Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: **Abc**

Autorzy: **Abc**

Grupa: I1-210A

Kierunek: informatyka

Rok akademicki: …

Poziom i semestr: I/4

Tryb studiów: stacjonarne/niestacjonarne

*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](#_heading=h.30j0zll) Odnośniki do innych źródeł 4

[3](#_heading=h.1fob9te) Słownik pojęć 5

[4](#_heading=h.3znysh7) Wprowadzenie 6

[4.1](#_heading=h.2et92p0) Cel dokumentacji 6

[4.2](#_heading=h.tyjcwt) Przeznaczenie dokumentacji 6

[4.3](#_heading=h.3dy6vkm) Opis organizacji lub analiza rynku 6

[4.4](#_heading=h.1t3h5sf) Analiza SWOT organizacji 6

[5](#_heading=h.4d34og8) Specyfikacja wymagań 7

[5.1](#_heading=h.2s8eyo1) Charakterystyka ogólna 7

[5.2](#_heading=h.3rdcrjn) Wymagania funkcjonalne 7

[5.3](#_heading=h.lnxbz9) Wymagania niefunkcjonalne 8

[6](#_heading=h.35nkun2) Zarządzanie projektem 9

[6.1](#_heading=h.1ksv4uv) Zasoby ludzkie 9

[6.2](#_heading=h.44sinio) Harmonogram prac 9

[6.3](#_heading=h.2jxsxqh) Etapy/kamienie milowe projektu 9

[7](#_heading=h.z337ya) Zarządzanie ryzykiem 10

[7.1](#_heading=h.3j2qqm3) Lista czynników ryzyka 10

[7.2](#_heading=h.1y810tw) Ocena ryzyka 10

[7.3](#_heading=h.4i7ojhp) Plan reakcji na ryzyko 10

[8](#_heading=h.2xcytpi) Zarządzanie jakością 11

[8.1](#_heading=h.1ci93xb) Scenariusze i przypadki testowe 11

[9](#_heading=h.3whwml4) Projekt techniczny 12

[9.1](#_heading=h.2bn6wsx) Opis architektury systemu 12

[9.2](#_heading=h.qsh70q) Technologie implementacji systemu 12

[9.3](#_heading=h.3as4poj) Diagramy UML 12

[9.4](#_heading=h.1pxezwc) Charakterystyka zastosowanych wzorców projektowych 12

[9.5](#_heading=h.49x2ik5) Projekt bazy danych 12

[9.6](#_heading=h.2p2csry) Projekt interfejsu użytkownika 12

[9.7](#_heading=h.147n2zr) Procedura wdrożenia 13

[10](#_heading=h.3o7alnk) Dokumentacja dla użytkownika 14

[11](#_heading=h.23ckvvd) Podsumowanie 15

[11.1](#_heading=h.ihv636) Szczegółowe nakłady projektowe członków zespołu 15

[12](#_heading=h.32hioqz) Inne informacje 16

# 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.

# Wprowadzenie

## Cel dokumentacji

po co ją robimy i co zawiera (poziom szczegółowości)

## Przeznaczenie dokumentacji

dla kogo ona jest

## Opis organizacji lub analiza rynku

Jedna z dwóch opcji:

1. Jeśli dla konkretnej organizacji: Czym jest organizacja, dla której realizowany będzie system; jak działa (lub będzie działała), kiedy system będzie wdrożony – tutaj nie odwołujemy się do samego systemu, tylko opisujemy samo działanie organizacji i role. W szczególności – jak wyglądają główne procesy biznesowe.
2. Jeśli na masowy rynek: Pobieżna analiza rynku. Dla kogo będzie przydatny taki system. Ile jest organizacji, które będą mogły z niego skorzystać, użytkowników w poszczególnych organizacjach. Czy te organizacje stanowią jednorodną grupę czy są różne rodzaje. Co one mają ze sobą wspólnego. Jak ta liczba będzie się zmieniała w najbliższej przyszłości.

## Analiza SWOT organizacji

Tylko jeśli dla konkretnej organizacji

Wystarczy sama tabela 2x2 (silne-słabe-szanse-zagrożenia)

# 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

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

### 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

# 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

Numer 1

* Przełączanie między dostępnymi językami programistycznymi
* Rodzaj testów: testy dymne
* Tester – Użytkownik
* Termin – nie wcześniej niż po realizacji Zaprojektowanie GUI (Punkt 6.2)
* Narzędzia wspomagające – brak

|  |  |
| --- | --- |
| Przebieg działań: Przełączanie między dostępnymi językami programistycznymi | |
| Użytkownik | System |
|  | System czeka na polecenie od Użytkownika |
| Użytkownik przy pomocy kursora myszy najeżdża na przycisk, który wybiera dostępny język |  |
| Użytkownik zatwierdza wybór poprzez kliknięcie myszą. |  |
|  | System odbiera wydane polecenie |
|  | System podmienia treść w oknie na wybrany język programistyczny |

* Założenia – Użytkownikowi wyświetli się poprawnie treść kodu w wybranym języku
* Środowisko – aplikacja desktopowa
* Warunki wstępne – Użytkownik musi mieć uruchomioną aplikację, wybrane w głównym menu “Opis algorytmu” oraz wybrany algorytm “Drzewo AVL” z listy algorytmów.
* Zestaw danych testowych

|  |  |
| --- | --- |
| Przebieg działań: Przełączanie między dostępnymi językami programistycznymi | |
| Użytkownik | System |
|  | System czeka na polecenie od Użytkownika |
| Użytkownik przy pomocy kursora myszy najeżdża na przycisk z podpisem “Java” |  |
| Użytkownik zatwierdza wybór poprzez kliknięcie myszą. |  |
|  | System odbiera wydane polecenie |
|  | System podmienia treść w oknie na kod napisany w języku Java |

|  |  |
| --- | --- |
| Przebieg działań: Przełączanie między dostępnymi językami programistycznymi (brak dostępnego języka) | |
| Użytkownik | System |
|  | System czeka na polecenie od Użytkownika |
| Użytkownik przy pomocy kursora myszy najeżdża na przycisk z podpisem “Java” |  |
| Użytkownik zatwierdza wybór poprzez kliknięcie myszą. |  |
|  | System odbiera wydane polecenie |
|  | System pozostaje w stanie bezczynności |

Numer 2

* Dodanie nowej wartości do węzła w wizualizacji
* Rodzaj testów: testy dymne
* Tester – Użytkownik
* Termin – nie wcześniej niż po realizacji Zaprojektowanie GUI (Punkt 6.2)
* Narzędzia wspomagające – brak

|  |  |
| --- | --- |
| Przebieg działań: Dodanie nowej wartości do węzła w wizualizacji | |
| Użytkownik | System |
|  | System czeka na polecenie od Użytkownika |
| Użytkownik przy pomocy bocznego menu w polu “Dodaj węzeł” wprowadza z klawiatury wartość. |  |
| Użytkownik zatwierdza wybór poprzez kliknięcie przycisku z symbolem strzałki |  |
|  | System odbiera wydane polecenie |
|  | System dodaje do wizualizacji wybraną wartość |

* Założenia – Użytkownikowi wyświetli się poprawnie wybrana przez niego wartość na wizualizacji
* Środowisko – aplikacja desktopowa
* Warunki wstępne – Użytkownik musi mieć uruchomioną aplikację, wybrane w głównym menu “Wizualizacja” oraz wybrany algorytm “Drzewo Czerwono-Czarne” z listy algorytmów

|  |  |
| --- | --- |
| Przebieg działań: Dodanie nowej wartości do węzła w wizualizacji | |
| Użytkownik | System |
|  | System czeka na polecenie od Użytkownika |
| Użytkownik przy pomocy bocznego menu w polu “Dodaj węzeł” wprowadza z klawiatury wartość=12. |  |
| Użytkownik zatwierdza wybór poprzez kliknięcie przycisku z symbolem strzałki |  |
|  | System odbiera wydane polecenie |
|  | System dodaje do wizualizacji wartość=12 |

|  |  |
| --- | --- |
| Przebieg działań: Dodanie nowej wartości do węzła w wizualizacji (błędna wartość) | |
| Użytkownik | System |
|  | System czeka na polecenie od Użytkownika |
| Użytkownik przy pomocy bocznego menu w polu “Dodaj węzeł” wprowadza z klawiatury wartość=a. |  |
| Użytkownik zatwierdza wybór poprzez kliknięcie przycisku z symbolem strzałki |  |
|  | System odbiera wydane polecenie |
|  | System pozostaje w stanie bezczynności |

|  |  |
| --- | --- |
| Przebieg działań: Dodanie nowej wartości do węzła w wizualizacji (brak danych) | |
| Użytkownik | System |
|  | System czeka na polecenie od Użytkownika |
| Użytkownik przy pomocy bocznego menu w polu “Dodaj węzeł” wprowadza z klawiatury wartość=’ ‘ (pusty znak). |  |
| Użytkownik zatwierdza wybór poprzez kliknięcie przycisku z symbolem strzałki |  |
|  | System odbiera wydane polecenie |
|  | System pozostaje w stanie bezczynności |

Numer 3

* Usunięcie węzła w wizualizacji
* Rodzaj testów: testy dymne
* Tester – Użytkownik
* Termin – nie wcześniej niż po realizacji Zaprojektowanie GUI (Punkt 6.2) i zrealizowaniu przypadku testowego oznaczonego numerem 2 (8.1)
* Narzędzia wspomagające – brak

|  |  |
| --- | --- |
| Przebieg działań: Usunięcie węzła w wizualizacji | |
| Użytkownik | System |
|  | System czeka na polecenie od Użytkownika |
| Użytkownik przy pomocy bocznego menu w polu “Usuń węzeł” wprowadza z klawiatury wartość. |  |
| Użytkownik zatwierdza wybór poprzez kliknięcie przycisku z symbolem strzałki |  |
|  | System odbiera wydane polecenie |
|  | System usuwa z wizualizacji wybraną wartość |

* Założenia – Użytkownikowi wyświetli się poprawnie wizualizacja po wprowadzeniu zmian
* Środowisko – aplikacja desktopowa
* Warunki wstępne – Użytkownik musi mieć uruchomioną aplikację, wybrane w głównym menu “Wizualizacja” oraz wybrany algorytm “Drzewo Czerwono-Czarne” z listy algorytmów.

|  |  |
| --- | --- |
| Przebieg działań: Usunięcie węzła w wizualizacji | |
| Użytkownik | System |
|  | System czeka na polecenie od Użytkownika |
| Użytkownik przy pomocy bocznego menu w polu “Usuń węzeł” wprowadza z klawiatury wartość=12. |  |
| Użytkownik zatwierdza wybór poprzez kliknięcie przycisku z symbolem strzałki |  |
|  | System odbiera wydane polecenie |
|  | System usuwa z wizualizacji wartość=12 |

|  |  |
| --- | --- |
| Przebieg działań: Dodanie nowej wartości do węzła w wizualizacji (błędna wartość) | |
| Użytkownik | System |
|  | System czeka na polecenie od Użytkownika |
| Użytkownik przy pomocy bocznego menu w polu “Usuń węzeł” wprowadza z klawiatury wartość=a. |  |
| Użytkownik zatwierdza wybór poprzez kliknięcie przycisku z symbolem strzałki |  |
|  | System odbiera wydane polecenie |
|  | System pozostaje w stanie bezczynności |

|  |  |
| --- | --- |
| Przebieg działań: Dodanie nowej wartości do węzła w wizualizacji (brak węzłów do usunięcia) | |
| Użytkownik | System |
|  | System czeka na polecenie od Użytkownika |
| Użytkownik przy pomocy bocznego menu w polu “Usuń węzeł” wprowadza z klawiatury wartość=12 |  |
| Użytkownik zatwierdza wybór poprzez kliknięcie przycisku z symbolem strzałki |  |
|  | System odbiera wydane polecenie |
|  | System usuwa z wizualizacji wartość=12 |
| Użytkownik przy pomocy bocznego menu w polu “Usuń węzeł” wprowadza z klawiatury wartość=12 |  |
| Użytkownik zatwierdza wybór poprzez kliknięcie przycisku z symbolem strzałki |  |
|  | System odbiera wydane polecenie |
|  | System pozostaje w stanie bezczynności |

# 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

### 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

okien, stron, aktywności (Android)

### 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

# Inne informacje

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach