Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: **System zarządzania cenami i promocjami dla hipermarketów**

Autorzy: **Daniel Marcinowski, Adam Wasiak**

Grupa: I1-220A

Kierunek: informatyka

Rok akademicki: 2019/2020

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.

VISUAL PARADIGM LUB INNY – PROGRAM DO WYKRESÓW I DIAGRAMÓW

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

* + Zarządzania projektem – sugerowane JazzHub
  + Wersjonowanie kodu – sugerowany Git (hosting np. na Bitbucket lub Github), ew. SVN
  + System obsługi defektów – np. Bitbucket, JazzHub

# Słownik pojęć

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

Konfiguracja – obecny stan w bazie danych, listy produktów i ich właściwości, np. ceny. Konfiguracja taka może być wczytana i zapisana.`

Właściciel – Też osoba upoważniona przez właściciela (w przypadku diagramów)

# Wprowadzenie

## Cel dokumentacji

Dokumentacja ta jest tworzona po to, aby użytkownik mógł zapoznać się ze wszystkimi możliwościami i technikaliami systemu. Będzie zapewne przydatnym odniesieniem, gdy użytkownik się zgubi.

## Przeznaczenie dokumentacji

Dokumentacja przeznaczona jest dla użytkownika systemu – najpewniej dla kierownika danego budynku czy sieci sklepów, aby mógł ustalać promocje i zmieniać ceny. Nie jest ona dla potencjalnych klientów, jako że nie będą oni wpływać na ceny produktów (nie chcemy, żeby ktoś na czas zakupu ustawił cenę zakupionych produktów na 0,00zł, prawda?)

## Opis organizacji lub analiza rynku

System ten będzie przydatny dla wszystkich właścicieli sklepów – popyt powinien być bardzo duży, jako że ogólna przydatność systemu jest wystarczająca, aby go zakupić oraz potencjalna audiencja jest bardzo szeroka – czy to potężny moloch, czy losowy spożywczak za rogiem, wszędzie przyda się zautomatyzowany system zarządzania cenami, promocjami oraz podgląd statystyk dt. Sprzedawalności produktów. Na każdy sklep przypada jeden właściciel, który może ten system zakupić. Dopóki ilość sklepów będzie rosnąć, ilość potencjalnych klientów będzie rosnąć.

## Analiza SWOT organizacji

|  |  |  |  |
| --- | --- | --- | --- |
| Mocne strony | Słabe strony | Szanse | Zagrożenia |
| Uniwersalne zastosowanie, trafia do szerokiej audiencji | Szeroka audiencja sprawi, że nie będziemy mierzyć w żadną konkretną stronę, przez co produkt może nie być wyspecjalizowany dla potencjalnego klienta (możliwe wprowadzenie wyspecjalizowanych potrzeb jako pakietów do pobrania i zainstalowania) | Wprowadzenie kilku dodatków sprawi, że będziemy odstawać na tle innych producentów proponujących podobne systemy. | Ktoś może wpaść na ten sam pomysł i nas wygonić  Potężne firmy mogą nie chcieć dodatkowej konkurencji |

# Specyfikacja wymagań

## Charakterystyka ogólna

### Definicja produktu

jedno zdanie o systemie

### Podstawowe założenia

do czego będzie służył ten system – kilka/kilkanaście zdań wprowadzających

### Cel biznesowy

co organizacja docelowa chce osiągnąć wdrażając system

### Użytkownicy

lista – ew. wyjaśnienia dodać do słownika pojęć

### Korzyści z systemu

dla poszczególnych grup użytkowników – każdy element z unikalnym numerem identyfikacyjnym

### Ograniczenia projektowe i wdrożeniowe

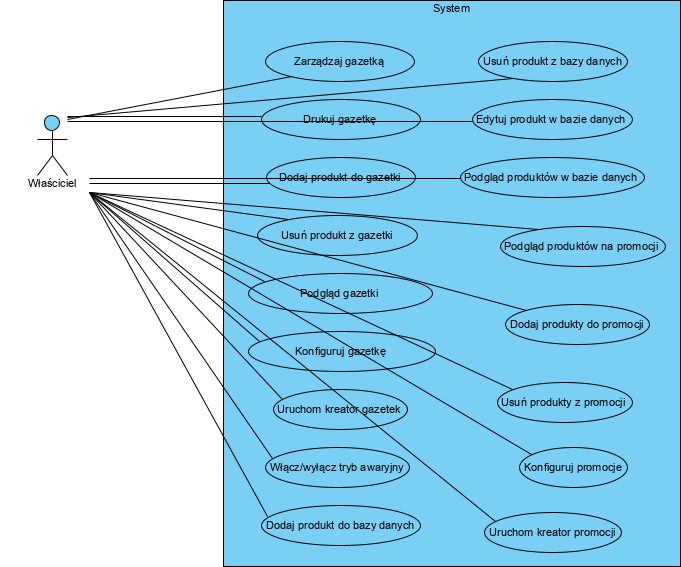
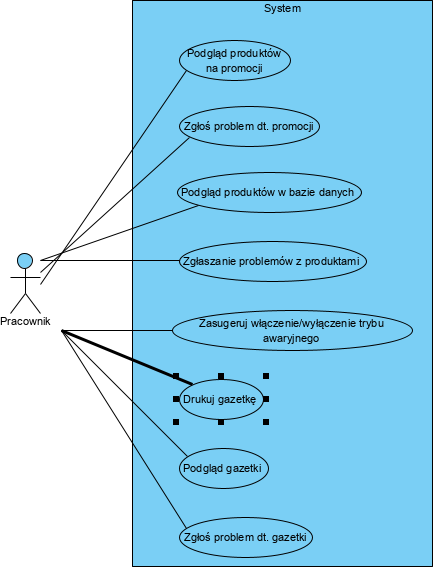
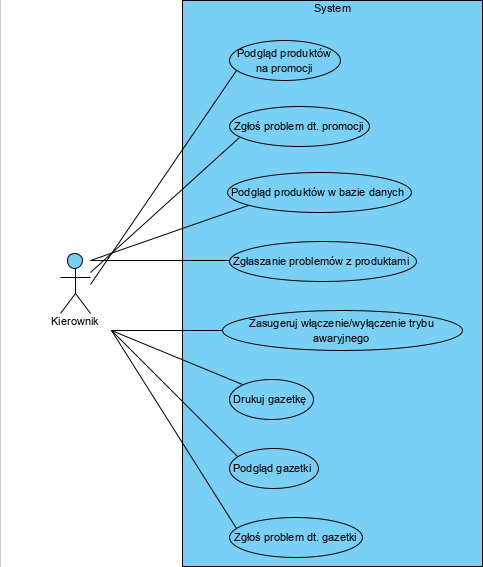
przepisy prawne, specyficzne technologie, narzędzia, b.d., protokoły komunikacyjne, aspekty zabezpieczeń, zgodność ze standardami, powiązania z innymi aplikacjami, platforma sprzętowa, system operacyjny, inne komponenty niezbędne do współpracy – wszystko wraz z uzasadnieniem!

## Wymagania funkcjonalne

### Lista wymagań

lista numerowana – czyli lista przypadków użycia lub bardziej ogólnie sformułowane wymagania

### Diagramy przypadków użycia



### Szczegółowy opis wymagań

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:
* 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

wobec całego systemu

1. Wydajność – w odniesieniu do konkretnych sytuacji – funkcji systemu
2. Bezpieczeństwo – utrata, zniszczenie danych, zniszczenie innego systemu przez nasz – wraz z działaniami zapobiegawczymi i ograniczającymi skutki
3. Zabezpieczenia
4. Inne cechy jakości – najlepiej ilościowo, żeby można było zweryfikować (zmierzyć) – adaptowalność, dostępność, poprawność, elastyczność, łatwość konserwacji, przenośność, awaryjność, testowalność, użyteczność

1.

