klienta. Pierwszym standardem jest opcja "Oszczędna" (tzw. żydowska), która jest najtańsza i ma podstawowe wyposażenie w postaci telewizora, podstawowego pakietu telewizji naziemnej, łazienki nie znajdują się w pokojach, lecz są zbiorcze na końcu korytarzy, pokoje są cztero i pięcioosobowe, ta opcja nie posiada możliwości posiłków. Kolejnym standardem jest opcja "Bed&Breakfast”, są to pokoje trzy i czteroosobowe, w pokojach znajduje się osobna łazienka, telewizory w tych pokojach mają pełny pakiet telewizji kablowej, łącznie z pakietami premium tj. ("Canal+, HBO, Cinemax itp.), w tej opcji klienci pensjonatu mają zagwarantowane śniadanie w postaci kawy i herbaty, jak również słodkich Croissant-ów. Najwyższym i jednocześnie najdroższym standardem jest opcja Exclusive, w której klient ma dostęp do bogatego rodzaju śniadań jak również obiadokolacji. Pokoje są maksymalnie dwuosobowe, a także mają największą powierzchnię pomiędzy 15-20 metrów kwadratowych, w pokojach znajdują się bogato wykończone łazienki z jacuzzi. Recepcjonistka sprawdza dostępność wybranego pokoju w danej chwili, jak również jego dostępność w przyszłości. Czy nie doszło już do jego rezerwacji w przyszłych terminach. Gdy okaże się, że wybrane przez klienta pokoje są dostępne przekazuje te informacje do niego. Jeśli klient jest zdecydowany - wtedy recepcjonistka dokonuje rezerwacji wybranego pokoju. Aby dokonać rezerwacji pokoju w późniejszym terminie, klient musi wpłacić zaliczkę w wysokości minimum 30% wartości całej rezerwacji maksymalnie w ciągu 3 dni roboczych, jeżeli rezerwacja odbywa się telefonicznie. Jeżeli środki w tym terminie nie wpłyną to rezerwacja jest anulowana. Po dokonaniu rezerwacji, klient otrzymuje od recepcji potwierdzenie dokonania rezerwacji jak również klucze do wybranego pokoju w przypadku, jeżeli klient wybrał pokój na miejscu w danym czasie. Potwierdzenie rezerwacji dostaje w postaci dowodu kasowego.

Ważnym elementem odpowiedzialnym za sprawne funkcjonowanie pensjonatu jest serwis sprzątający, który ma za zadanie przynajmniej raz w ciągu dnia posprzątać użytkowane w danej chwili pokoje. Projektowany system nie przewiduje jednak żadnych funkcjonalności dla tego personelu.

Za sprawne działanie całego pensjonatu odpowiada Manager. Do czynności, którymi się zajmuje, należy bieżąca kontrola i uzupełnianie stanu magazynowego pensjonatu, w którym przechowywane są materiały niezbędne do sprawnego działania ośrodka. Do tych materiałów należą środki czystości, pościele, i oczywiście żywność, która jest zamawiana przez kucharza pensjonatu a manager musi za to wszystko zapłacić. Manager jest też odpowiedzialny za obsługę klientów w przypadku wystąpienia jakichkolwiek niejasności, reklamacji ze strony klientów. Wtedy osobiście nadzoruje kontakt z klientem w celu jak najsprawniejszego rozwiązania zgłoszonej reklamacji ze strony klienta.

Następną ważną osobą odpowiedzialną za działanie pensjonatu jest kucharz, który jest osobą z wieloletnim doświadczeniem kucharskim i przygotowaną na spełnianie różnych zachcianek klientów pensjonatu. Na co dzień zajmuje się realizacją zleceń wystawionych mu przez recepcje, gdyż pensjonat posiada możliwość wykupienia standardu "Bed&Breakfast" jak również opcji "Exclusive", która jest połączona z pakietem wyżywienia "Full-Board". W przypadku wybrania przez klienta opcji noclegu z wyżywieniem, kucharz zajmuje się przygotowaniem posiłków dla klientów pensjonatu, jak również bezpośrednią obsługą klientów podczas spożywania posiłków.

Analizując działanie pensjonatu trzeba koniecznie wspomnieć o kliencie, który tak naprawdę jest osobą kluczową w działania pensjonatu. Klient jest osobą fizyczną lub firmą, która ma możliwość oglądania pokoi według różnych kategorii. Może zadawać pytania recepcji odnośnie oferowanych pokoi. Ponadto Klient może dokonywać rezerwacji pokoi, zmieniać rodzaj rezerwacji, anulować rezerwacje. Również klient może reklamować zakupioną usługę, jak również składać uwagi odnośnie działania pensjonatu, poza tym może rezerwować pokoje, które w danej chwili nie są dostępne z powodu rezerwacji a będą w przyszłości.

### Założenia ogólne systemu.

Słownik pojęć:

* **Klient** – osoba lub firma korzystająca z oferty pensjonatu
* **Recepcja** – jest to osoba odpowiedzialna za bezpośredni kontakt z klientem oraz pozostałymi pracownikami w pensjonacie
* **Serwis sprzątający** – jest to grupa osób odpowiedzialna za czystość w pokojach pensjonatu, uzupełniają na bieżąco stan pokoi, bezpośrednio komunikują się z recepcją
* **Manager -** jest to osoba odpowiedzialna za obsługę stanu magazynowego, aktualizuje jego zawartość jak również rozbudowuje o nowe elementy, planuje budżet firmy, komunikuje się ze Recepcją, Kucharzem, jak również z Klientem w przypadku reklamacji
* **Rezerwacja** – jest to zamówienie złożone przez klienta
* **Pokój** – towar handlowy, który jest oferowany przez pensjonat
* **Dokument kasowy** – rozliczenie potwierdzające dokonaną rezerwacje

Założenia:

* W bazie danych znajdują się podstawowe informacje o klientach (imię, nazwisko lub nazwa firmy i NIP, telefon, e-mail)
* W bazie danych znajdują się podstawowe dane o pokojach (standard wyposażenia, cena wynajmu pokoju, termin dostępności danego pokoju)
* W bazie danych znajdują się informacje o przeprowadzonych transakcjach (pracownik przeprowadzający transakcje, wybrany pokój, koszt, data złożenia zamówienia na rezerwacje, data wydania pokoju dla klienta)
* W bazie danych znajdują się informacje o dokumentach kasowych (typ dokumentu, wartość dokumentu, nr zlecenia, rodzaj klienta (osoba prywatna lub firma))
* W bazie danych znajdują się informacje o zleceniach (klient, pokój, data złożenia zamówienia, data realizacji zamówienia)
* Klient, który zgłasza się do pensjonatu zgłasza zapotrzebowanie na pokój i termin rezerwacji pokoju.
* Pokój należy do danego klienta, klient może złożyć zapotrzebowanie na wiele pokoi z różnych standardów (oszczędna, bed&breakfast, exclusive).
* Recepcja w pensjonacie może zrealizować wiele zleceń, pojedyncze zlecenie może być zrealizowane przez wielu pracowników pensjonatu.
* Wiele pokoi może być wykorzystanych w celu realizacji danego zlecenia.
* Klient, który zamawia pokój, otrzymuje informacje odnośnie kosztu zamówionej usługi
* Dokument kasowy jest to paragon lub faktura, która jest wystawiana na osobę fizyczną lub na firmę
* Recepcja na podstawie zakończonego zlecenia może wystawić jeden dokument kasowy
* Klient może otrzymać kilka dokumentów kasowych z różnych zleceń
* Uproszczenie (uogólnienie):
* Marża za wynajem pokoi będzie stała, bez względu na inne czynniki (np. inflacja)
* Cena pokoju w recepcji nie będzie się zmieniała w czasie, chyba że kończy się okres planowanych rezerwacji, wtedy stosowane będą z góry ustalone rabaty w wysokości 10%, 20% i 30% aby ograniczyć straty finansowe związane z tzw. przestojem, czyli brakiem wynajmu pokoi w danym czasie.

## Specyfikacja funkcjonalna

### Diagram hierarchii funkcji (FHD)

Dla zobrazowania najważniejszych funkcji systemu, został wykorzystany diagram hierarchii funkcji (FHD). Istotą tej metody jest opracowanie i utworzenie zhierarchizowanego modelu funkcji, poprzez proces dekomponowania funkcji nadrzędnych na podrzędne. Funkcje podrzędne również mogą być potem zdekomponowane w celu uzyskania bardziej szczegółowego modelu funkcji. Diagram FHD powstał poprzez przeanalizowanie opisanych przypadków i scenariuszy użycia.



Rysunek 1. Diagram hierarchii funkcji (FHD) – poziom 1-2. Najważniejsze funkcje systemu.

### Lista funkcji z opisem ich zastosowania

W formie opisowej zostały określone przypadki oraz scenariusze użycia, na podstawie których został opracowany diagram hierarchii funkcji (FHD). Przedstawione poniżej funkcjonalności systemu uwzględniają każdego z aktorów.

Zarządzanie użytkownikami (Administrator):

1. **Rejestracja użytkownika** – użytkownik o uprawnieniach administratora wchodzi do panelu administracji, wybiera funkcję zarejestruj użytkownika (funkcja dostępna tylko i wyłącznie dla użytkownika z uprawnieniami administratora). Podane muszą zostać dane: imię, nazwisko, pesel, telefon, email, hasło oraz poziom uprawnień.
2. **Edycja danych użytkownika** – użytkownik o uprawnieniach administratora wchodzi do panelu administracji, wybiera funkcję użytkownicy. Po wybraniu tej funkcji pojawi się lista wszystkich użytkowników, którzy są zarejestrowani w systemie. Użytkownik o uprawnieniach administratora może zmienić dane któregokolwiek z użytkowników systemu oraz poziom uprawnień.
3. **Usuwanie użytkownika** – użytkownik o uprawnieniach administratora wchodzi do panelu administracji, wybiera funkcję usuń użytkownika. Po wybraniu tej funkcji pojawi się lista wszystkich użytkowników zarejestrowanych w systemie. Aby usunąć użytkownika, należy zaznaczyć danego użytkownika na liście, i zatwierdzić przyciskiem ok. Po tej operacji powinno wyskoczyć okno z pytaniem „Czy na pewno chcesz usunąć tego użytkownika?” z dwoma opcjami do wyboru: tak, nie.

