

**Politechnika Opolska**  
**Wydział Elektrotechniki, Automatyki i Informatyki**  
**Katedra Informatyki**



**POLITECHNIKA**  
**OPOLSKA**

**Autorzy projektu - Grupa Developerska P2D4N**

Filip Statkiewicz

Dawid Romanów

Jan Śliwak

Michał Owsiak

**Kierunek:** Informatyka

**Przedmiot:** Projekt zespołowy systemu informatycznego

**Rok akademicki:** 2023/2024

**Semestr:** IV

**Temat:**

Program typu Open Source do zarządzaniem barem

**Prowadzący przedmiot:**

Dr inż. Artur Pala

## Spis treści

1. STRONA TYTUŁOWA .....	1
2. SPIS TREŚCI .....	2
3. OPIS CELU PROJEKTU .....	3
4. WYMAGANIA FUNKCJONALNE .....	3
5. WYMAGANIA NIEFUNKCJONALNE.....	4
6. OPIS WYBRANEGO ŚRODOWISKA PROGRAMISTYCZNEGO .....	5
7. USER STORIES .....	7
8. NARZĘDZIA PRACY ZESPOŁOWEJ.....	12
9. PRODUCT BACKLOG.....	13

## 1. Opis celu projektu

Celem tego projektu jest stworzenie kompleksowego systemu POS (Point of Sale) dedykowanego pracownikom baru. System ma umożliwić obsługę procesu zamawiania, płacenia oraz zarządzania menu i stanami magazynowymi w sposób efektywny i intuicyjny. Projekt ma na celu dostarczenie narzędzia umożliwiającego personelowi baru szybką i sprawną obsługę klientów, optymalne zarządzanie zamówieniami i składnikami, a także generowanie raportów sprzedażowych i analitycznych. Dodatkowo, system ma umożliwiać konfigurację promocji i rabatów. Celem projektu jest również stworzenie intuicyjnego i łatwego w obsłudze interfejsu dla personelu baru, co przyczyni się do poprawy doświadczenia użytkownika i zadowolenia klientów.

Pracownik po zalogowaniu się do systemu zostaje przeniesiony do głównego menu, gdzie dostęp do kluczowych funkcji. Gdy klient zamawia drinka pracownik tworzy zamówienie wybierając odpowiedni napój z listy lub po wyszukaniu / filtrowaniu. System informuje o możliwych do zamówienia drinkach na podstawie dostępności składników. Pracownik ma również możliwość naliczenia rabatu. Po zakończeniu dodawania kolejnych produktów do zamówienia system wyświetla podsumowanie a pracownik wybiera odpowiednią wybraną przez klienta formę płatności. W przypadku płatności gotówką system wskazuje pracownikowi resztę do wydania. Po dokonaniu płatności system generuje paragon, aktualizuje stany magazynowe a dla barmana zostaje wyświetlona lista składników i przepis.

Osoby posiadające odpowiednie uprawnienia (kierownicy) mają możliwość edycji menu, cen, danych pracowników oraz przejrzenia automatycznie generowanych przez system raportów i analiz.

Od strony magazynowej system informuje pracowników o zbliżającym się upływie terminu ważności składników oraz o konieczności ich zamówienia na podstawie przewidywanego zużycia i aktualnego stanu.

## 2. Wymagania funkcjonalne

### **SPRZEDAŻ**

System musi pozwolić na:

- Logowanie i zarządzaniem czasem pracy
- Tworzenie, edycje i usunięcie zamówienia

- Wyszukiwanie i filtrowanie produktów na podstawie słów kluczowych i parametrów
- Możliwość sumowania zamówienia
- Obsługę płatności
- Drukowanie paragonu
- Podglądu przepisu

## **MAGAZYN**

System musi pozwolić na:

- Dodawanie, usuwanie, edytowanie, wyszukiwanie produktów
- Przydzielenie produktu do danej grupy towarowej
- Sortowanie produktów według wybranych parametrów
- Tworzenie dokumentów inwentaryzacyjnych
- Raportowanie o końcu terminu przydatności produktów
- Tworzenie, edycje, usunięcie, anulowanie zamówienia

## **RAPORTY I ANALIZY**

System musi pozwolić na:

- Analizę sprzedaży, na podstawie ilości sprzedanych produktów
- Analiza zadowolenia produktem na podstawie opinii o produkcie
- Analiza cen konkurencji, dostawców i przewoźników
- Analiza odczuć klientów do przedsiębiorstwa (opinie i propozycje usprawnienia)
- Generowanie raportów i wykresów na podstawie analiz

## **3. Wymagania niefunkcjonalne**

**Wydajność** – Aplikacja POS powinna być szybka i wydajna, aby obsługiwać transakcje sprzedaży w czasie rzeczywistym, zwłaszcza w okresach dużej aktywności w barze.

**Dostępność** - Aplikacja powinna być dostępna i stabilna przez większość czasu działania baru, aby uniknąć przestojów w obsłudze klientów.

**Bezpieczeństwo** - Zapewnienie wysokiego poziomu bezpieczeństwa danych finansowych

**Zgodność z przepisami** - Aplikacja musi być zgodna z lokalnymi i krajowymi przepisami i regulacjami dotyczącymi obsługi transakcji finansowych i danych osobowych.

**Skalowalność** - Aplikacja powinna być skalowalna, aby można było ją dostosować do różnych poziomów działalności baru, od małych pubów po duże kluby nocne.

**Integracje** - Wsparcie dla integracji z innymi systemami, takimi jak systemy zarządzania magazynem, kontroli dostępu itp.

**Obsługa różnych urządzeń** - Aplikacja powinna działać na różnych urządzeniach, w tym terminalach kasowych, tabletach i smartfonach.

#### 4. Opis wybranego środowiska programistycznego



**Microsoft Visual Studio** – zintegrowane środowisko programistyczne pozwalające na tworzenie oprogramowania konsolowego jak i z interfejsem na różne platformy – w Naszym przypadku wykorzystamy obsługę tworzenia aplikacji okienkowych WPF (Windows Presentation Foudation) dla platformy Windows.

Do największych zalet tego środowiska należą:

- bogate funkcje debugowania.
- wiele darmowych rozszerzeń.
- inteligentne podpowiedzi.



**Microsoft SQL Server** - kompleksowy system zarządzania bazą danych, mimo swojego zaawansowania pozwala on w prosty sposób tworzyć, modyfikować i usuwać bazy danych. Dla Naszego projektu kluczowe będzie przechowywanie sporych ilości informacji w bazie danych.

Do najważniejszych funkcji należą:

- integracja zewnętrzna (w naszym przypadku z Visual Studio).
- skalowalność i wydajność.
- obsługa relacyjnych baz danych.



**Figma** - oprogramowanie służące do projektowania prototypów, UX oraz UI kładące nacisk na pracę zespołową, także w czasie rzeczywistym. Umożliwia wygodne udostępnianie plików i różnych wersji szkiców oraz komentowanie. Są

to funkcje bardzo pomocne także podczas współpracy z klientem, gdyż projektant może szybko otrzymywać informacje zwrotne.

Cechy tego środowiska:

- Umożliwia projektowanie w czasie rzeczywistym, co oznacza, że wiele osób może współpracować nad projektem.
- Działa w przeglądarkach internetowych, dzięki czemu użytkownicy mogą dostęp do projektów i edycję na różnych urządzeniach.
- Można tworzyć biblioteki komponentów, które ułatwiają tworzenie spójnych i zharmonizowanych projektów. Komponenty można łatwo przeciągać i upuszczać w projekcie.
- Automatycznie śledzi historię projektu, co ułatwia cofanie się do wcześniejszych wersji projektu.

## **5. User Stories**

### **Logowanie i zarządzanie czasem pracy**

Jako barman/kierownik baru, chciałbym móc zalogować się do systemu i zarządzać moim czasem pracy, aby skutecznie monitorować moje godziny pracy.

Warunki satysfakcji:

- System umożliwia zalogowanie się za pomocą unikalnych danych uwierzytelniających.
- System umożliwia rejestrację godzin pracy oraz generuje raporty z przepracowanych godzin.

### **Tworzenie, edycja i usunięcie zamówienia**

Jako barman, chciałbym móc tworzyć, edytować i usuwać zamówienia, aby obsługiwać klientów w barze.

Warunki satysfakcji:

- System umożliwia dodawanie produktów do zamówienia, modyfikowanie ich i usuwanie.
- Wszystkie zmiany w zamówieniach są od razu zapisywane.

## **Wyszukiwanie i filtrowanie produktów**

Jako barman, chciałbym móc łatwo znaleźć produkty w systemie za pomocą słów kluczowych i parametrów, aby obsługiwać zamówienia klientów.

Warunki satysfakcji:

- System umożliwia szybkie wyszukiwanie produktów za pomocą słów kluczowych.
- System umożliwia filtrowanie produktów według różnych parametrów, takich jak rodzaj alkoholu, smak, itp.

## **Możliwość sumowania zamówienia**

Jako barman, chciałbym mieć możliwość podsumowania zamówienia, aby łatwo informować klientów o łącznej kwocie do zapłacenia.

Warunki satysfakcji:

- System wyświetla łączną sumę zamówienia w czasie rzeczywistym podczas tworzenia zamówienia.

## **Obsługa płatności**

Jako barman, chciałbym obsługiwać płatności klientów za zamówienia, aby ułatwić transakcje.

Warunki satysfakcji:

- System umożliwia różne metody płatności, takie jak gotówka, karty kredytowe itp.
- Po dokonaniu płatności, zamówienie zostaje zaktualizowane jako opłacone.



## **Podgląd przepisu**

Jako barman, chciałbym mieć dostęp do przepisów na przygotowanie napojów, aby być w stanie obsłużyć klientów zgodnie z recepturami baru.

Warunki satysfakcji:

- System umożliwia dostęp do listy przepisów i instrukcji na przygotowanie napojów.

## **Dodawanie, usuwanie, edytowanie, wyszukiwanie produktów**

Jako kierownik baru, chciałbym móc zarządzać produktem w magazynie, w tym dodawać, usuwać, edytować i wyszukiwać produkty.

Warunki satysfakcji:

- System umożliwia dodawanie nowych produktów do magazynu, usuwanie istniejących i edytowanie danych produktu.
- System umożliwia szybkie wyszukiwanie produktów w magazynie.

## **Przydzielenie produktu do danej grupy towarowej**

Jako kierownik baru, chciałbym mieć możliwość przypisywania produktów do odpowiednich grup towarowych, aby łatwiej zarządzać magazynem.

Warunki satysfakcji:

- System umożliwia przypisanie produktów do określonych kategorii i grup towarowych.

## **Sortowanie produktów według wybranych parametrów**

Jako kierownik baru, chciałbym móc sortować produkty według różnych parametrów, aby lepiej zorganizować magazyn.

Warunki satysfakcji:

- System umożliwia sortowanie produktów według kategorii, daty ważności, dostawców itp.

### **Tworzenie dokumentów inwentaryzacyjnych**

Jako kierownik baru, chciałbym móc tworzyć dokumenty inwentaryzacyjne, aby śledzić stan magazynu.

Warunki satysfakcji:

- System umożliwia tworzenie dokumentów inwentaryzacyjnych, które odzwierciedlają stan magazynu w określonym czasie.

### **Analiza sprzedaży na podstawie ilości sprzedanych produktów**

Jako kierownik baru, chciałbym być w stanie analizować sprzedaż na podstawie ilości sprzedanych produktów, aby podejmować strategiczne decyzje.

Warunki satysfakcji:

- System generuje raporty z analizą sprzedaży, w tym informacje o ilości sprzedanych produktów.

### **Generowanie raportów i wykresów na podstawie analiz**

Jako kierownik baru, chciałbym móc generować raporty i wykresy na podstawie analiz sprzedaży, opinii klientów i innych danych, aby podejmować informowane decyzje.

Warunki satysfakcji:

- System umożliwia generowanie raportów i wykresów na podstawie zgromadzonych danych i analiz.

## **Wydajność**

Jako użytkownik systemu POS, chciałbym, aby aplikacja była szybka i wydajna, aby obsługiwać transakcje sprzedaży w czasie rzeczywistym, zwłaszcza w okresach dużej aktywności w barze.

Warunki satysfakcji:

- Aplikacja obsługuje transakcje sprzedaży w czasie rzeczywistym bez opóźnień.
- Odpowiedź aplikacji jest natychmiastowa, nawet w okresach dużej aktywności.

## **Dostępność**

Jako użytkownik systemu POS, chciałbym, aby aplikacja była dostępna i stabilna przez większość czasu działania baru, aby uniknąć przestojów w obsłudze klientów.

Warunki satysfakcji:

- Aplikacja jest dostępna przez większość czasu działania baru, minimalizując przestoje w obsłudze.
- System jest stabilny i rzadko ulega awariom.

## **Integracje**

Jako użytkownik systemu POS, chciałbym, aby aplikacja wspierała integracje z innymi systemami, takimi jak systemy zarządzania magazynem, kontroli dostępu itp.

Warunki satysfakcji:

- Aplikacja umożliwia integrację z różnymi systemami, aby ułatwić zarządzanie i kontrolę wszystkimi aspektami działalności baru.

## **Obsługa różnych urządzeń**

Jako użytkownik systemu POS, chciałbym, aby aplikacja działała na różnych urządzeniach, w tym terminalach kasowych, tabletach i smartfonach.

Warunki satysfakcji:

- Aplikacja jest dostępna i działa sprawnie na różnych urządzeniach, zapewniając elastyczność obsługi.

## 6. Narzędzia pracy zespołowej



**Jira** - to popularne narzędzie do zarządzania projektami i śledzenia zadań, stworzone przez firmę Atlassian.

Oto główne cechy Jiry w formie podpunktów:

- Pozwala użytkownikom tworzyć zadania, historie użytkownika i błędy, a następnie śledzić postęp ich realizacji.
- Umożliwia tworzenie projektów i przypisywanie zadań do projektów, co ułatwia organizację pracy zespołu.
- Obsługuje zarówno metodyki pracy w metodyce Kanban, jak i Scrum, co umożliwia zespołom dostosowanie procesu pracy do swoich potrzeb.



**Discord** - bezpłatna usługa internetowa oparta na chmurze, służąca do rozmów głosowych i komunikacji za pomocą wiadomości tekstowych z możliwością umieszczania zdjęć oraz filmów, zaprojektowana początkowo jako platforma skierowana dla graczy komputerowych.

Cechy jakimi się wyróżnia:

- Umożliwia użytkownikom prowadzenie rozmów głosowych i tekstowych w czasie rzeczywistym. Użytkownicy mogą tworzyć serwery i kanały, na których mogą prowadzić rozmowy.
- Użytkownicy mogą tworzyć własne serwery lub dołączać do istniejących społeczności. Na serwerach można tworzyć różne kanały tematyczne, co ułatwia organizację rozmów.
- Obsługuje różne rodzaje treści multimedialnych, takie jak obrazki, filmy i pliki dźwiękowe, co sprawia, że jest bardziej interaktywny.
- Umożliwia streamowanie treści w czasie rzeczywistym i udostępnianie ekranu, co jest przydatne podczas prowadzenia wideokonferencji, gier lub demonstracji.

## 7. Product backlog

1. Utworzenie projektowego serwera Discord do komunikacji między sobą.
2. Utworzenie projektu zespołowego na platformie Jira.
3. Utworzenie projektu zespołowego na platformie Github.
4. Zaprojektowanie głównego okna interfejsu graficznego programu.
5. Zaprojektowanie interfejsu graficznego panelu sprzedaży.

6. Zaprojektowanie interfejsu graficznego panelu logowania i rozpoczęcia, zakończenia czasu pracy.
7. Zaprojektowanie interfejsu graficznego funkcji magazynowych.
8. Zaprojektowanie interfejsu graficznego systemu raportów i analiz
9. Zaprojektowanie interfejsu graficznego funkcji zarządzania kadrą pracowników
10. Zaprojektowanie bazy danych produktów w menu
11. Zaprojektowanie bazy danych stanu magazynowego
12. Zaprojektowanie bazy danych przepisów
13. Zaprojektowanie bazy danych pracowników
14. Zaprojektowanie bazy danych raportów i analiz
15. Implementacja głównego okna interfejsu graficznego programu.
16. Implementacja interfejsu graficznego panelu sprzedaży.
17. Implementacja interfejsu graficznego panelu logowania i rozpoczęcia, zakończenia czasu pracy.
18. Implementacja interfejsu graficznego funkcji magazynowych.
19. Implementacja interfejsu graficznego systemu raportów i analiz
20. Implementacja interfejsu graficznego funkcji zarządzania kadrą pracowników
21. Implementacja bazy danych
22. Implementacja systemu logowania i zarządzaniem czasem pracy
23. Implementacja systemu tworzenia, edycji i usunięcia zamówienia
24. Implementacja możliwości użycia opcji rabatowej (np. -15%)
25. Implementacja funkcji sumującej zamówienia (cena, ilość)
26. Implementacja systemu płatności
27. Implementacja systemu drukowania paragonu
28. Implementacja możliwości podglądu przepisu
29. Implementacja wyszukiwarki do produktów
30. Implementacja funkcji zarządzania kadrą i uprawnieniami
31. Implementacja systemu zarządzania zapasami w magazynie
32. Implementacja funkcji filtracji
33. Implementacja systemu tworzenia, edycji, zmian przepisów
34. Implementacja systemu zarządzania produktami dostępnymi w panelu sprzedaży
35. Implementacja funkcji generowania i wyświetlania raportów sprzedaży
36. Implementacja funkcji generowanie raportu zużycia produktów
37. Implementacja funkcji sprawdzenia stanu kasy fiskalnej
38. Implementacja generowania wykazu czasu pracy pracowników
39. Implementacja generowania wykazu produktywności pracowników na podstawie ilości wykonanych zamówień
40. Implementacja funkcji generowania analizy popularności produktów.
41. Implementacja funkcji tworzenia zamówień magazynowych

- 42. Integracja systemu magazynowego z systemem punktu sprzedaży
- 43. Implementacja systemu alertów i przypomnień, np. o terminie ważności produktów
- 44. Implementacja funkcji historii zamówień