2. Regularne tworzenie obrazów konfiguracji.

3. System logowania, mechanizmy szyfrowania, tryb awaryjny, podczas którego w systemie nie mogą zostać wprowadzone żadne zmiany i jest ustawiona konfiguracja domyślna.

4.

# 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

Jedna osoba do zarządzania projektem, kilku programistów, jeden ankietant, może jeden charyzmatyczny promotor systemu.

## Harmonogram prac

## Etapy/kamienie milowe projektu

1. Szkielet

2. Podstawowa obsługa jednego systemu, np. zarządzanie bazą – dodawanie, podgląd, edycja i usuwanie produktów.

3. Stworzenie interfejsu

4. Dodawanie kolejnych programów:

4.1. Podgląd statystyk

4.2. Drukowanie gazetek

4.3. Automatyczne generowanie promocji wg ustalonych kryteriów

5. Dalsze naprawianie błędów, które mogły umknąć przy wypuszczeniu projektu (np. z powodu deadline’ów)

# Zarządzanie ryzykiem

## Lista czynników ryzyka

1. Nie jesteśmy jedyni, którzy mają taki pomysł.

2. Nieczyste zagrywki ze strony konkurencji.

3. Zmiana prawa.

4. Zmiana trendów.

5. Brak wzrostu ilości sklepów.

## Ocena ryzyka

1. Nie jesteśmy jedyni, ale możemy wejść w rynek dzięki dodatkowym funkcjom systemu.
2. Duże, zachodnie firmy wykorzystają wszystkie kruczki prawne, a może i łapówki, żeby nas przegonić i nie dopuścić nas do rynku. Może być to największe zagrożenie, z jakim się zmierzymy.
3. Prawo jest bardzo płynne – nigdy nie wiadomo, czy nie będziemy musieli czegoś zmienić, ale raczej zmiany w prawie nie powinny nas zmusić do zerwania projektu.
4. Nigdy nie wiadomo, co będzie modne za tydzień, miesiąc czy rok. Musimy przewidzieć, czy dalej będzie zapotrzebowanie oraz na co. Nie powinno to jednak zmienić toru pracy, raczej kilka odgałęzień, by sprostać potrzebom.
5. Jeżeli sklepy nie będą powstawać, nie będziemy mieli nowych klientów.

## Plan reakcji na ryzyko

1. Dzięki dodatkowym funkcjom, możemy się wybić z motłochu.
2. Dużo pieniędzy w kieszeni pozwoli wynająć dobrych prawników, gdy najdzie taka potrzeba. Zaczynamy pracę, jeśli upewnimy się, że mamy pieniędzy na utrzymanie przynajmniej trzech rozpraw.
3. Dostosować się do nich w razie potrzeby.
4. Podobnie jak wyżej.
5. Zawsze można wprowadzić licencję – nawet, jeśli niską, da to trwały przypływ pieniędzy.

# 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

## 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

### Diagram(-y) czynności

### Diagramy sekwencji

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

### 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

## Projekt interfejsu użytkownika

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

### Lista głównych elementów interfejsu

Pasek z kartami (tutaj przełączamy m-dzy wszystkimi pożądanymi funkcjami systemu)

Ekran karty (tutaj będzie zawartość wybranej karty)

### Przejścia między głównymi elementami

### Projekty szczegółowe poszczególnych 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

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 – udział każdej osoby w skali całego projektu

|  |  |  |
| --- | --- | --- |
| Zadanie | Ja | Adaś |
| Punkt 3 |  |  |
| Punkt 4 |  |  |
| Punkt 5 |  |  |
| Punkt 6 |  |  |
| Punkt 7 |  |  |
| Punkt 8 |  |  |
| Punkt 9 |  |  |
| Punkt 10 |  |  |
| Punkt 11 |  |  |
| Średnia procentu zrobionych zadań |  |  |
| Średnia poświęconego czasu |  |  |
| Średnia średnich |  |  |

# Inne informacje

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach