**ETAP 2**

SYSTEM USŁUG SPRZĄTAJĄCYCH

|  |  |
| --- | --- |
| Imię i nazwisko studenta: | **Jakub Owczarzak, Andrzej Chmielewski**  **Rafał Spiżewski, Oliwia Andres, Michał Mackiewicz** |
| Grupa: | **Z710** |
| Specjalność: | **Inżynieria Internetu** |

Spis treści

[1. System i jego otoczenie 3](#_Toc87703788)

[**2.** **Specyfikacja** 4](#_Toc87703789)

[2.1. Niefunkcjonalna 4](#_Toc87703790)

[2.2 . Funkcjonalna 5](#_Toc87703791)

[2.3 Funkcje systemu 6](#_Toc87703792)

[**3** **Architektura systemu** 10](#_Toc87703793)

[3.1 Opis architektury systemu 10](#_Toc87703794)

[3.2 Diagram przypadków użycia 11](#_Toc87703795)

[**4** **Model danych** 13](#_Toc87703796)

[3.2 Diagram ERD notacji Chen 13](#_Toc87703797)

## System i jego otoczenie

Projektowana aplikacja to system zajmujący się obsługą firmy świadczącej usługi z zakresu sprzątania mieszkań i biur. W związku z tym, powinna w pełni wspierać proces obsługi i rejestracji zleceń oraz zarządzanie klientami.

Tego typu usługi, możliwe do zamówienia nie poprzez telefon czy wizytę, ale za pośrednictwem sieci, są w Polsce dalej niszą. Jest to doskonały moment, żeby wprowadzić na rynek nowy produkt, który jakością będzie zauważalnie przewyższał konkurencję, a co za tym idzie – przyciągał klientów, korzystających dotychczas z usług konkurencji.

Aplikacja będzie stroną internetową, a więc uruchamiana będzie w dowolnej przeglądarce, a także na dowolnym urządzeniu. Użytkownik będzie mógł z niej skorzystać z dowolnego miejsca, pod warunkiem dostępu do Internetu. Serwis będzie posiadał przejrzysty interfejs graficzny, który dodatkowo dostosowany do różnych wielkości ekranów, zapewniając użytkownikowi wygodę korzystania również na urządzeniach mobilnych – smartfonach i tabletach.

Zgodnie z panującymi obecnie trendami, serwis zostanie wykonany głównie w języku JavaScript, przy użyciu biblioteki React.js. Z jego pomocą zbudowany interfejs będzie szybki i dynamiczny, a jego funkcje będą niezwykle przystępne i przyjemne w odbiorze dla klientów. Część kliencka serwisu będzie łączyć się z warstwą serwerową, również napisaną w JavaScripcie, przy użyciu Node.js. Dane będą magazynowane za pomocą MongoDB.

Przystępując do prac projektowych, duży nacisk położona na wygodę i funkcjonalność. Głównym założeniem jest, żeby serwis nie odstraszał klientów, a wręcz przeciwnie – swoim wyglądem oraz zachowaniem przyciągał nowych. Ważne jest, aby serwis działał poprawnie na różnych przeglądarkach, żeby nie narzucać użytkownikom tylko jednego typu.

## **Specyfikacja**

### Niefunkcjonalna

Do najważniejszych wymagań niefunkcjonalnych należy zaliczyć:

* Aplikacja ma być dostępna przez cały czas, 24 godziny na dobę, 7 dni w tygodniu
* Aplikacja będzie dostępna dla wielu użytkowników jednocześnie
* Aplikacja będzie dostępna dla użytkowników bez względu na ich lokalizację
* Serwis będzie dostępny za pośrednictwem sieci Internet
* Strona będzie działała poprawnie na wszystkich popularnych przeglądarkach internetowych:
  + Chrome od wersji 87.0.
  + Chrome Mobile od wersji 87.0.
  + Firefox od wersji 84.0.
  + Microsoft Edge od wersji 87.0.
  + Safari od wersji 14.1.
* Serwis będzie działał poprawnie na wszystkich typach urządzeń, używających systemów operacyjnych pozwalających na uruchomienie przeglądarki w wersjach wymienionych powyżej, w tym na urządzeniach desktopowych oraz mobilnych – telefonach i tabletach
* Interfejs będzie automatycznie dostosowywał się do typu oraz rozmiaru urządzenia, na którym serwis jest wyświetlany
* System będzie zarządzał danymi osobistymi w sposób zgodny z obecnie panującym prawem
* System będzie stworzony w sposób, który umożliwi proste rozszerzanie go o nowe funkcje
* Serwis zapewni bezpieczne mechanizmy logowania – autoryzacji oraz dostępu do zasobów

### . Funkcjonalna

W systemie będą istniały trzy typy użytkowników zalogowanych: klient, administrator oraz pracownik. Każda z grup będzie posiadała inne uprawnienia, a system będzie oferować różne funkcjonalności. Wymagania funkcjonalne z podziałem na użytkowników:

Użytkownik niezalogowany:

* Logowanie
* Rejestracja
* Tworzenie nowego zamówienia z możliwością, jednak bez konieczności rejestracji na końcu

Konto klienta:

* Tworzenie nowego zamówienia za pośrednictwem formularza
* Przeglądanie informacji o obecnych oraz starych, zrealizowanych zamówieniach
* Zapisywanie i edycja miejsc, w celu uniknięcia potrzeby konfiguracji miejsca oddzielnie dla każdego zamówienia
* Wyświetlanie i edycja profilu

Konto pracownika:

* Wyświetlanie listy nowych zamówień
* Przyjmowanie zamówień
* Zmiana statusu zamówienia

Konto administratora:

* Zarządzanie użytkownikami systemu
* Pełna kontrola nad zamówieniami
* Zarządzanie usługami

### Funkcje systemu























## **Architektura systemu**

### 3.1 Opis architektury systemu



Aplikacja kliencka zostanie napisana przy pomocy popularnej biblioteki języka JavaScript – React.js. Zapewni to dobrą wydajność, a także możliwość stworzenia bardzo dynamicznego i wygodnego w użytkowaniu interfejsu.

Warstwa backendowa wykonana zostanie za pomocą Node.js oraz frameworka Express.js. Również jest to jedno z bardziej popularnych obecnie rozwiązań, zapewniające stabilność i wydajność. Backend będzie łączył się z bazą MongoDB oraz udostępniał REST API pozwalające na kontaktowanie się z nim od strony frontendu.

Na wybór technologii duży wpływ miały umiejętności całego zespołu. Okazało się bowiem, że każdy z członków najwięcej doświadczenia ma właśnie z JavaScriptem. Niewątpliwie duże znaczenie miała również popularność wybranych rozwiązań. Zapewnia to mnogość gotowych paczek tworzonych przez społeczność, a także bardzo szeroki dostęp do materiałów dydaktycznych oraz wszelkiej pomocy, która może okazać się pomocna podczas procesu tworzenia systemu.

### 3.2 Diagram przypadków użycia





## **Model danych**

### 3.2 Diagram ERD notacji Chen