Rezerwacja apartamentów:

1. **Składanie rezerwacji** – klient wchodzi do panelu rezerwacji apartamentów, wybiera funkcję złóż rezerwację. Wybrana musi zostać data rezerwacji, standard apartamentu, konkretny pokój. Podane muszą zostać również dane: imię, nazwisko lub nazwa firmy, PESEL lub NIP, telefon, e-mail. Złożenie rezerwacji klient potwierdza przyciskiem złóż rezerwację. Po wykonaniu tej operacji powinien pojawić się numer zlecenia, które może posłużyć do zmiany rezerwacji bądź odwołania jej.
2. **Zmiana rezerwacji** – klient wchodzi do panelu rezerwacji apartamentów, wybiera funkcję zmień rezerwację. W pole tekstowe należy wpisać numer zlecenia, który został podany po złożeniu rezerwacji, po czym zatwierdzić przyciskiem szukaj. Klient może zmienić datę rezerwacji, o ile jest dostępna taka możliwość, standard apartamentu jak i konkretny pokój.
3. **Anulowanie rezerwacji** – klient wchodzi do panelu rezerwacji apartamentów, wybiera funkcję anuluj rezerwację. W pole tekstowe należy wpisać numer zlecenia, który został podany po złożeniu rezerwacji, po czym zatwierdzić przyciskiem anuluj. Po wykonaniu tej operacji powinien pokazać się podgląd rezerwacji. Pod podglądem rezerwacji klient zatwierdza anulowanie rezerwacji przyciskiem anuluj bądź wraca do panelu rezerwacji pokojów przyciskiem wróć.
4. **Reklamacja rezerwacji** – klient wchodzi do panelu rezerwacji apartamentów, wybiera funkcję złóż reklamację. W osobnych polach tekstowych muszą zostać podane: numer zlecenia, kategoria reklamacji (wybierana z listy np. czystość), treść reklamacji. Klient zatwierdza wysłanie reklamacji przyciskiem wyślij.

Zarządzanie rezerwacjami i dokumentami kasowymi:

1. **Przyjmowanie rezerwacji** – użytkownik systemu o uprawnieniach do przyjmowania rezerwacji, pracownik recepcji, wchodzi do panelu zarządzania rezerwacjami i dokumentami kasowymi. W panelu powinna się pojawić lista złożonych rezerwacji przez klientów. Użytkownik ma możliwość podglądu rezerwacji poprzez wybór zlecenia i zatwierdzenie przyciskiem pokaż. Pod podglądem rezerwacji użytkownik ma możliwość potwierdzenia rezerwacji przyciskiem potwierdź rezerwację lub wrócić do listy złożonych rezerwacji przyciskiem wróć.
2. **Podgląd rezerwacji** – użytkownik systemu o uprawnieniach do przyjmowania rezerwacji, pracownik recepcji, wchodzi do panelu zarządzania rezerwacjami i dokumentami kasowymi. W panelu powinna się pojawić lista złożonych rezerwacji przez klientów. Użytkownik ma możliwość podglądu rezerwacji poprzez wybór zlecenia i zatwierdzenie przyciskiem pokaż.
3. **Anulowanie rezerwacji** – użytkownik systemu o uprawnieniach do przyjmowania rezerwacji, pracownik recepcji, wchodzi do panelu zarządzania rezerwacjami i dokumentami kasowymi. W panelu, powinna się pojawić lista złożonych rezerwacji przez klientów. Użytkownik ma możliwość podglądu rezerwacji poprzez wybór zlecenia i zatwierdzenie przyciskiem pokaż. Pod podglądem rezerwacji użytkownik ma możliwość anulowania rezerwacji przyciskiem anuluj rezerwację.
4. **Wystawianie dokumentów kasowych** – użytkownik systemu o uprawnieniach do zarządzania dokumentami kasowymi, recepcja, wchodzi do panelu zarządzania rezerwacjami i dokumentami kasowymi. W panelu powinna się pojawić lista złożonych rezerwacji przez klientów. Użytkownik ma możliwość podglądu rezerwacji poprzez wybór zlecenia i zatwierdzenie przyciskiem pokaż. Pod podglądem rezerwacji użytkownik ma możliwość wystawienia dokumentu kasowego przyciskiem wystaw dokument kasowy lub wrócić do listy złożonych rezerwacji przyciskiem wróć.
5. **Podgląd dokumentów kasowych** – użytkownik systemu o uprawnieniach do zarządzania dokumentami kasowymi, recepcja, wchodzi do panelu zarządzania rezerwacjami i dokumentami kasowymi. W panelu powinna pojawić się lista złożonych rezerwacji przez klientów. Użytkownik ma możliwość podglądu rezerwacji poprzez wybór zlecenia i zatwierdzenie przyciskiem pokaż. Pod podglądem rezerwacji użytkownik ma możliwość podglądu dokumentu kasowego przyciskiem pokaż dokument kasowy (jeśli do danej rezerwacji został wystawiony) lub wrócić do listy złożonych rezerwacji przyciskiem wróć.

Zarządzanie reklamacjami:

1. **Podgląd reklamacji** – użytkownik o uprawnieniach do zarządzania reklamacjami, manager, wchodzi do panelu zarządzania reklamacjami. Zostaje wyświetlona lista reklamacji. Użytkownik ma możliwość podglądu reklamacji poprzez wybór reklamacji i zatwierdzenie przyciskiem pokaż.
2. **Uwzględnianie reklamacji** – użytkownik o uprawnieniach do zarządzania reklamacjami, manager, wchodzi do panelu zarządzania reklamacjami. Zostaje wyświetlona lista reklamacji. Użytkownik wybiera reklamację z listy i zatwierdza przyciskiem pokaż. Pod podglądem zlecenia, użytkownik ma możliwość uwzględnienia reklamacji przyciskiem uwzględnij, nieuwzględnienia reklamacji przyciskiem nie uwzględniaj lub powrotu do listy reklamacji przyciskiem wróć. Po wykonaniu funkcji uwzględnij, pojawi się pole tekstowe, w którym użytkownik musi napisać komentarz do reklamacji, które trafi na adres e-mail klienta. Zatwierdza przyciskiem wyślij. Po wykonaniu funkcji nie uwzględniaj, pojawi się pole tekstowe, w którym użytkownik musi napisać komentarz do reklamacji, które trafi na adres e-mail klienta. Zatwierdza przyciskiem wyślij.

Zarządzanie kontem:

1. **Rejestracja** – niezalogowany użytkownik systemu wybiera przycisk rejestracja. W osobnych polach tekstowych należy wpisać dane: imię, nazwisko lub nazwa firmy, PESEL lub NIP, telefon, e-mail. W kolejnych polach tekstowych należy wpisać hasło i je powtórzyć. Hasło będzie służyło do logowania się użytkownika do systemu. Użytkownik potwierdza rejestrację przyciskiem rejestruj.
2. **Logowanie** – niezalogowany użytkownik systemu wybiera przycisk logowanie. W osobnych polach tekstowych należy wpisać swój login i hasło. Funkcja logowania następuje po przyciśnięciu przycisku zaloguj.
3. **Edycja danych konta** – zalogowany użytkownik systemu wchodzi do panelu zarządzania kontem. W panelu powinny pojawić dane konta. Użytkownik ma możliwość edycji danych konta poprzez kliknięcie przycisku edytuj dane konta. W odpowiednich polach zostaną podstawione dotychczasowe dane konta, które można zmienić. Po wpisaniu nowych danych, użytkownik zatwierdza funkcję edycji danych konta przyciskiem zatwierdź lub wraca do panelu zarządzania kontem przyciskiem wróć.
4. **Zmiana hasła** – zalogowany użytkownik systemu wchodzi do panelu zarządzania kontem. W panelu powinny pojawić się dane konta. Użytkownik ma możliwość zmiany hasła poprzez kliknięcie przycisku zmień hasło. W polach tekstowych należy wpisać odpowiednio: stare hasło, nowe hasło, powtórzyć nowe hasło. Po wpisaniu danych użytkownik zatwierdza operację zmiany hasła przyciskiem zatwierdź lub wraca do panelu zarządzania kontem przyciskiem wróć.

### Diagramy przypadków użycia

Kolejnym etapem przedstawiania wymagań funkcjonalnych jest diagram przypadków użycia.

Istotą tego diagramu jest stworzenie modelu systemu z punktu widzenia poszczególnych klas poprzez identyfikację funkcji istotnych dla poszczególnych klas. Diagram został wykonany na podstawie analizy opisanych przypadków i scenariuszy użycia.



Rysunek 2. Diagram przypadków użycia (DPU) Systemu wspierającego wystawianie/rezerwacji apartamentów pensjonatu "Sleep Well"

## Model danych

Do przedstawienia modelu projektowanego systemu użyto notacji strukturalnych, a w szczególności diagramów związków encji (ang. ERD), ponieważ są one ukierunkowane na modelowanie struktury relacyjnych baz danych.

Pod pojęciem encji należy rozumień zbiór obiektów posiadających tożsamość i stan. Stan obiektów opisywany jest przez atrybuty, z których składa się encja. Jeśli któryś z występujących w niej atrybutów w sposób jednoznaczny i unikalny identyfikuje wystąpienie encji, taki atrybut nazywa się kluczem podstawowym.

Przed sporządzeniem ERD należy najpierw wyodrębnić z modelu istotne dane (encje) oraz związki między nimi. Związek oznacza tutaj zbiór powiązań między jednym lub większą liczbą zbiorów encji, które uczestniczą w danym związku. W modelu projektowanego systemu można wyróżnić następujące encje:

* użytkownik,
* pokój,
* rezerwacja,
* reklamacja,
* dokument kasowy,
* prawa użytkownika,
* stan rezerwacji.

### Tabela z opisem encji oraz związków

W tabeli poniżej przedstawiono krótkie opisy każdej z encji wraz z nazwami używanymi na diagramie ER.

| Nazwa encji | Opis | Nazwa używana w ERD |
| --- | --- | --- |
| Użytkownik | Opisuje wszystkich użytkowników systemu i ich dane personalne, a także zawiera informacje o typie klienta (firma lub osoba prywatna). | użytkownik |
| Pokój | Opisuje wszystkie pokoje dostępne w pensjonacie. | pokój |
| Rezerwacja | Opisuje wszystkie rezerwacje (wskazuje klienta, cenę, termin, liczbę osób itp.) | rezerwacja |
| Reklamacja | Opisuje wszystkie złożone reklamacje. | reklamacja |
| Dokument kasowy | Opisuje wszystkie wydane dokumenty kasowe. | dokument kasowy |
| Prawa (użytkownika) | Opisuje prawa wszystkich użytkowników, czyli role (np. klient, manager). | rola |
| Stan rezerwacji | Opisuje możliwe stany rezerwacji (wskazuje czy jest ona opłacona, zaakceptowana itp.). | stan |

Każdy występujący między encjami związek ma nadaną nazwę, opisującą jego funkcję. Zdefiniowano następujące związki:

* ma,
* zarządza,
* zawiera,
* tworzy,
* dotyczy.

### Diagram związków encji (ERD)



## Model architektury systemu

Pisanie aplikacji działających na serwerze wymaga wzięcia pod uwagę wiele czynników, przede wszystkim w środowisku wielozadaniowym liczą się:

* Niezawodność;
* Bezpieczeństwo;
* Wydajność.

W naszym przypadku klientami, zainteresowanymi aplikacją są pensjonaty oraz hotele. Znając zakres działalności tych firm, staje się jasne, jak trudno napisać dobrą aplikację, spełniającą wcześniej wymienione czynniki.

Architektura systemu rezerwacji pokoi pensjonatu jest bazowana na metodzie trójpoziomowej.

Możemy wyróżnić w niej trzy poziomy:

* Warstwa prezentacji
* Warstwa logiczna
* Warstwa danych

Warstwa prezentacji: zawiera komponenty odpowiedzialne za wygląd aplikacji na monitorze klienta.   
Warstwa logiczna: zawiera komponenty zajmujące się rozwiązywaniem konkretnych problemów programistycznych.   
Warstwa danych: warstwa wykorzystywana przez warstwę logiczną do zachowywania informacji.

W naszym przypadku podział aplikacji na warstwy pozwala na odizolowanie każdej z nich, tak aby zmiany dokonywane w jednej z nich miały jak najmniejszy wpływ na pozostałe.

### Opis diagramu architektury systemu w postaci tabeli

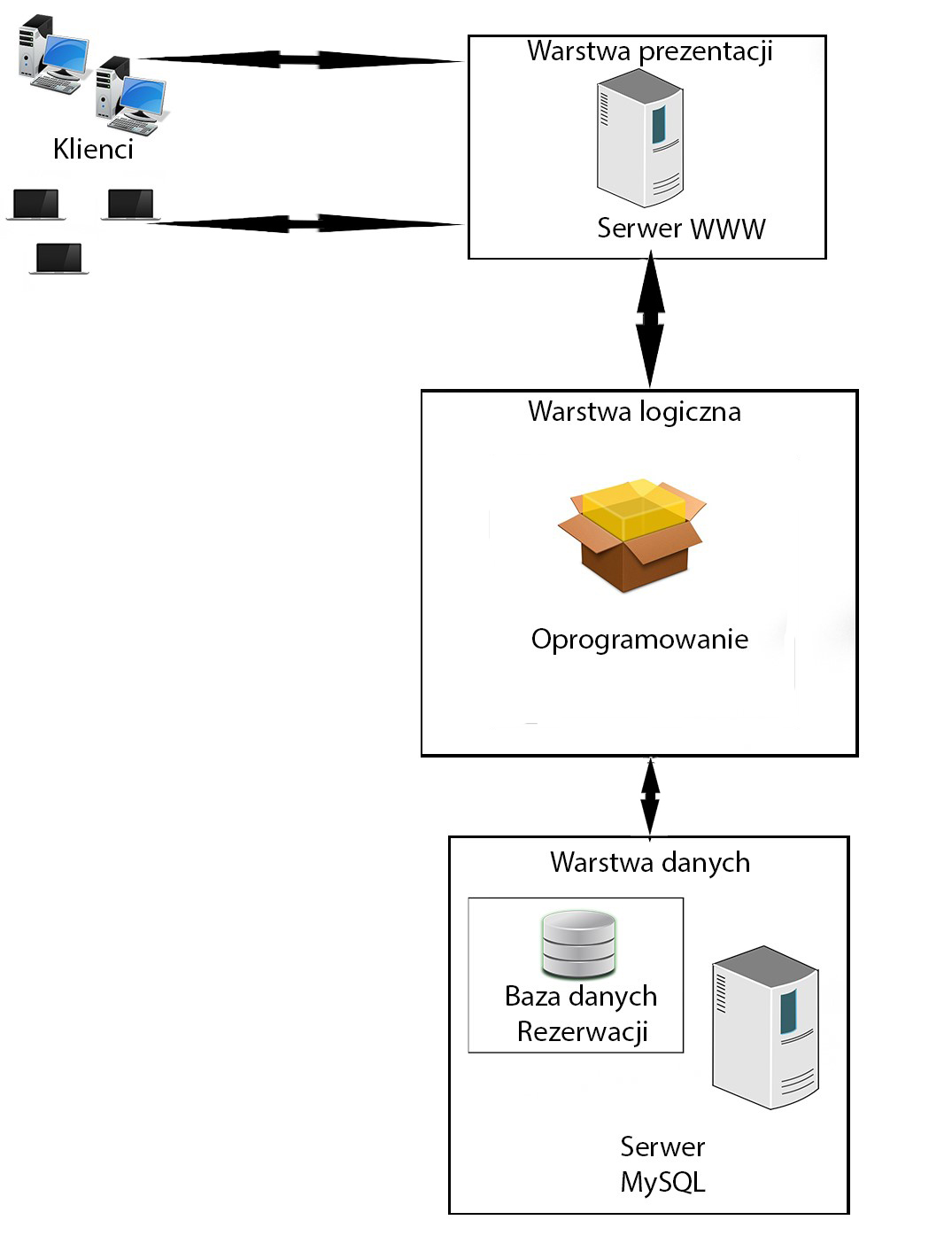
|  |  |
| --- | --- |
| Nazwa | Opis |
| **Klienci** | Opisuje wszystkich użytkowników systemu, którzy łączą się do oprogramowania przez serwer www w celu rezerwacji pokoju. |
| **Serwer WWW** | Opisuje serwer, na którym przechowywana jest zawartość widoczna dla użytkowników. |
| **Oprogramowanie** | Opisuje całkowite działanie oprogramowania. |
| **Baza danych** | Opisuje bazę danych z którą połączone jest oprogramowanie. Służy do przechowywania danych. |
| **Serwer MySQL** | Opisuje serwer, na którym przechowywane są bazy danych. |

### Uzasadnienie wybranego modelu

Model ten został wybrany ze względu na zalety projektowania aplikacji trójpoziomowych:

* Redukcja kosztów kupna i instalacji, sterowniki baz danych i jądro aplikacji jest instalowane po stronie serwera.
* Aktualizacja systemu baz danych lub warstwy logicznej jest znacznie uproszczona. Klient nie korzysta bezpośrednio z bazy danych co pozwala na migrację i zmianę bazy danych bez wiedzy, i potrzeby zmiany interfejsu po stronie klienta.
* Zasoby serwera mogą być udostępniane przez tzw. pulę zasobów. Pula zasobów jest mechanizmem wydatnie zwiększającym wydajność aplikacji.
* Zastosowanie jej do baz danych zmniejsza ilość jednoczesnych połączeń z motorem bazy i pozwala zmniejszyć ilość licencji dla klientów.
* Każda warstwę można zmieniać niezależnie od pozostałych.
* Proste monitorowanie działania systemu. Łatwo zlokalizować wystąpienie błędów krytycznych lub tzw. wąskich gardeł. Obciążenie jednej z warstw nie wpływa na pozostałe.

### Diagram architektury systemu



# Projekt systemu

## Projekt architektury systemu

Architektura określa podział oprogramowania na komponenty oraz definiuje funkcje tych komponentów i występujące między nimi relacje.

Opis architektury oprogramowania pełni w projekcie rolę schematu jego budowy.

Architektura to zbiór znaczących decyzji dotyczących:

* organizacji systemu komputerowego
* wyboru elementów strukturalnych i ich interfejsów, z których system jest zbudowany
* zachowania tych elementów opisanego w kooperacjach
* składania elementów strukturalnych i czynnościowych w coraz to większe podsystemy
* stylu architektonicznego, według którego tworzy się konstrukcję
* systemu, to znaczy charakterystycznych elementów statycznych i dynamicznych oraz ich interfejsów, kooperacji i składania.

