Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: restauracja Subway

Autorzy: **Klaudia Wrocławska & Maurycy Demidowicz**

Grupa: I1-221B

Kierunek: informatyka

Rok akademicki 2022/23

Poziom i semestr: I/4

Tryb studiów: stacjonarne

Należy pozostawić wszelkie nagłówki tego dokumentu, a umieszczać treść w odpowiednich miejscach zamiast obecnych objaśnień.

Stronę tytułową można sformatować w dowolny sposób, ale należy pozostawić zawartość informacyjną w układzie pokazanym powyżej.

Praca powinna zostać złożona wyłącznie w formacie pdf. Przed wygenerowaniem ostatecznej wersji należy zaktualizować spis treści – wyświetlane dwa poziomy.

Niniejszą informację należy również usunąć z wersji końcowej.

1. Spis treści

[2 Odnośniki do innych źródeł 4](#_Toc1976793)

[3 Słownik pojęć 5](#_Toc1976794)

[4 Wprowadzenie 6](#_Toc1976795)

[4.1 Cel dokumentacji 6](#_Toc1976796)

[4.2 Przeznaczenie dokumentacji 6](#_Toc1976797)

[4.3 Opis organizacji lub analiza rynku 6](#_Toc1976798)

[4.4 Analiza SWOT organizacji 6](#_Toc1976799)

[5 Specyfikacja wymagań 7](#_Toc1976800)

[5.1 Charakterystyka ogólna 7](#_Toc1976801)

[5.2 Wymagania funkcjonalne 7](#_Toc1976802)

[5.3 Wymagania niefunkcjonalne 8](#_Toc1976803)

[6 Zarządzanie projektem 9](#_Toc1976804)

[6.1 Zasoby ludzkie 9](#_Toc1976805)

[6.2 Harmonogram prac 9](#_Toc1976806)

[6.3 Etapy/kamienie milowe projektu 9](#_Toc1976807)

[7 Zarządzanie ryzykiem 10](#_Toc1976808)

[7.1 Lista czynników ryzyka 10](#_Toc1976809)

[7.2 Ocena ryzyka 10](#_Toc1976810)

[7.3 Plan reakcji na ryzyko 10](#_Toc1976811)

[8 Zarządzanie jakością 11](#_Toc1976812)

[8.1 Scenariusze i przypadki testowe 11](#_Toc1976813)

[9 Projekt techniczny 12](#_Toc1976814)

[9.1 Opis architektury systemu 12](#_Toc1976815)

[9.2 Technologie implementacji systemu 12](#_Toc1976816)

[9.3 Diagramy UML 12](#_Toc1976817)

[9.4 Charakterystyka zastosowanych wzorców projektowych 12](#_Toc1976818)

[9.5 Projekt bazy danych 12](#_Toc1976819)

[9.6 Projekt interfejsu użytkownika 12](#_Toc1976820)

[9.7 Procedura wdrożenia 13](#_Toc1976821)

[10 Dokumentacja dla użytkownika 14](#_Toc1976822)

[11 Podsumowanie 15](#_Toc1976823)

[11.1 Szczegółowe nakłady projektowe członków zespołu 15](#_Toc1976824)

[12 Inne informacje 16](#_Toc1976825)

# Odnośniki do innych źródeł

tj. do wykorzystywanych narzędzi / projektów w tych narzędziach

Zarządzania projektem: – Jira, Trello, itp.

Wersjonowanie kodu: <https://github.com/demidowicz/Inzynieria_oprogramowania.git>

System obsługi defektów: – np. Bitbucket, Github, Bugzilla.

# Słownik pojęć

Tabela lub lista z pojęciami, które wymagają wyjaśnienia, wraz z tymi wyjaśnieniami – w szczególności synonimy różnych pojęć używanych w dokumentacji.

# Wprowadzenie

## Cel dokumentacji

po co ją robimy i co zawiera (poziom szczegółowości)

## Przeznaczenie dokumentacji

Dla pracowników restauracji obsługujących system.

## Opis organizacji lub analiza rynku

Firma D&W jest właścicielem restauracji znajdującej się pod franczyzą Subway. Znajduje się ona w galerii Galaxy (al. Wyzwolenia 18/20, 70-554 Szczecin) i jest otwarta dla klientów od 9 do 21 w ciągu tygodnia, a w niedzielę od 12 do 20.

Subway jest franczyzą barów szybkiej obsługi, specjalizującej się w produkcji kanapek, sałatek i wrapów. Sprzedawane produkty są przygotowywane na miejscu. Poszczególne restauracje są od siebie niezależnie.

