



AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

KATEDRA INFORMATYKI STOSOWANEJ

Inżynieria oprogramowania

System obsługi restauracji

Autor:

Aleksandra Kasznia, Paweł Gałka, Marcin Grzyb

Kierunek studiów:

Informatyka

Opiekun pracy:

dr inż. Marek Zachara

Kraków, 2019

Spis treści

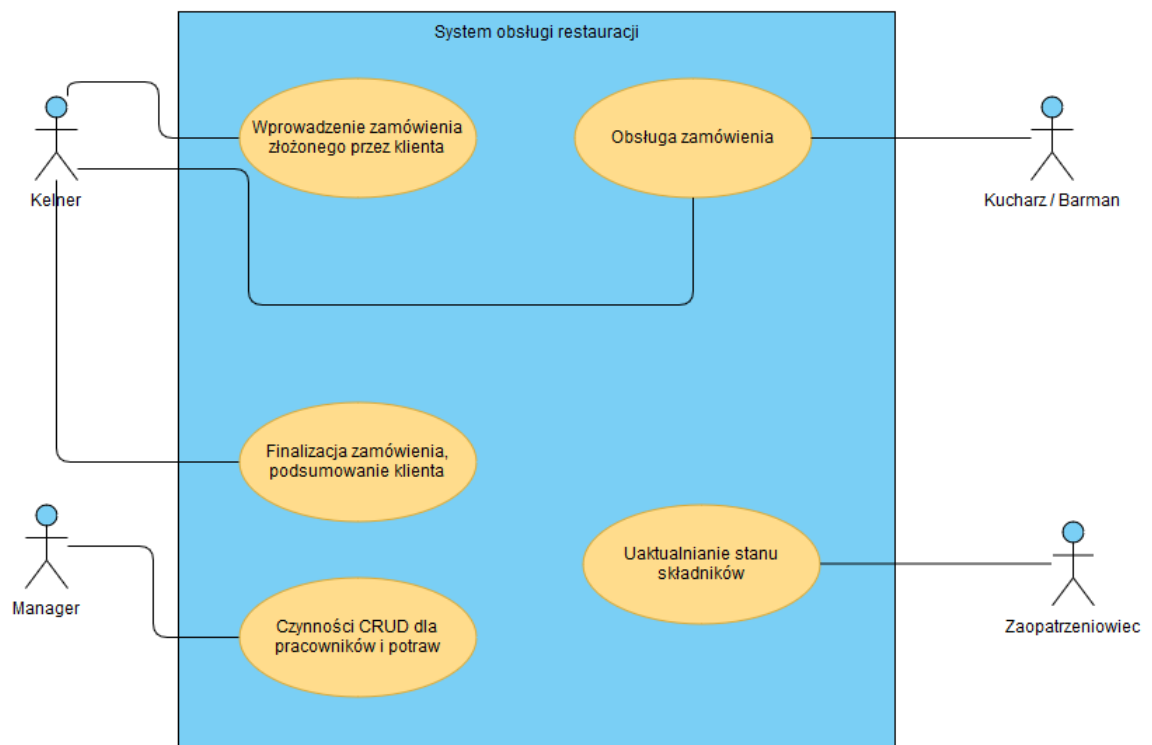
1. Wymagania	4
2. Opis poszczególnych wariantów działania systemu ze względu na rolę użytkownika.	6
2.1. Kelner	6
2.2. Kucharz / Barman	6
2.3. Manager	6
2.4. Zaopatrzeniowiec	6
2.5. Klient	7
3. Diagram kontekstowy ogólnych procesów systemu	8
4. Use Cases	9
4.1. Obsługa zamówienia przez barmana	9
4.2. Obsługa zamówienia przez kelnera	10
4.3. Obsługa braku produktów przez zaopatrzeniowca	11
4.4. Obsługa zamówienia przez kucharza	12
5. Architektura Systemu	13
5.1. Moduły Systemu	13
5.2. Struktura bazy danych	14
6. Opis Interfejsów	15
7. Stos Technologiczny	16
7.1. Technologie do serwera wystawiającego API	16
7.2. Technologie do aplikacji mobilnej	17
7.3. Technologie do platformy webowej	17
7.4. Współdzielone technologie	17
8. Projekt Testów	18
8.1. Testy jednostkowe	18
8.2. Testy integracyjne	18
8.3. Testy wydajnościowe	18
8.4. Testy funkcjonalne	18

8.5. Testy frontendowe	19
9. Analiza ryzyka.....	20
9.1. Lista ryzyk i odpowiednie akcje.....	20
9.2. Macierz ryzyka	22
10. Narzędzia używane w projekcie.....	23

1. Wymagania

Funkcjonalności:

- Pracownik loguje się do aplikacji, w zależności od funkcji, którą pełni dostaje różne możliwe czynności do wykonania.
- Kelner po wprowadzeniu id stolika przypisuje go do siebie, wprowadza zamówienia do aplikacji z określonej bazy, które zostają przekazane do kucharza. Kelner otrzymuje powiadomienie o gotowości zamówień. Po zakończonej obsłudze klient ocenia obsługę w aplikacji kelnera.
- Kucharz otrzymuje w aplikacji zamówienia do przygotowania i zaznacza ich gotowość, po wykonaniu dania w bazie danych aktualizuje się stan dostępnych składników.
- Barman otrzymuje w aplikacji zamówienia i zaznacza ich gotowość, po wykonaniu napoju w bazie danych aktualizuje się stan dostępnych składników.
- Magazynier wprowadza dostawy do bazy danych i dostaje ostrzeżenia o kończących się produktach.
- Manager otrzymuje raporty na temat jakości obsługi i popularności dań
- Klient rezerwuje wizytę poprzez internetową platformę
- Klient po zakończeniu wizyty wystawia opinie na temat jakości obsługi



Rys. 1.1. Poglądowy diagram zależności w systemie

2. Opis poszczególnych wariantów działania systemu ze względu na rolę użytkownika.

2.1. Kelner

Aplikacja dostępna dla kelnera ma na celu usprawnienie obsługi klientów restauracji. Kelner przyjmując zamówienie wybiera pozycje dostępne z menu, jednocześnie w ten sposób jest możliwie natychmiast informowany o awaryjnej niedostępności którejś z pozycji. Zamówienie jest przypisane do stolika który aktualnie obsługuje, w systemie również zostaje zaznaczone, które stoliki obsługuje dana osoba. Po wprowadzeniu zamówienia, kelner zostanie poinformowany kiedy ma je odebrać z kuchni lub baru. W momencie finalizacji, system umożliwia szybkie podsumowanie klienta. Kelner może pokazać klientowi podsumowanie, na tym samym ekranie jest dostępna również krótka ankieta dla klienta.

2.2. Kucharz / Barman

Dla kucharza jak i barmana dostępne są analogiczne funkcjonalności. Zostają oni poinformowani o nowym zamówieniu, które następnie muszą zaakceptować. Po wykonaniu go zaznaczają w aplikacji iż zamówienie jest gotowe.

2.3. Manager

Jest to właściwie interfejs stworzony do obsługi bazy danych. Manager ma możliwość wykonywania czynności CRUD dla wszystkich tabel w bazie. Dodatkowo ma możliwość wyświetlenia raportu z poprzedniego miesiąca podsumowującego pracę restauracji.

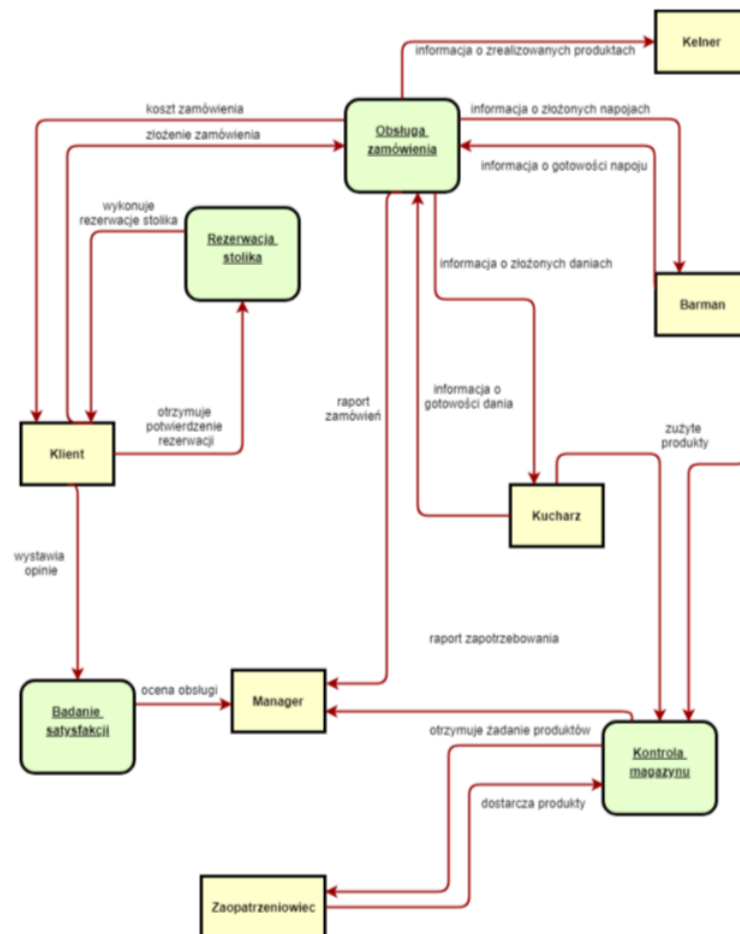
2.4. Zaopatrzeniowiec

Zaopatrzeniowiec Zostaje poinformowany o konieczności uzupełnienia zaopatrzenia gdy poziom potrzebnego składnika spada poniżej poziomu ostrzegawczego. Ma możliwość uaktualniania stanu produktów w bazie jak i wglądu w niego.

2.5. Klient

Dla klienta przez stronę internetową będzie udostępniony system rezerwacyjny, umożliwiający mu rezerwację stolika na konkretną godzinę i dzień.

3. Diagram kontekstowy ogólnych procesów systemu



Rys. 3.1. Diagram kontekstowy

4. Use Cases

4.1. Obsługa zamówienia przez barmana

Nazwa	Obsługa zamówienia przez barmana
Główny aktor	Barman
Aktorzy drugoplanowi	Kelner
Wyzwalacz	Wprowadzenie przez kelnera zamówienia na napoje do systemu
Warunki początkowe	Barman jest zalogowany do systemu i rozpoznany jako użytkownik o funkcji barmana.
Warunki końcowe	Zamówienie jest zrealizowane i usunięte z ekranu barmana.
Przepływ normalny	<ol style="list-style-type: none">1. Barman zatwierdza otrzymane zamówienie.2. System odejmuje z bazy dostępnych produktów te, które są potrzebne do wykonania zamówienia.3. Po przygotowaniu barman zaznacza w systemie, iż zamówienie jest gotowe do odbioru.4. System przesyła informację o zakończeniu zamówienia do aplikacji kelnera.5. System usuwa zamówienie z ekranu barmana.
Przepływ alternatywny	<ol style="list-style-type: none">1a. Barman anuluje zamówienie.1b. System informuje kelnera o akcji barmana.1c. Powrót do kroku 5.
Wyjątki	<ol style="list-style-type: none">3a. Barman traci możliwość wykonania zamówienia.3b. Barman anuluje zamówienie.3c. System pyta barmana których napojów nie jest w stanie wykonać.3d. Barman zaznacza odpowiednie pozycje.3e. System wybrane przez barmana napoje wyklucza tymczasowo z menu.

4.2. Obsługa zamówienia przez kelnera

Nazwa	Obsługa zamówienia przez kelnera
Główny aktor	Kelner
Aktorzy drugoplanowi	Kucharz, Barman
Wyzwalacz	Klient restauracji chce złożyć zamówienie u kelnera.
Warunki początkowe	Kelner, kucharz i barman są zalogowani do systemu i rozpoznani.
Warunki końcowe	Zamówienie jest zrealizowane i usunięte z systemu.
Przepływ normalny	<ol style="list-style-type: none"> 1. Kelner wybiera id stolika który obsługuje. 2. System pyta kelnera co chce wykonać. 3. Kelnera wybiera z menu opcję wprowadzenia zamówienia. 4. System wyświetla dostępne napoje i potrawy. 5. Kelner wybiera napoje i potrawy zamawiane przez klienta. 6. Kelner zatwierdza zamówienie. 7. System przesyła zamówienie do aplikacji kucharza i barmana 8. Kelner jest informowany przez system gdy zamówienia przygotowane przez kucharza oraz przez barmana są gotowe do odbioru. 9. Kelner zaznacza iż zamówienie zostało dostarczone do stolika. 10. Kelner wybiera opcję podsumowania danego stolika. 11. System wyświetla kelnerowi kwotę do zapłaty przez klienta. 12. Kelnera potwierdza finalizację płatności. 13. System wyświetla zapytanie o stopień zadowolenia klienta. 14. Klient wybiera odpowiedni obrazek. 15. System zapisuje wyniki ankiety w bazie danych.
Przepływ alternatywny	<ol style="list-style-type: none"> 10a. Kelner potrzebuje wprowadzić dalsze zamówienie 10b. Powrót do punktu 3.
Wyjątki	<ol style="list-style-type: none"> 8a. Zamówienie zostaje anulowane przez barmana lub kucharza. 8b. Powrót do punktu 10.

4.3. Obsługa braku produktów przez zaopatrzeniowca

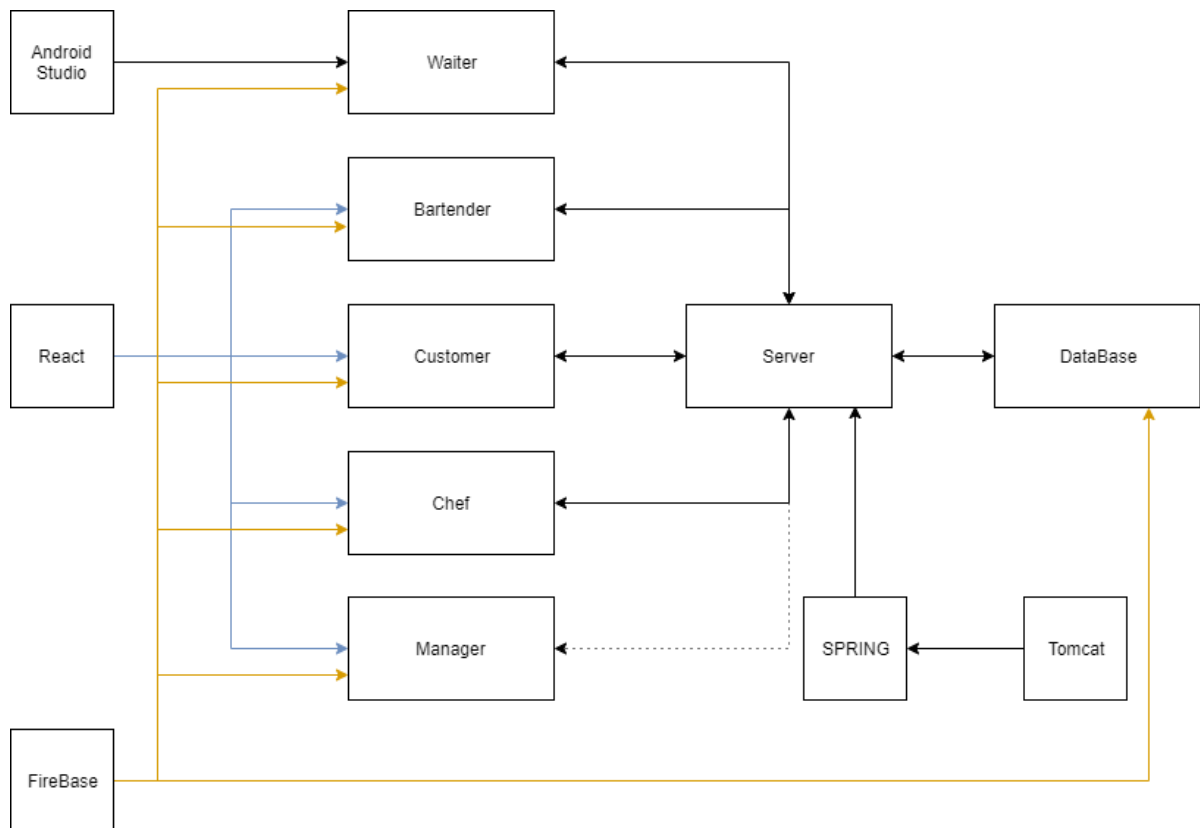
Nazwa	Obsługa braku produktów przez zaopatrzeniowca
Główny aktor	Zaopatrzeniowiec
Aktorzy drugoplanowi	
Wyzwalacz	Ilość sztuk produktu spada poniżej ustalonego poziomu granicznego.
Warunki początkowe	Zaopatrzeniowiec jest zalogowany do systemu i rozpoznany jako użytkownik o funkcji zaopatrzeniowca.
Warunki końcowe	Ilość sztuk produktu jest powyżej ustalonego poziomu granicznego.
Przepływ normalny	<ol style="list-style-type: none"> 1. Zaopatrzeniowiec zostaje poinformowany przez system o niskim poziomie sztuk danych produktów. 2. Po uzupełnieniu braków zaplecza zaopatrzeniowiec wybiera produkt z listy i wprowadza ile sztuk dokupił. 3. System aktualizuje bazę danych. 4. System usuwa informację o niskim poziomie produktów.
Przepływ alternatywny	<ol style="list-style-type: none"> 4a. Po aktualizacji ilość sztuk produktu wciąż jest poniżej ustalonego poziomu granicznego. 4b. Powrót do punktu 1.
Wyjątki	Brak

4.4. Obsługa zamówienia przez kucharza

Nazwa	Obsługa zamówienia przez kucharza
Główny aktor	Kucharz
Aktorzy drugoplanowi	Kelner
Wyzwalacz	Wprowadzenie przez kelnera zamówienia na jedzenie do systemu
Warunki początkowe	Kucharz jest zalogowany do systemu i rozpoznany jako użytkownik o funkcji kucharza
Warunki końcowe	Zamówienie jest zrealizowane i usunięte z ekranu kucharza
Przepływ normalny	<ol style="list-style-type: none"> 1. Kucharz zatwierdza otrzymane zamówienie. 2. System odejmuje z bazy dostępnych produktów te, które są potrzebne do wykonania zamówienia. 3. Po przygotowaniu kucharz zaznacza w systemie iż zamówienie jest gotowe do odbioru. 4. System przesyła informację o zakończeniu zamówienia do aplikacji kelnera. 5. System usuwa zamówienie z ekranu kucharza.
Przepływ alternatywny	<ol style="list-style-type: none"> 1a. Kucharz anuluje zamówienie. 1b. System informuje kelnera o akcji kucharza. 1c. Powrót do kroku 5.
Wyjątki	<ol style="list-style-type: none"> 3a. Kucharz traci możliwość wykonania zamówienia. 3b. Kucharz anuluje 3c. System pyta kucharza których potraw nie jest w stanie 3d. Kucharz zaznacza odpowiednie 3e. System wybrane przez kucharza dania wyklucza tymczasowo z menu.

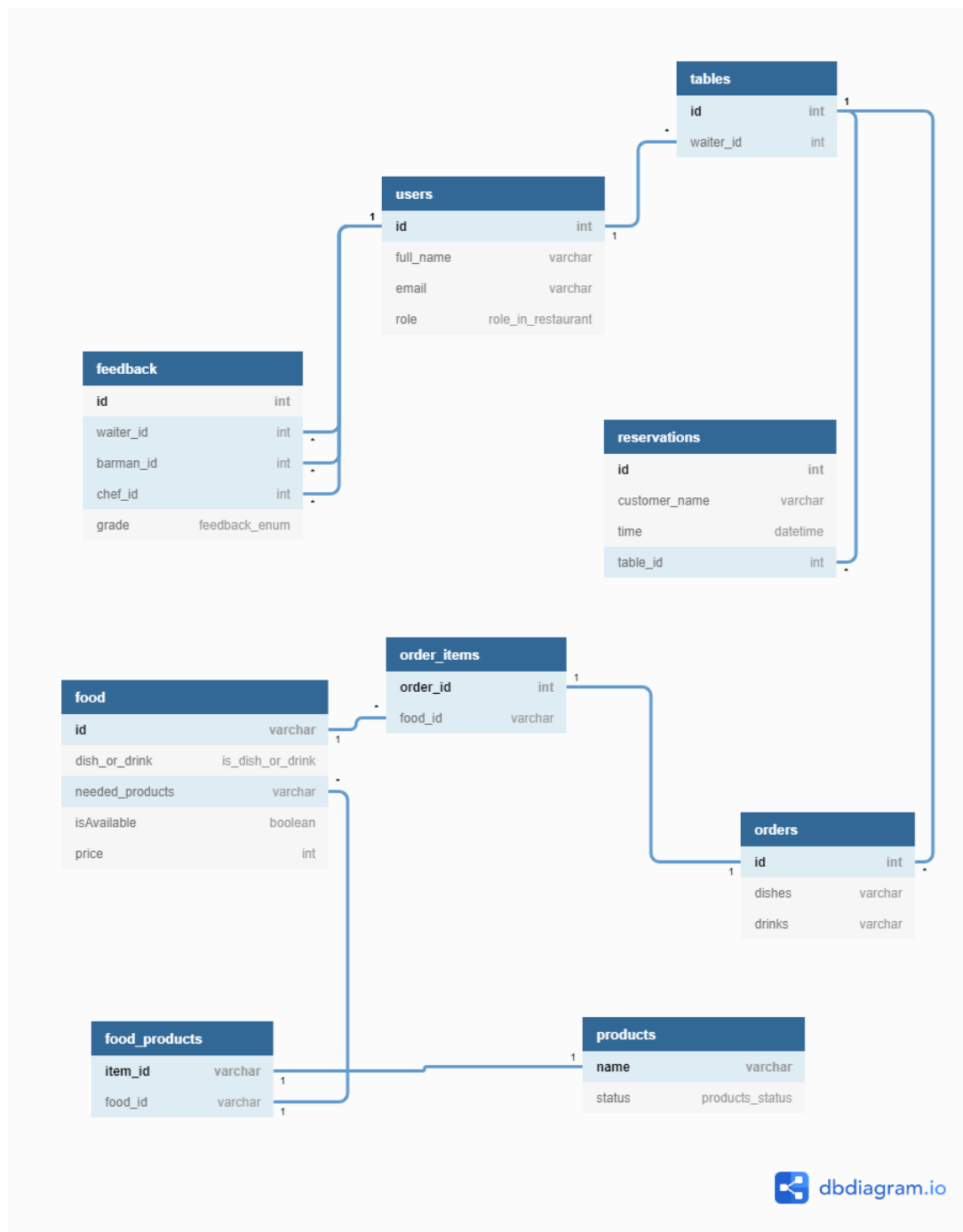
5. Architektura Systemu

5.1. Moduły Systemu



Rys. 5.1. Moduły systemu

5.2. Struktura bazy danych



Rys. 5.2. Baza danych

6. Opis Interfejsów

Opis interfejsów:

- REST API – API wystawiane przez serwer dla aplikacji kelnera, stron managera, barmana kuchara, podzielone według następujących „podinterfejsów”
 - / - dostęp dla wszystkich
 - /waiter– kelner, manager
 - /bartender – barman, manager
 - /management – manager
 - /supplier – zaopatrzeniowiec, manager
- Połączenie z baza danych – FireBase/SQL poprzez JDBC

7. Stos Technologiczny

7.1. Technologie do serwera wystawiającego API

:

- Java 11 – użycie obiektowego programowania ułatwi zarządzanie systemem
- SpringBoot 2.11 – technologia pozwala na zbudowanie aplikacji webowej udostępniającej interfejs API oraz połączenie z bazą danych
- Spring JDBC – interfejs połączenia bazy danych z aplikacją serwera
- MongoDB/ PostgreSQL - użycie bazy danych ułatwi przechowywanie danych i ewentualny restore ich w razie nieoczekiwanych awarii systemu
- Hibernate – obsługa operacji CRUD w systemie do bazy danych
- Junit5 – wykorzystany w celu przeprowadzenia testów aplikacji i logicznego podziału scenariuszy testowych na grupy itp.
- AssertJ – biblioteka ułatwi wykonanie odpowiednich asercji na scenariusze w celu zwiększenia czytelności i maintainencu kodu źródłowego
- Tomcat – umożliwi uruchomienie aplikacji na zdalnym serwerze (wraz z SpringBoot)
- JavaMailSender (automatyzacja wysyłania zapotrzebowania do zaopatrzeniowca) – użyty w celu automatyzacji wysyłania wiadomości do odpowiednich osób w restauracji
- Jmeter – wykorzystany do przeprowadzenia symulacji dużego obciążenia serwera
- Maven – wykorzystany w celu importu bibliotek i zależności potrzebnych do implementacji funkcjonalności

7.2. Technologie do aplikacji mobilnej

- Android SDK - SDK do napisania aplikacji na platformę Android
- Material Design - biblioteka użyta do wyświetlania grafik związanych z gastronomią
- Gradle - do pobierania bibliotek i zależności używanych w projekcie

7.3. Technologie do platformy webowej

- JavaScript - służący do napisania kodu strony webowej
- React (oraz dostępne do niego biblioteki m.in. redux, react test library i inne rozszerzenie dostępne przez komendy npm) - do stworzenia funkcjonalnego frontendu aplikacji webowej
- HTML - do stworzenia szkieletu strony w której osadzony będzie kod witryny
- NodePackageModules - do pobierania bibliotek i zależności używanych w projekcie
- Enzyme - biblioteka służąca do przetestowania działania witryny webowej

7.4. Współdzielone technologie

- FireBase - służący do zarządzania użytkownikami i do uwierzytelniania
- SonarQube - do przeprowadzenia statycznej analizy kodu

8. Projekt Testów

8.1. Testy jednostkowe

Testy jednostkowe będą służyły walidacji działania niskopoziomowych metod operujących na obiektach. Przykładowo czy zamówienie dania powoduje usunięcie odpowiedniej ilości produktów z listy w danym zakresie. W szczególności będą używane przy tworzeniu aplikacji webowej, gdzie będziemy testować działanie komponentów, z których będzie zbudowana aplikacja.

8.2. Testy integracyjne

Testy integracyjne będą sprawdzały integracje aplikacji z baza danych, czy nawiązana jest sesja, czy operacje CRUD są poprawnie wykonywane

8.3. Testy wydajnościowe

Z użyciem Jmeter będziemy testować zachowanie systemu pod dużym ale znormalizowanym obciążeniem sprawdzając czasy odpowiedzi serwera na zapytania http

8.4. Testy funkcjonalne

Testy te będą wykonywane w operacjach które dotyczą biznesowych aspektów funkcjonowania systemu:

- Pomyślna rezerwacja stolika itp.
- Generowanie requestu zapotrzebowania produktu
- Pomyślne przeprosowanie oceny obsługi klienta
- Generowanie raportu miesięcznego na temat obsługi do managera
- Pomyślny przebieg zamówienia
- Generowanie statystyk dań do managera

8.5. Testy frontendowe

Testy frontendu będą wykonane z użyciem biblioteki Enzyme. Testowane będzie:

- renderowanie się komponentu w trybie shallow (bez sprawdzania interakcji z innymi komponentami DOM)
- testy przekazywanych propsów - czy w zależności od podanych parametrów komponent przyjmuje odpowiednie wartości
- interakcje komponentu w z innymi (np. czy pole tekstowe odpowiednio filtruje liste

9. Analiza ryzyka

9.1. Lista ryzyk i odpowiednie akcje

Ryzyko	Prawdopod.	Wpływ	Działanie
System nie odpowiada	15%	Critical	Zgłoszenie problemu procesowane przez odpowiednie osoby
Brak połączenia internetowego do aplikacji rezerwacji	5%	High	Umieszczona zostaje informacja o tymczasowym zawieszeniu działania rezerwacji, uruchomienie rezerwacji telefonicznej
Zmiana scope'u projektu	20%	High	Przeprowadzane jest replanowanie projektu tak aby nie zaburzyć jego funkcjonowania
Brak odpowiedzi od zaopatrzeniowca	25%	Medium	Manager kontaktuje się z zaopatrzeniowcem w celu wyjaśnienia problemu
Problemy z uwierzytelnieniem użytkowników w aplikacji	25%	Critical	Sprawdzenie poprawności danych w module autentykacyjnym, issue-track trace'u procesu autentykacji, restore użytkownika
Problemy z utrzymaniem terminu projektu	60%	Minor	Reorganizacja teamu, zmiana scope'u pracy osób w celu wyrównania zaległości
Problemy z implementacją funkcjonalności z przyczyn technicznych	45%	High	Wydzielenie osobnego modułu z innym stosem technologicznym umożliwiającym implementację

Ryzyko	Prawdopod.	Wpływ	Działanie
Jeden z modułów aplikacji przestaje pracować zgodnie z wymaganiami	35%	Critical	Moduł jest czasowo wygaszany do momentu naprawy, informowanie managera, analogowa obsługa zadań modułu
System nie odpowiada	15%	Critical	Zgłoszenie problemu procesowane przez odpowiednie osoby
Problemy z połączeniem do bazy danych	45%	High	Wykonywana jest kopia zapasowa bazy danych, odłączany jest moduł bazodanowy w celu przeprowadzenia inwestygacji
System napotyka krytyczny błąd w środowisku produkcyjnym niezaleziony w środowisku dewowym	20%	Critical	Aplikacja jest rebootowana, restorowana jest baza danych, zbierane sa logi aplikacji, moduł serwera jest odłączany od aplikacji klienckich, podłączany jest zespół dewowy
Klient nie akceptuje gotowej funkcjonalności	80%	Minor	Forma funkcjonalności jest ponownie konsultowana z klientem, reimplementacja
Nowa funkcjonalność zmienia architekturę systemu	65%	High	Zespół architektów systemu replanuje architekturę, identyfikacja nowych modułów/interfejsów
Kluczowy członek zespołu odchodzi z projektu	45%	High	Przygotowywana jest dokumentacja „as-is” i „to-be”, job shadowing w celu wdrożenia osoby do nowych zadań
Zmniejszenie budżetu projektu	30%	Critical	Przeprowadzane są analizy kosztów projektu, prezentowane są statystyki dla rady projektu

9.2. Macierz ryzyka

Elementy macierzy:

- **Acceptable** – mały impakt na projekt
- **Tolerable** – wpływ na projekt bez krytycznych konsekwencji
- **Undesirable** – poważny wpływ na projekt, wymaga rozwiązania
- **Intolerable** – krytyczny wpływ na projekt, wymaga rozwiązania w pierwszej kolejności

		Wpływ			
Prawdopodob.		Mały	Średni	Duży	Krytyczny
	Małe 0% -25%				
	Możliwe 26% - 50%				
	Prawdopodobne 51% - 75%				
	Bardzo prawdopodobne 76% - 100%				

10. Narzędzia używane w projekcie

- GitHub - przechowywanie repozytorium i kodu źródłowego
- Slack - komunikator nakierowany na tworzenie oprogramowania, do synchronizacji pracy w te-
amie
- ClickUp - narzędzie do issue-trackingu
- Scrum - do planowania zadań na najbliższy okres
- Kanban Tool - do monitorowania postępu pracy
- IntelliJ IDEA - do tworzenia kodu serwera
- WebStrom oraz Visual Studio Code - do tworzenia kodu aplikacji webowej
- Rozszerzenie do React i przeglądarki internetowej 'react dev tools' - do ułatwienia tworzenia apli-
kacji webowej i śledzenia przepływu i przetwarzania przez nią danych
- Android Studio - do tworzenia aplikacji mobilnej
- Git - do utrzymania projektu na repozytorium zdalnym
- Google Chrome - do testów manualnych aplikacji webowej
- urządzenie z systemem Android - do testów manualnych aplikacji mobilnej
- PgAdmin - GUI do bazy PostgreSQL do sprawdzania poprawności danych przechowywanych w
bazie