### Schemat i diagram warstw

**Warstwa prezentacji** - zawiera komponenty interfejsu użytkownika (okna, strony etc.)

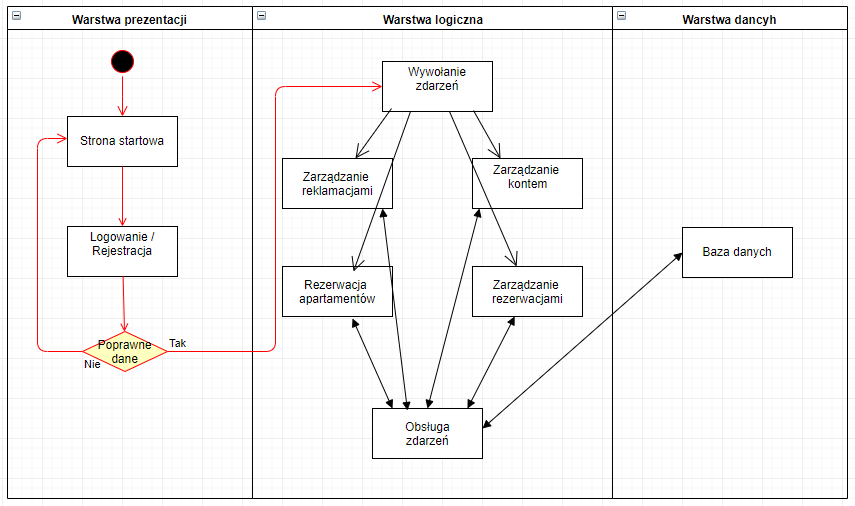
* Umożliwia użytkownikowi wydawanie poleceń dla systemu i wprowadzanie danych. • Prezentuje dla użytkownika dane przetwarzane przez warstwę logiki aplikacji. • Typowa implementacja – HTML
* Może być połączona w jeden pakiet z logiką prezentacji (specjalna aplikacja po stronie klienta).

**Warstwa logiczna**- zawiera funkcje obsługi interfejsu użytkownika.

* Umożliwia obsługę zdarzeń w formularzach – zmianę stanu pewnych kontrolek po zmianie stanu innych kontrolek (np. wypełnienie zbioru pól „adres dostawy” na podstawie pól „adres zamieszkania” przy kliknięciu przycisku „dostawa na adres zamieszkania”).
* Zapewnia walidację wprowadzanych danych i powstrzymuje klienta przed wysłaniem niewypełnionych formularzy.
* Tłumaczy żądania z HTTP na wywołania procedur w warstwie logiki aplikacji.
* Generuje odpowiedzi HTTP (strony HTML) na podstawie wyników z warstwy aplikacji.

**Warstwa danych** - zawiera komponenty odpowiedzialne za przechowywanie zapisywanie i odczytywanie danych.

* bazy danych (tabele, kwerendy)
* repozytoria plików
* jaki sposób przechowywania będzie stosowany (baza danych, pliki)
* jaki typ bazy danych będzie stosowany
* schematu bazy danych
* Typowa implementacja: serwer SQL



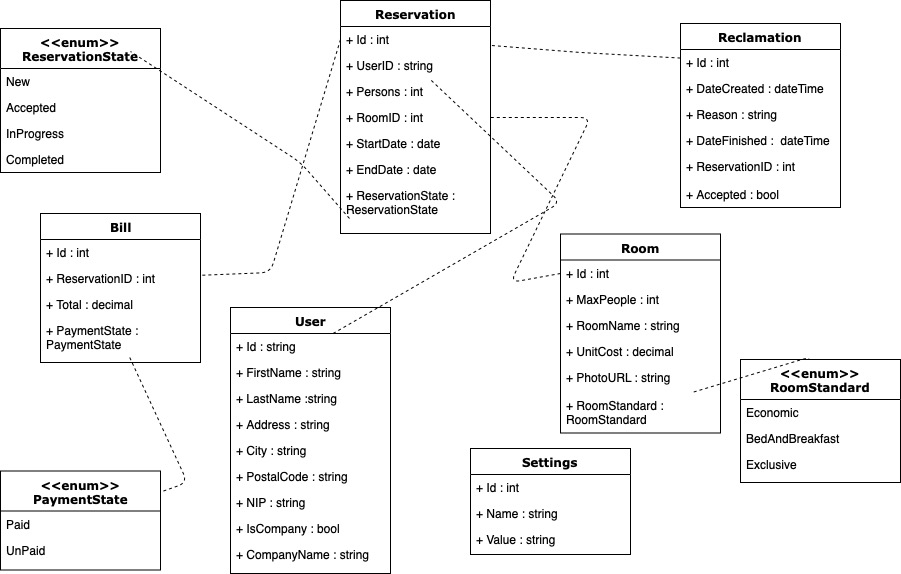
### Diagram klas

Diagram klas statyczny, koncepcyjny widok projektowanej aplikacji. Prezentuje strukturę klas i relacji między nimi. Jest podstawowym diagramem struktury logicznej systemu

Zawiera informacje o statycznych związkach między elementami, klasy są ściśle powiązane z technikami programowania zorientowanego obiektowo.

Klasy związane z systemem logowania zostały pominięte, gdyż są tworzone przez framework automatycznie.

**Klasy modelu**



**Klasa Bill** modeluje rachunki rozliczane w systemie rezerwacji apartamentów, wspiera ona funkcjonowanie serwisu w kierunku rozliczeń kasowych.

**Klasa User** modeluje główne informacje dotyczące użytkowników systemu rezerwacji w tym informacje kontaktowe oraz dane osobowe.

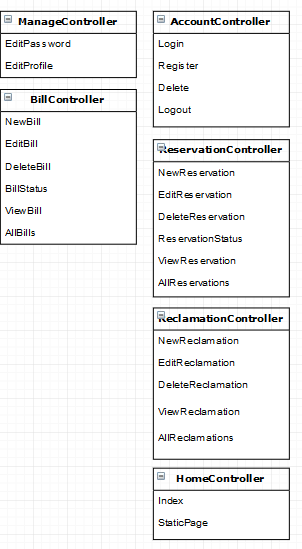
**Klasa Reservation** modeluje postać rezerwacji przeprowadzanych w systemie, zbieranie głównych informacji niezbędnych do przeprowadzenia prawidłowego procesu rezerwacji pokoju. Określa czas oraz stan rezerwacji danego pokoju.

**Klasa Reclamation** modeluje informacje dotyczące reklamacji przeprowadzonych przez użytkowników oraz ich statusu.

**Klasa Room** modeluje informacje dotyczące pokoi w pensjonacie wraz z ilością miejsc i standardem.

**Klasa Settings** modeluje postać globalnych ustawień systemu.

**Kontrolery z akcjami**



## Projekt bazy danych

### Tabele bazy danych

Poniżej przedstawiono wykaz tabel projektowanej bazy danych wraz z ogólnym opisem zawartości oraz klasą modelu, której odpowiada każda z nich.

|  |  |  |
| --- | --- | --- |
| Nazwa tabeli | Opis | Klasa modelu |
| Reservations | Tabela zawierająca wszystkie zlecenia rezerwacji pokoi i informacje o nich. | Reservation |
| Users | Tabela mechanizmów ASP.NET Identity zawierająca dane o użytkownikach systemu. | Domyślna klasa ASP.NET Identity |
| Reclamations | Tabela zawierająca wszystkie reklamacje klientów hostelu i informacje o nich. | Reclamation |
| Bills | Tabela zawierająca wszystkie dane związane z płatnościami i informacje o nich. | Bill |
| Rooms | Tabela zawierająca wszystkie dane pokoi dostępnych w pensjonacie. | Room |
| Roles | Tabela mechanizmów ASP.NET Identity zawierająca możliwe role użytkowników systemu. | Domyślna klasa ASP.NET Identity |
| UserClaims | Tabela mechanizmów ASP.NET Identity zawierająca restrykcje, które dodatkowo potwierdzają tożsamość użytkownika. | Domyślna klasa ASP.NET Identity |
| UserLogins | Tabela mechanizmów ASP.NET Identity zawierająca dane o zewnętrznych logowaniach użytkowników (np. poprzez konto Facebook lub Google). | Domyślna klasa ASP.NET Identity |
| UserRoles | Tabela mechanizmów ASP.NET Identity zawierająca przyporządkowania użytkowników systemu do ról. | Domyślna klasa ASP.NET Identity |
| Settings | Tabela zawierająca ustawienia systemu. | Setting |

### Pola bazy danych

Z uwagi na podejście Code First, baza danych definiowana jest za pomocą modelu w klasach aplikacji. W tabelach poniżej przedstawiono nazwy pól każdej z projektowanych tabel oraz typ ich wartości odpowiadający typom w języku C#, a także przewidziane restrykcje.

Wartości będące kluczami podstawowymi oznaczono **pogrubieniem**, a kluczami obcymi *pochyleniem*.

#### Tabela Reservations

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa pola | Typ wartości | Restrykcje | Komentarz |
| **ReservationId** | Int | Wymagane |  |
| *UserId* | String | Wymagane |  |
| *RoomId* | int | Wymagane |  |
| Persons | int | Wymagane | Liczba osób |
| StartDate | DateTime | Wymagane | Data rozpoczęcia zaplanowanego noclegu |
| EndDate | DateTime | Wymagane | Data zakończenia noclegu |
| ReservationState | ReservationState [Enum] |  | Wartość enumeryczna określająca stan rezerwacji |

#### Tabela Reclamations

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa pola | Typ wartości | Restrykcje | Komentarz |
| **ReclamationId** | Int | Wymagane |  |
| *ReservationId* | Int | Wymagane |  |
| Reason | String | Wymagane | Treść reklamacji |
| DateCreated | DateTime |  | Data stworzenia reklamacji |
| DateFinished | DateTime |  | Data rozpatrzenia reklamacji |
| Accepted | bool |  | Reklamacja zaakceptowana lub nie |

#### Tabela Bills

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa pola | Typ wartości | Restrykcje | Komentarz |
| **BillId** | Int | Wymagane |  |
| *ReservationId* | Int | Wymagane |  |
| Total | decimal |  | Suma rachunku |
| PaymentSate | PaymentState [Enum] |  | Wartość enumeryczna określająca stan rozliczenia klienta z pensjonatem |

#### Tabela Rooms

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa pola | Typ wartości | Restrykcje | Komentarz |
| **RoomId** | Int | Wymagane |  |
| MaxPeople | Int | Wymagane | Maksymalna liczba osób w pokoju |
| RoomName | string |  | Nazwa pokoju |
| UnitCost | decimal |  | Koszt jednej nocy za osobę |
| PhotoUrl | string |  | Link do zdjęcia pokoju |
| RoomStandard | RoomStandard [Enum] |  | Wartość enumeryczna określająca standard pokoju |

#### Tabela Users

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa pola | Typ wartości | Restrykcje | Komentarz |
| **Id** | String | Wymagane |  |
| FirstName | String |  | Imię |
| LastName | String |  | Nazwisko |
| Address | String |  | Adres |
| City | String |  | Miasto |
| PostalCode | String |  | Kod pocztowy |
| NIP | String |  | Numer NIP w przypadku firmy |
| IsCompany | Bool |  | Czy użytkownik jest firmą |
| CompanyName | String |  | Nazwa firmy |
| Email | String |  | Adres email użytkownika |
| EmailConfirmed | Bool | Wymagane | Adres email potwierdzony |
| PasswordHash | String |  | Zaszyfrowane hasło |
| SecurityStamp | String |  | Pieczęć szyfrująca |
| PhoneNumber | String |  | Numer telefonu |
| PhoneNumberConfirmed | Bool | Wymagane | Numer telefonu potwierdzony |
| TwoFactorEnabled | Bool | Wymagane | Dwustopniowe logowanie |
| LockoutEndDateUtc | DateTime |  | Data zakończenia blokady konta |
| LockoutEnabled | Bool | Wymagane | Konto zablokowane |
| AccessFailedCount | Int | Wymagane | Liczba nieudanych logowań |
| UserName | String | Wymagane | Adres email lub login |

#### Tabela Roles

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa pola | Typ wartości | Restrykcje | Komentarz |
| **Id** | String | Wymagane |  |
| Name | String | Wymagane | Nazwa roli |

#### Tabela UserClaims

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa pola | Typ wartości | Restrykcje | Komentarz |
| **Id** | Int | Wymagane |  |
| *UserId* | String | Wymagane |  |
| ClaimType | String |  | Typ oświadczenia |
| ClaimValue | String |  | Wartość oświadczenia |

#### Tabela UserLogins

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa pola | Typ wartości | Restrykcje | Komentarz |
| **LoginProvider** | String | Wymagane | Dostawca logowania |
| **ProviderKey** | String | Wymagane | Klucz dostawcy logowania |
| ***UserId*** | String | Wymagane |  |

#### Tabela UserRoles

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa pola | Typ wartości | Restrykcje | Komentarz |
| ***UserId*** | String | Wymagane |  |
| ***RoleId*** | String | Wymagane |  |

#### Tabela Settings

|  |  |  |  |
| --- | --- | --- | --- |
| Nazwa pola | Typ wartości | Restrykcje | Komentarz |
| **SettingId** | Int | Wymagane |  |
| Name | String | Wymagane | Nazwa ustawienia |
| Value | String | Wymagane | Wartość ustawienia |

### Projekt schematu relacji bazy danych.



### Mechanizmy bazy danych.

W obecnej chwili nie będą miały zastosowanie żadne procedury i wyzwalacze. Wszelkie modyfikacje struktur danych będą zachodziły poprzez kod aplikacji. Struktura bazy danych będzie aktualizowana do najnowszej wersji za pomocą mechanizmu migracji, który wbudowany jest w Entity Framework.

## Projekt algorytmów

### Diagramy czynności

#### Diagram czynności dla funkcji: Zarządzanie kontem



Rysunek 3. Diagram czynności dla funkcji Zarządzanie kontem

Poniżej przedstawiono dekompozycję czynności: Rejestracja



Rysunek 4. Diagram czynności - dekompozycja czynności: „Rejestracja”

Poniżej przedstawiono dekompozycję czynności: Logowanie



Rysunek 5. Diagram czynności - dekompozycja czynności: „Logowanie”

Poniżej przedstawiono dekompozycję czynności: Edycja danych konta



Rysunek 6. Diagram czynności - dekompozycja czynności: „Edycja danych konta”

Poniżej przedstawiono dekompozycję czynności: Zmiana hasła



Rysunek 7. Diagram czynności - dekompozycja czynności: „Zmiana hasła”

#### Diagram czynności dla funkcji: Rezerwacja apartamentów



Rysunek 8. Diagram czynności - Rezerwacja apartamentów

Poniżej przedstawiono dekompozycję czynności: Zmiana rezerwacji



Rysunek 9. Diagram czynności - dekompozycja czynności: „Zmiana rezerwacji”

Poniżej przedstawiono dekompozycję czynności: Anulowanie rezerwacji



Rysunek 10. Diagram czynności - dekompozycja czynności: „Anulowanie rezerwacji”

Poniżej przedstawiono dekompozycję czynności: Reklamacja rezerwacji



Rysunek 11. Diagram czynności - dekompozycja czynności: „Reklamacja rezerwacji”

### Diagramy sekwencji

Poniżej przedstawiono diagram sekwencji do czynności: Uwzględnianie reklamacji.

Diagram ma charakter poglądowy, nazwy klas nie są wiążące, ponieważ niektóre z nich będą generowane automatycznie przez framework, jednakże cały sens i zasada działania zostały zachowane zgodnie ze wzorcem.



Rysunek 12. Diagram sekwencji - Uwzględnianie reklamacji

### Diagramy współpracy

Poniżej przedstawiono diagram współpracy dla czynności: Uwzględnianie reklamacji



Rysunek 13. Diagram współpracy - Uwzględnianie reklamacji

## Projekt interfejsu użytkownika

Poniżej przedstawiono wybrane widoki interfejsu użytkownika systemu z poziomu Klienta, Pracownika Recepcji oraz Managera. Obrazują one najważniejszą funkcję systemu, czyli proces rezerwacji, a także zarządzanie reklamacjami od strony Managera.

### Interfejs Klienta

#### Menu główne oraz Menu użytkownika

Intuicyjne dla użytkownika jest umieszczenie menu w prawym górnym rogu strony. Zdecydowano się podzielić menu na Menu główne – niezależne od zalogowania użytkownika oraz na Menu użytkownika – wyświetlające odnośniki do zarządzania kontem i rezerwacjami.

**Menu główne:**



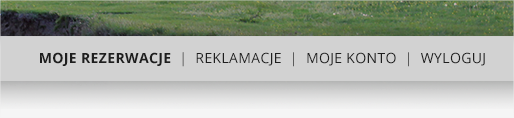
Menu główne zawiera odnośniki do dwóch statycznych stron – strony głównej (O PENSJONACIE) oraz kontaktowej (KONTAKT) oraz do, kluczowej dla systemu, rezerwacji online (1).

**Menu użytkownika przed zalogowaniem:**



Z każdej podstrony aplikacji dostępne jest Menu użytkownika znajdujące się tuż pod Menu głównym. Przed zalogowaniem wyświetlają się odnośniki do rejestracji i logowania.

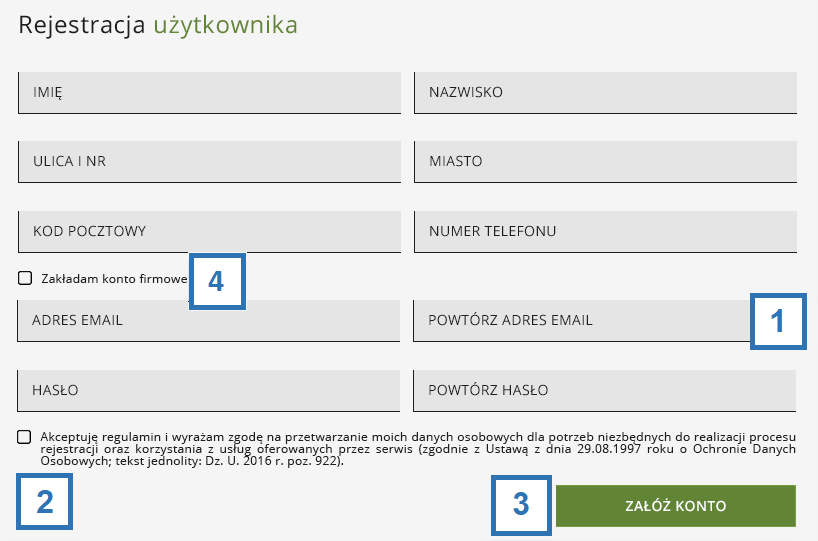
**Menu użytkownika po zalogowaniu:**



Po zalogowaniu Klienta, odnośniki w Menu użytkownika kierują do: rezerwacji Klienta, reklamacji Klienta, edycji konta Klienta (zmiana adresu, hasła) oraz wylogowania z systemu.

#### Formularz rejestracji

Formularz rejestracji pojawia się po kliknięciu przycisku w Menu użytkownika „REJESTRACJA” lub jeśli Klient wejdzie na podstronę systemu, do której dostęp wymaga zalogowania (np. dokonywanie rezerwacji).



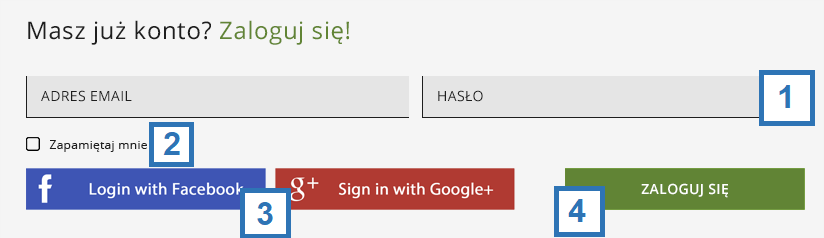
Wyświetla się 10 pól formularza (1) do wpisania danych niezbędnych do rejestracji użytkownika w Systemie. Poniżej znajduje się pole typu checkbox (2) służące do akceptacji regulaminu oraz wyrażenia zgody na przetwarzanie danych osobowych. Formularz potwierdzany jest za pomocą przycisku „ZAŁÓŻ KONTO” (3). W przypadku, gdy Klient chce założyć konto firmowe powinien zaznaczyć pole checkbox „Zakładam konto firmowe” (4). Pokażą się wtedy dodatkowe 2 pola formularza do wprowadzenia nazwy firmy i numeru NIP.

Jeśli użytkownik nie uzupełni wszystkich pól lub podane dane będą błędne, formularz nie wyśle się, a zostaną wyświetlone komunikaty o błędach pod polami formularza, które zostały błędnie wypełnione:



#### Formularz logowania

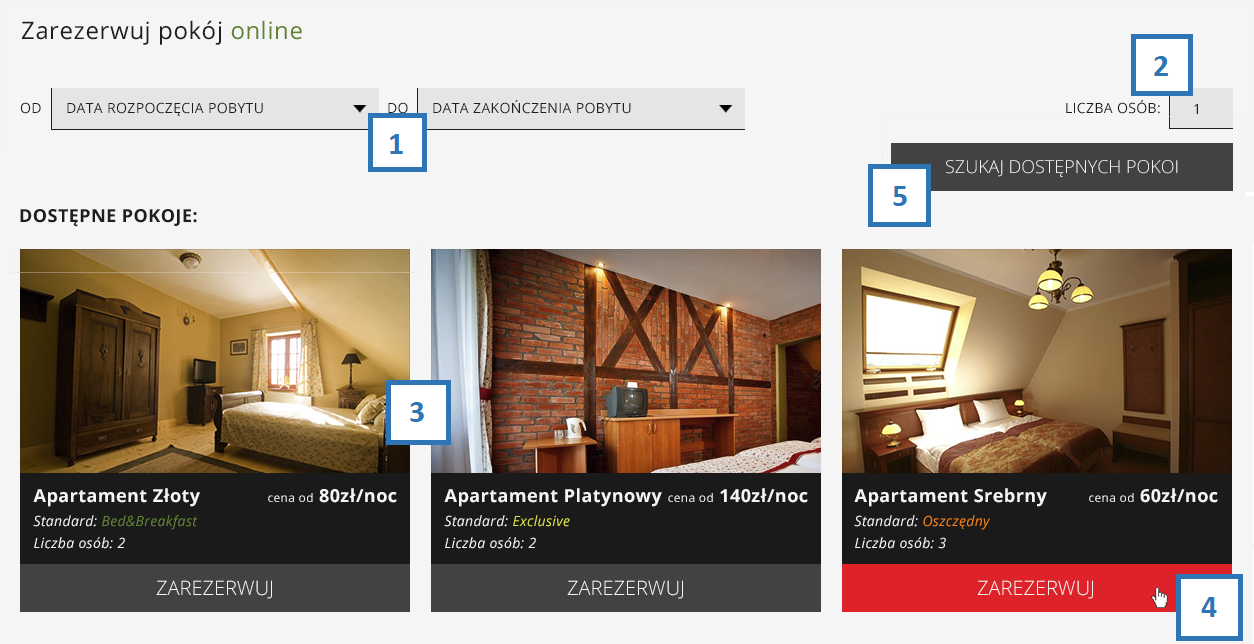
Formularz logowania pojawia się po kliknięciu przycisku w Menu użytkownika „LOGOWANIE” lub jeśli Klient wejdzie na podstronę systemu, do której dostęp wymaga zalogowania (np. dokonywanie rezerwacji).



Wyświetlają się 2 pola formularza (1) do wpisania adresu email oraz hasła użytkownika. Poniżej znajduje się pole typu checkbox (2) służące do aktywowania funkcji „Zapamiętaj mnie”, dzięki której użytkownik po ponownym wejściu na stronę będzie cały czas zalogowany. Istnieje możliwość logowania poprzez konto Facebook i Google (3). Formularz logowania zatwierdzany jest przyciskiem „ZALOGUJ SIĘ” (4).

#### Proces rezerwacji pokoju

Po kliknięciu przycinku Menu głównego „REZERWACJA ONLINE” wyświetlają się dostępne pokoje.

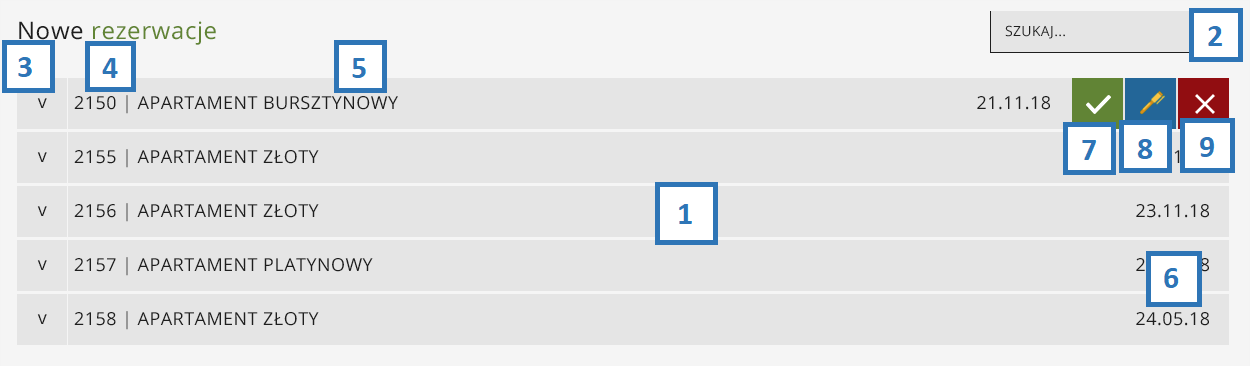


W pierwszym kroku rezerwacji wybierany jest termin pobytu od – do za pomocą pól formularza z datami (1). Następnie należy wpisać żądaną liczbę osób (domyślnie 1) w polu (2) i aby zatwierdzić szukanie pokoju kliknąć przycisk (5).

Wyszukiwanie rozpoczyna się i poniżej (3) prezentowane są pokoje spełniające wymagania co do dat i liczby osób. Aby wybrać pokój do rezerwacji należy kliknąć przycisk „ZAREZERWUJ” (4), który zatwierdzi formularz i przekieruje do formularza rejestracji/logowania (jeśli użytkownik nie jest zalogowany) lub podsumowania rezerwacji.

### Interfejs Pracownika Recepcji

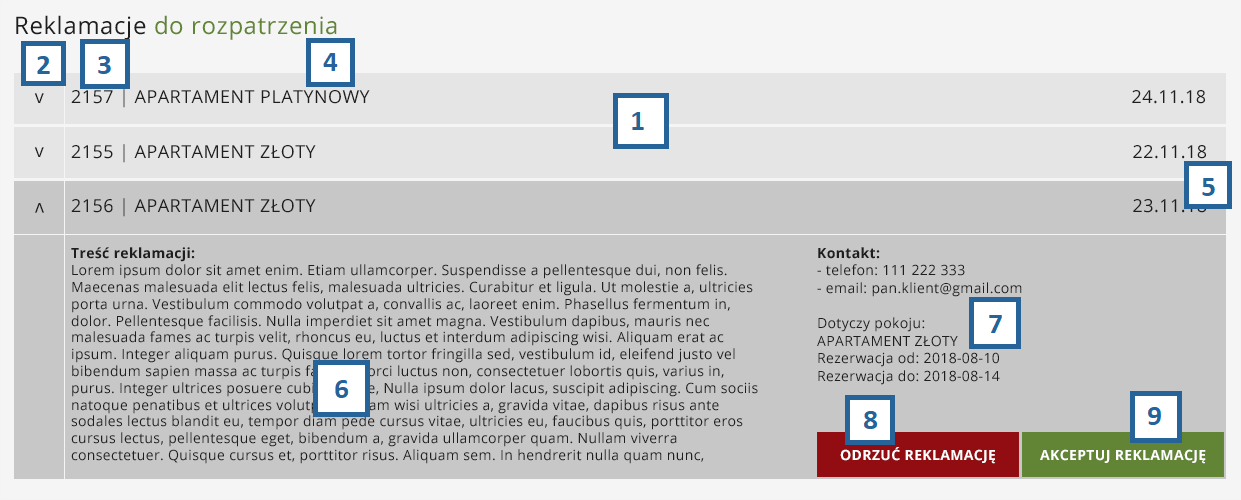
Pracownik Recepcji, po zalogowaniu się na swoje konto oraz kliknięciu w Menu użytkownika „NOWE REZERWACJE”, wyświetla listę rezerwacji (1), które posiadają status „New”.



Pracownik wyszukuje na liście rezerwację np. za pomocą wyszukiwarki (2), której stan należy zmienić na zaakceptowaną (zaliczka została wpłacona). Rezerwacje na liście opisane są następująco: numer rezerwacji (4), nazwa pokoju (5), data (6). Każdą z rezerwacji można rozwinąć i podejrzeć więcej informacji poprzez kliknięcie przycisku (3). Po najechaniu na element list wyświetlają się trzy ikony – akceptacji (7), edycji (8) oraz usunięcia rezerwacji (9).

### Interfejs Managera

Manager, po zalogowaniu się na swoje konto oraz kliknięciu w Menu użytkownika „REKLAMACJE”, wyświetla listę nierozpatrzonych reklamacji (1).



Manager wyszukuje na liście reklamację do rozpatrzenia. Reklamacje na liście opisane są poprzez: numer rezerwacji, której dotyczy reklamacja (3), nazwę pokoju, którego dotyczy reklamacja (4) oraz datę wpłynięcia reklamacji (5). Każdą z reklamacji można rozwinąć i podejrzeć więcej informacji poprzez kliknięcie przycisku (2). Po rozwinięciu, pokazana zostaje treść reklamacji (6) oraz dane kontaktowe Klienta wraz z podstawowymi danymi jego rezerwacji (7). Manager może z tego poziomu odrzucić reklamację (8) lub ją uznać (9).

## Projekt testów kontrolnych

### Testy jednostkowe

Testy jednostkowe maję na celu weryfikacje poprawności działania pojedynczych elementów, to jest jednostek tworzonego oprogramowania. Uzupełniają testy wykonywane manualnie przez użytkowników oraz są bardzo dobrym sposobem na walidację aplikacji mających rozbudowaną logikę biznesową. Główną zaletą tych testów jest automatyzacja procesu testowania aplikacji.

Do przeprowadzenia testów jednostkowych użyjemy programu Microsoft Visual studio 2017 community, dodając nowy projekt testów z frameworkiem.

Każdy test składa się z 3. podstawowych elementów:

* przygotowanie – polega ono na przygotowaniu wszystkich danych wejściowych,
* działanie – polega na testowaniu wybranej funkcji,
* zapewnienie – polega na weryfikacji rezultatu testowanej funkcji.

### Testy bezpieczeństwa

W celu spełnienia wymagań niefunkcjonalnych dotyczących bezpieczeństwa, poniżej przedstawiony zostaje projekt, który ma na celu analizę zabezpieczeń tworzonego systemu.

System wspierający wystawianie/rezerwację apartamentów jest aplikacją web-ową, w której został zaimplementowany mechanizm uwierzytelniania użytkowników w oparciu o ich role. Aby zapewnić prawidłowe funkcjonowanie aplikacji, niezbędna jest poprawna walidacja danych. W związku z tym niżej wymienione obszary podatności na potencjalne błędy logiczne i konfiguracyjne zostaną przetestowane:

* autoryzacja,
* uprawnienia,
* walidacja danych,
* uwierzytelnianie.

Poniżej zostały opracowane scenariusze testów bezpieczeństwa oraz opisy.

1. **Test – Błędne logowanie**

**Scenariusz** – użytkownik podczas logowanie do systemu podał niepoprawne hasło użytkownika.

1. **Test – Walidacja długości hasła**

**Scenariusz** – użytkownik podczas zmiany hasła podał nowe hasło o liczbie znaków równych 2.

1. **Test – Nieautoryzowany dostęp do zasobów**

**Scenariusz** – niezalogowany użytkownik w adres przeglądarki wpisał nazwę kontrolera „ReservationController” w celu uzyskania dostępu do nieautoryzowanej rezerwacji pokoju.

1. **Test – Dostęp do zasobów użytkownika bez uprawnień.**

**Scenariusz** – zalogowany użytkownik niebędący pracownikiem w adres przeglądarki wpisał nazwę kontrolera „Admin” i metodę „Register” w celu uzyskania dostępu do formularza rejestracji nowego użytkownika.

### Testy zgodności

Interfejsem graficznym systemu wspierającego wystawianie/rezerwację apartamentów jest przeglądarka internetowa działająca po stronie klienta. W związku z tym należy poddać weryfikacji, czy aplikacja poprawnie działa na różnych środowiskach.

Zaimplementowany system powinien poprawnie działać co najmniej w czterech najpopularniejszych środowiskach. Warunkiem koniecznym jest działanie na Google Chrome, Mozilla Firefox, Safari oraz Microsoft Edge w najnowszych wersjach.

Do przeprowadzenia testów zgodności wybrano poniższe elementy interfejsu graficznego strony:

* Sprawdzenie pól logowania do systemu
* Sprawdzenie działania i wygląd list rozwijanych.
* Sprawdzenie kontrolki kalendarza.
* Sprawdzenie reakcji przycisków na najechanie myszą
* Sprawdzenie formularza kontaktowego

# Testy systemu

## Testy jednostkowe

Testy jednostkowe maję na celu weryfikacje poprawności działania pojedynczych elementów, to jest jednostek tworzonego oprogramowania.

Główną zaletą tych testów jest automatyzacja procesu testowania aplikacji.

Do przeprowadzenia testów jednostkowych użyliśmy programu Microsoft Visual studio 2017 Community, dodając nowy projekt testów z frameworkiem.

Każdy test składa się z 3. podstawowych elementów:

* przygotowanie – polega ono na przygotowaniu wszystkich danych wejściowych.
* działanie – polega na testowaniu wybranej funkcji.
* zapewnienie – polega na weryfikacji rezultatu testowanej funkcji.

W poniższej tabeli zamieściliśmy opracowane i przeprowadzone testy jednostkowe dla wybranych funkcji systemu. Tabela zawiera nazwę testu oraz jego cel.

|  |  |  |
| --- | --- | --- |
| **Nr.** | **Nazwa testu** | **Cel testu** |
| 1. | CheckRegister | Sprawdzenie, czy metoda Register przy podaniu prawidłowych wartości utworzy rekord w bazie danych. |
| 2. | CheckLogin | Sprawdzenie, czy metoda Login zwróci elementy bazy danych przy podaniu prawidłowych wartości. |
| 3. | CheckAllReclamations | Sprawdzenie, czy metoda AllReclamations zwróci listę o liczbie elementów większych od zera. |
| 4. | CheckUserReclamations | Sprawdzenie, czy metoda UserReclamations zwróci listę o liczbie elementów większych od zera przy podaniu „id” określonego użytkownika. |
| 5. | CheckNewReclamation | Sprawdzenie, czy metoda NewReclamation przy podaniu prawidłowych wartości utworzy rekord w bazie danych. |
| 6. | CheckGetTimeEntry | Sprawdzenie, czy metoda GetTimeEntry dla zdefiniowanego id użytkownika zwróci listę deklaracji czasu pracy. |
| 7. | CheckUserReservations | Sprawdzenie, czy metoda UserReservations zwróci listę o liczbie elementów większych od zera przy podaniu „id” określonego użytkownika. |
| 8. | CheckRolesForUser | Sprawdzenie, czy metoda RolesForUser przy zdefiniowanym id klienta zwróci rolę użytkownika. |
| 9. | CheckUserList | Sprawdzenie, czy metoda UserList zwróci listę użytkowników systemu. |
| 10. | CheckAllBills | Sprawdzenie, czy metoda AllBills dla podanego id zwróci wszystkie rachunki użytkownika. |

## Testy bezpieczeństwa

W celu spełnienia wymagań niefunkcjonalnych dotyczących bezpieczeństwa, poniżej przedstawiony zostaje projekt, który ma na celu analizę zabezpieczeń tworzonego systemu.

System wspierający wystawianie/rezerwację apartamentów jest aplikacją web-ową, w której został zaimplementowany mechanizm uwierzytelniania użytkowników w oparciu o ich role. Aby zapewnić prawidłowe funkcjonowanie aplikacji, niezbędna jest poprawna walidacja danych. W związku z tym niżej wymienione obszary podatności na potencjalne błędy logiczne i konfiguracyjne zostaną przetestowane:

* autoryzacja
* uprawnienia
* walidacja danych
* uwierzytelnianie

Poniżej zostały opracowane scenariusze testów bezpieczeństwa oraz ich wyniki.

1. **Test – Błędne logowanie**

**Scenariusz** – użytkownik podczas logowanie do systemu podał niepoprawne hasło użytkownika.

**Wynik** – walidacja danych przebiegła poprawnie. Otrzymany komunikat to: „Nieudana próba logowania.”

1. **Test – Walidacja długości hasła**

**Scenariusz** – użytkownik podczas zmiany hasła podał nowe hasło o liczbie znaków równych 2.

**Wynik** – walidacja danych przebiegła poprawnie. Użytkownik otrzymał komunikat: „Hasło nie może być krótsze niż 6 znaków”

1. **Test – Nieautoryzowany dostęp do zasobów**

**Scenariusz** – niezalogowany użytkownik w adres przeglądarki wpisał nazwę kontrolera „ReservationController” w celu uzyskania dostępu do nieautoryzowanej rezerwacji pokoju.

**Wynik** – system zgodnie z przewidywaniem przekierował użytkownika na stronę logowania.

1. **Test – Dostęp do zasobów użytkownika bez uprawnień.**

**Scenariusz** – zalogowany użytkownik niebędący pracownikiem w adres przeglądarki wpisał nazwę kontrolera „Admin” i metodę „Register” w celu uzyskania dostępu do formularza rejestracji nowego użytkownika.

**Wynik** – system zgodnie z przewidywaniem, przekierował użytkownika na stronę logowania.

## Testy zgodności

Interfejsem graficznym systemu wspierającego wystawianie/rezerwację apartamentów jest przeglądarka internetowa działająca po stronie klienta. W związku z tym należy poddać weryfikacji, czy aplikacja poprawnie działa na różnych środowiskach.

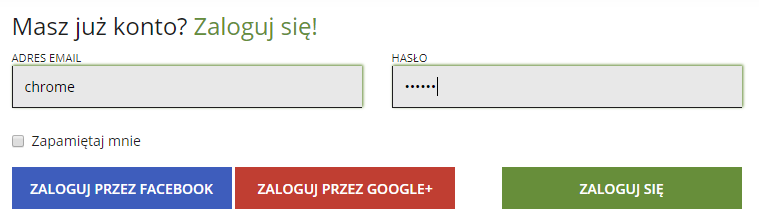
Zaimplementowany system powinien poprawnie działać co najmniej na trzech środowiskach. Warunkiem koniecznym jest działanie na Google Chrome, Mozilla Firefox oraz Microsoft Edge w najnowszych wersjach.

Do przeprowadzenia testów zgodności wybraliśmy poniższe elementy interfejsu graficznego strony:

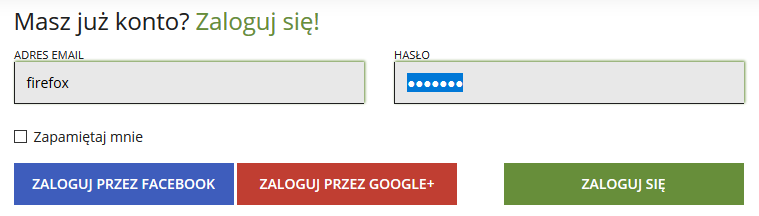
* Sprawdzenie pól logowania do systemu
* Sprawdzenie działania i wygląd list rozwijanych.
* Sprawdzenie pól edycji daty i czasu oraz formaty daty.
* Sprawdzenie reakcji przycisków na najechanie myszą

**Test 1. Sprawdzenie działania i wyglądu pól logowania do systemu**

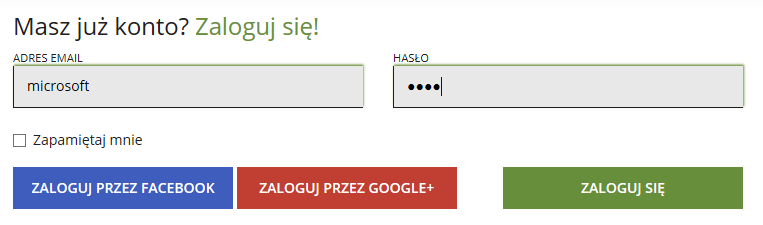
Rezultat dla Google Chrome:



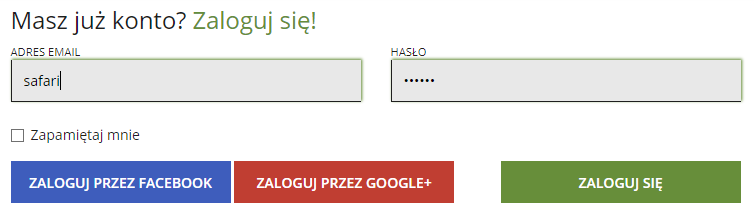
Rezultat dla Mozilla Firefox:



Rezultat dla Microsoft Edge:



Rezultat dla Safari:

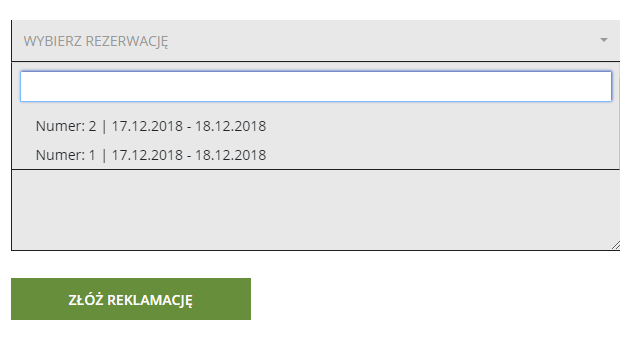


Wynik testu:

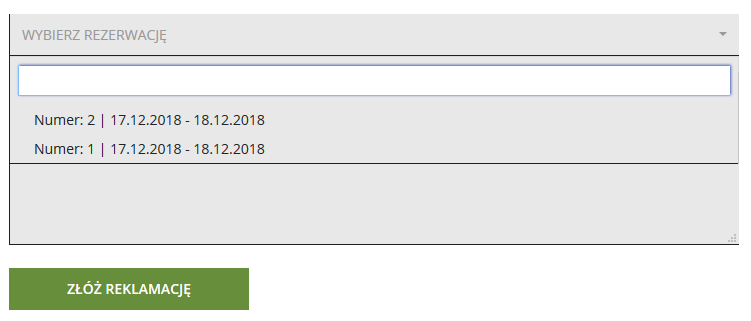
* Pola logowania poprawnie działają na wszystkich testowanych środowiskach.
* Ich wygląd jest niemal identyczny dla testowanych przeglądarek.

**Test 2. Sprawdzenie działania i wygląd list rozwijanych.**

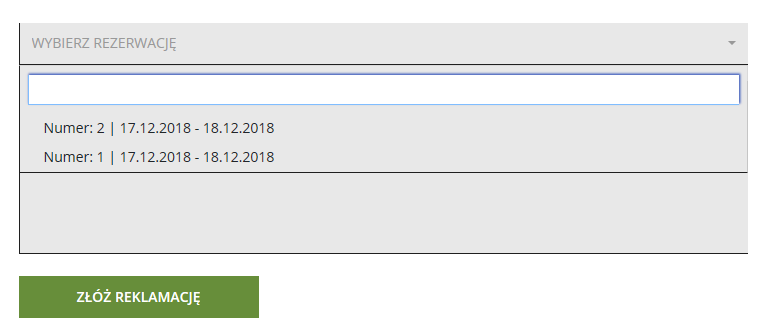
Rezultat dla Google Chrome:



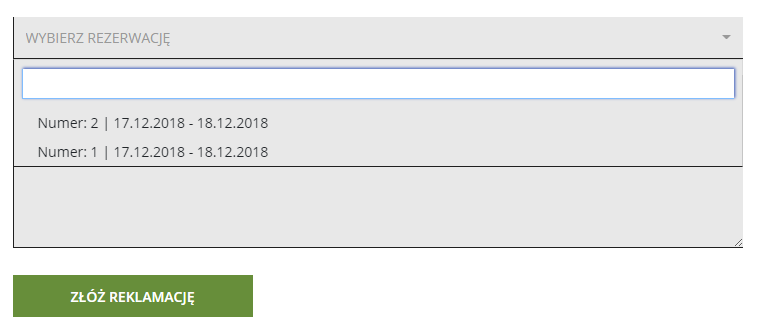
Rezultat dla Mozilla Firefox:



Rezultat dla Microsoft Edge:



Rezultat dla Safari:

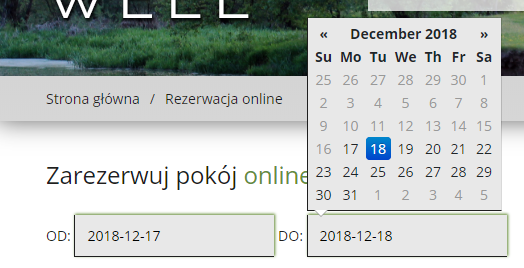


Wynik testu:

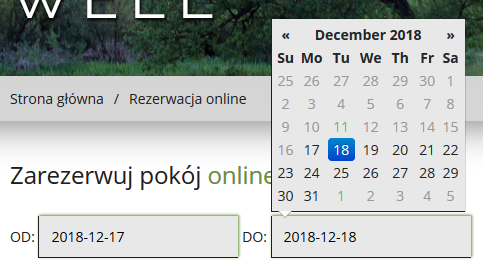
* Listy rozwijane poprawnie działają na wszystkich testowanych środowiskach.
* Ich wygląd jest identyczny dla wszystkich testowanych przeglądarek.

**Test 3. Sprawdzenie pól edycji daty i czasu oraz formaty daty.**

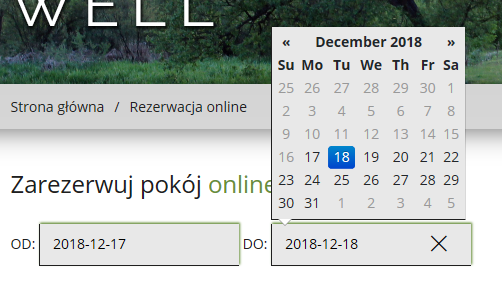
Rezultat dla Google Chrome:



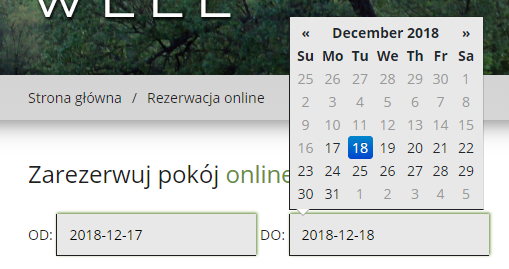
Rezultat dla Mozilla Firefox:



Rezultat dla Microsoft Edge:



Rezultat dla Safari:



Wynik testu:

* kontrolki daty we wszystkich testowanych przeglądarkach mają taki sam wygląd
* Format daty i czasu jest zgodny - „YYYY/MM/DD”.

**Test 4. Sprawdzenie reakcji przycisków na najechanie myszą**

Rezultat dla Google Chrome:



Rezultat dla Mozilla Firefox:



Rezultat dla Microsoft Edge:



Rezultat dla Safari:



Wynik testu:

* Reakcja przycisków działa na wszystkich testowanych środowiskach.
* Ich wygląd zmienia się w ten sam sposób, po najechaniu myszką w testowanych przeglądarkach.