Kierownik wpuszcza pracowników godzinę przed otwarciem. W tym czasie przygotowują oni lokal oraz jedzenie. Zmiana następuje o godzinie 15. Pracownicy drugiej zmiany dopilnowują aby restauracja była czysta po zamknięciu. W razie nagłych wypadków i usterek przy pracy, kierownik może zadzwonić do szefa. Włącznie w restauracji pracuje około 10 osób.

Klient składa zamówienie, wybierając produkt, poszczególne składniki oraz dodatki. Osoba obsługująca ladę odbiera zamówienie oraz je realizuje. Jeżeli w trakcie dnia zabraknie składników na konkretne zamówienie, zostaje ono tymczasowo usunięte z menu. Zamówienia mogą być składane na miejscu lub na wynos. W zależności od dnia, lokal realizuje od 100 do 300 zamówień.

Restauracja zamawia produkty spożywcze oraz inne potrzebne produkty na koniec dnia od firmy zewnętrznej, po zrobieniu inwentaryzacji i ocenie, ile produktów będzie potrzebnych na dzień następny. Na podstawie ilości sprzedaży są obliczane zyski.

## Analiza SWOT organizacji

|  |  |
| --- | --- |
| **SILNE STRONY** | **SŁABE STRONY** |
| * Duża rozpoznawalność marki * Lokalizacja w centrum miasta * Wysoka jakość używanych składników | * Dużo zatrudnionych studentów – ciągle zmieniająca się kadra * Mały lokal * Małe doświadczenie w gastronomii |
| **SZANSE** | **ZAGROŻENIA** |
| * Rosnąca popularność zdrowego jedzenia * Popularność zamówień z dowozem | * Rosnące ceny energii elektrycznej * Rosnące koszty prowadzenia działalności gospodarczej |

# Specyfikacja wymagań

## Charakterystyka ogólna

### Definicja produktu

SubSystem – system zarządzania zasobami sprzedaży restauracji Subway.

### Podstawowe założenia

System SubSystem będzie służył do zapisywania transakcji oraz informacji o sprzedaży. System nie będzie zapisywał informacji o kliencie, tylko o sprzedawanym produkcie. Dodatkowo system będzie monitorował stan magazynu.

Pracownik będzie zapisywał w systemie informacje o transakcjach i na ich podstawie system będzie obliczał premię od sprzedaży. Szef oraz kierownik będą mieli wgląd do tych informacji. Szef będzie w stanie ustalać próg i wielkość premii.

Przy każdej transakcji pracownik będzie zapisywał w systemie informacje o sprzedawanych produktach. Na koniec każdego miesiąca system będzie tworzył raport dotyczący informacji o sprzedaży.

Pracownik pierwszej zmiany będzie zapisywał w systemie produkty dostarczone danego dnia, przed otwarciem restauracji. Na koniec drugiej zmiany pracownik będzie zapisywał ilość zużytych produktów danego dnia. Na podstawie tych informacji system, po każdej zmianie danych, będzie aktualizował informacje o stanie magazynu. System będzie obliczał ilość produktów do zamówienia.

### Cel biznesowy

1. Zautomatyzowanie obliczania premii pracowników
2. Zautomatyzowanie wytwarzania raportów o sprzedaży
3. Zautomatyzowanie obliczania potrzebnych zasobów

### Użytkownicy

1. Szef
2. Kierownik
3. Pracownik

### Korzyści z systemu

1. Szef   
   1.1 brak błędów przy obliczaniu premii
2. Kierownik

2.1 nie musi tworzyć raportu

1. Pracownik

3.1 nie musi ręcznie liczyć stanu magazynu

### Ograniczenia projektowe i wdrożeniowe

Baza danych – przechowywanie informacji o transakcjach, produktach dostępnych do sprzedaży oraz stanie magazynu

## Wymagania funkcjonalne

### Lista wymagań

Każdy użytkownik powinien mieć dostęp do informacji o stanie magazynu

Każdy użytkownik powinien posiadać unikalny login

Każdy użytkownik powinien posiadać unikalne hasło dostępu

Pracownik powinien móc zapisywać informacje o płatnościach klientów

Pracownik powinien móc wpisywać ilość dostarczonych produktów

Pracownik powinien móc wpisywać ilość dostępnych gotowych przekąsek

Pracownik powinien móc wpisywać ilość zużytych produktów

Szef powinien móc ustalać próg premii od sprzedaży

Szef powinien mieć dostęp do informacji o premii od sprzedaży

Szef powinien mieć dostęp do informacji o raporcie o sprzedaży

Szef powinien mieć dostęp do informacji o płatnościach klientów

Kierownik powinien mieć dostęp do informacji o raporcie o sprzedaży

Kierownik powinien mieć dostęp do informacji o premii od sprzedaży

Kierownik powinien mieć dostęp do informacji o płatnościach klientów

System powinien móc tworzyć raport o sprzedaży

System powinien obliczać ilość produktów do zamówienia

System powinien obliczać premię od sprzedaży

System powinien obliczać stan magazynu

### Diagramy przypadków użycia

Tutaj same diagramy – bez specyfikacji, ale każdy diagram z tytułem i na osobnej stronie

//visual paradigm → wkleić przed punktem kontrolnym

### Szczegółowy opis wymagań

7 scenariuszy → najważniejsze przypadki użycia

każde na nowej stronie wg następujących punktów:

* Numer – jako ID
* Nazwa
* Uzasadnienie biznesowe – odwołanie (-a) do elementów wymienionych w 5.1.5. (id i treść elementu, do którego się odwołujemy)
* Użytkownicy
* Scenariusze, dla każdego z nich:
* Nazwa scenariusza
* Warunki początkowe
* **Przebieg działań –** numerowana lista kroków, ze wskazaniem, kto realizuje dany krok
* Efekty – warunki końcowe
* Wymagania niefunkcjonalne – szczegółowe wobec poszczególnych wymagań funkcjonalnych
* Częstotliwość - na skali 1-5 lub BN-BW
* Istotność – inaczej: zależność krytyczna, znaczenie - na skali 1-5 lub BN-BW

***Ważne!***

*Elementy od warunków początkowych do końca mogą być grupowane, tj. specyfikacja pojedynczego przypadku użycia może zawierać:*

*- pojedynczy przebieg działań (scenariusz główny) oraz ew. scenariusze alternatywne, albo*

*- wiele przebiegów głównych wraz z ew. scenariuszami alternatywnymi – wtedy każdy z przebiegów głównych powinien być opisany wg tych punktów (od warunków początkowych do końca).*

## Wymagania niefunkcjonalne

W odniesieniu do całego systemu, modułów lub innych składowych systemu

1. Wydajność – w odniesieniu do konkretnych sytuacji – funkcji systemu

Zapisywanie informacji o płatnościach nie powinno zajmować dłużej niż 1s

1. Bezpieczeństwo – utrata, zniszczenie danych, zniszczenie innego systemu przez nasz – wraz z działaniami zapobiegawczymi i ograniczającymi skutki

Kopia raportu o sprzedaży powinna zostać zapisywana poza systemem

1. Zabezpieczenia

Dane dostępu każdego użytkownika powinny być bezpiecznie przechowywane

1. Inne cechy jakości – najlepiej ilościowo, żeby można było zweryfikować (zmierzyć) – adaptowalność, dostępność, poprawność, elastyczność, łatwość konserwacji, przenośność, awaryjność, testowalność, użyteczność

# Zarządzanie projektem

## Zasoby ludzkie

(rzeczywiste lub hipotetyczne) – przy realizacji projektu

Należy założyć, że projekt byłby realizowany w całości jako projekt komercyjny a nie tylko częściowo w ramach zajęć na uczelni

## Harmonogram prac

Etapy mogą się składać z zadań.

Wskazać czasy trwania poszczególnych etapów i zadań – wykres Gantta.

obejmuje również harmonogram wdrożenia projektu – np. szkolenie, rozruch, konfiguracja, serwis – może obejmować różne wydania (tj. o różnej funkcjonalności – personal, professional, enterprise) i wersje (1.0, 1.5, itd.)

## Etapy/kamienie milowe projektu

dla głównych etapów projektu

# Zarządzanie ryzykiem

## Lista czynników ryzyka

Wypełniona lista kontrolna

## Ocena ryzyka

prawdopodobieństwo i wpływ

## Plan reakcji na ryzyko

Działania w odniesieniu do poszczególnych ryzyk.

Mogą być wg różnych strategii, tj. kilka strategii dla pojedynczego czynnika ryzyka

# Zarządzanie jakością

## Scenariusze i przypadki testowe

szczegółowy plan testowania systemu – głównie testowanie funkcjonalności; każdy scenariusz od nowej strony, musi zawierać co najmniej następujące informacje (sugerowany układ tabelaryczny, np. wg szablonu podanego w osobnym pliku lub na wykładzie):

* numer – jako ID
* nazwa scenariusza – co test w nim testowane (max kilka wyrazów)
* kategoria – poziom/kategoria testów
* opis – dodatkowe opcjonalne informacje, które nie zmieściły się w nazwie
* tester - konkretna osoba lub klient/pracownik,
* termin – kiedy testowanie ma być przeprowadzane,
* narzędzia wspomagające – jeśli jakieś są używane przy danym scenariuszu
* przebieg działań – tabela z trzema kolumnami: lp. oraz opisującymi działania testera i systemu
* założenia, środowisko, warunki wstępne, dane wejściowe – przygotowanie przed uruchomieniem testów
* zestaw danych testowych – najlepiej w formie tabelarycznej – jakie konkretnie dane mają być użyte przez testera i zwrócone przez system w poszczególnych krokach przebiegu działań
* *przebieg lub zestaw danych testowych musi zawierać jawną informację o warunku zaliczenia testu*

# Projekt techniczny

## Opis architektury systemu

z ew. rysunkami pomocniczymi

## Technologie implementacji systemu

tabela z listą wykorzystanych technologii, każda z uzasadnieniem

## Diagramy UML

każdy diagram ma mieć tytuł oraz ma być na osobnej stronie

diagramy przypadków użycia umieszczone w punkcie 5.2.2, a nie tutaj.

### Diagram(-y) klas

1 lub więcej

### Diagram(-y) czynności

1 lub więcej

### Diagramy sekwencji

co najmniej 5, w tym co najmniej 1 przypadek użycia zilustrowany kilkoma diagramami

### Inne diagramy

co najmniej trzy – komponentów, rozmieszczenia, maszyny stanowej itp.

## Charakterystyka zastosowanych wzorców projektowych

informacja opisowa wspomagana diagramami (odsyłaczami do diagramów UML); jeśli wykorzystano wzorce projektowe, to należy wykazać dwa z nich

## Projekt bazy danych

### Schemat

w trzeciej formie normalnej; jeśli w innej to umieć uzasadnić wybór

### Projekty szczegółowe tabel

w zależności, czy następujące elementy są widoczne na schemacie b.d.: nazwa tabeli, nazwy pól, typ danych, wartości NULL, klucz główny, klucz obcy –

- jeśli TAK: i nie ma potrzeby pokazania dodatkowych elementów b.d., to ten punkt może być pusty,

- jeśli NIE: to podać te elementy, których nie widać na schemacie.

dodatkowymi elementami mogą być np. triggery, procedury, funkcje, indeksy, użytkownicy, role.

## Projekt interfejsu użytkownika

co najmniej dla głównej funkcjonalności programu – w razie wątpliwości, uzgodnić z prowadzącym zajęcia

### Lista głównych elementów interfejsu

okien, stron, aktywności (Android)

### Przejścia między głównymi elementami

np. storyboard, schemat blokowy lub inna notacja

### Projekty szczegółowe poszczególnych elementów

dla 5-7 głównych elementów

każdy element od nowej strony z następującą minimalną zawartością:

* numer – ID elementu
* nazwa – np. formularz danych produktu
* projekt graficzny – wystarczy schemat w narzędziu graficznym lub zrzut ekranu – z przykładowymi informacjami (nie pusty!!!)
* opcjonalnie:
* opis – dodatkowe opcjonalne informacje o przeznaczeniu, obsłudze – jeśli nazwa nie będzie wystarczająco czytelna
* wykorzystane dane – jakie dane z bazy danych są wykorzystywane
* opis działania – tabela pokazująca m.in. co się dzieje po kliknięciu przycisku, wybraniu opcji z menu itp.

## Procedura wdrożenia

jeśli informacje w harmonogramie nie są wystarczające (a zapewne nie są)

# Dokumentacja dla użytkownika

Opcjonalnie – dla chętnych

Na podstawie projektu docelowej aplikacji, a nie zaimplementowanego prototypu architektury

4-6 stron z obrazkami (np. zrzuty ekranowe, polecenia do wpisania na konsoli, itp.)

* pisana językiem odpowiednim do grupy odbiorców – czyli najczęściej nie do informatyków
* może to być przebieg krok po kroku obsługi jednej głównej funkcji systemu, kilku mniejszych, instrukcja instalacji lub innej pomocniczej czynności.

# Podsumowanie

## Szczegółowe nakłady projektowe członków zespołu (10 – 15 podpunktów)

tabela (kolumny to osoby, wiersze to działania) pokazująca, kto ile czasu poświęcił na projekt oraz procentowy udział każdej osoby w danym zadaniu oraz wiersz podsumowania – procentowy udział każdej osoby w skali całego projektu

|  |  |  |
| --- | --- | --- |
| ZADANIE | MAURYCY DEMIDOWICZ | KLAUDIA WROCŁAWSKA |
|  |  |  |

# Inne informacje

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach