Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: Abc

Autorzy: Abc

Grupa: I1-210A

Kierunek: informatyka

Rok akademicki: 2013/2014

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.

Spis treści

[2 Odnośniki do innych źródeł 4](#_Toc383525735)

[3 Wprowadzenie 5](#_Toc383525736)

[3.1 Cel dokumentacji 5](#_Toc383525737)

[3.2 Przeznaczenie dokumentacji 5](#_Toc383525738)

[4 Specyfikacja wymagań 6](#_Toc383525739)

[4.1 Charakterystyka ogólna 6](#_Toc383525740)

[4.2 Historie użytkownika 6](#_Toc383525741)

[4.3 Wymagania funkcjonalne 6](#_Toc383525742)

[4.4 Wymagania niefunkcjonalne 7](#_Toc383525743)

[5 Zarządzanie projektem 8](#_Toc383525744)

[5.1 Zasoby ludzkie 8](#_Toc383525745)

[5.2 Etapy/kamienie milowe projektu 8](#_Toc383525746)

[5.3 Harmonogram prac 8](#_Toc383525747)

[6 Zarządzanie ryzykiem 9](#_Toc383525748)

[6.1 Lista czynników ryzyka 9](#_Toc383525749)

[6.2 Ocena ryzyka 9](#_Toc383525750)

[6.3 Plan reakcji na ryzyko 9](#_Toc383525751)

[7 Zarządzanie jakością 10](#_Toc383525752)

[7.1 Definicje 10](#_Toc383525753)

[7.2 Scenariusze testowe 10](#_Toc383525754)

[7.3 Proces obsługi defektów/awarii 10](#_Toc383525755)

[8 Projekt techniczny 11](#_Toc383525756)

[8.1 Opis architektury systemu 11](#_Toc383525757)

[8.2 Technologie implementacji systemu 11](#_Toc383525758)

[8.3 Diagramy UML 11](#_Toc383525759)

[8.4 Charakterystyka zastosowanych wzorców projektowych 11](#_Toc383525760)

[8.5 Projekt bazy danych 11](#_Toc383525761)

[8.6 Projekt interfejsu użytkownika 12](#_Toc383525762)

[8.7 Procedura wdrożenia 12](#_Toc383525763)

[9 Dokumentacja dla użytkownika 13](#_Toc383525764)

[10 Podsumowanie 14](#_Toc383525765)

[10.1 Wycena prac 14](#_Toc383525766)

[10.2 Szczegółowe nakłady projektowe członków zespołu 14](#_Toc383525767)

[11 Inne informacje 15](#_Toc383525768)

# 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

# Wprowadzenie

## Cel dokumentacji

po co ją robimy i co zawiera (poziom szczegółowości)

## Przeznaczenie dokumentacji

dla kogo ona jest

# Specyfikacja wymagań

## Charakterystyka ogólna

### Definicja produktu

jedno zdanie o systemie

### Cel biznesowy

co chcemy osiągnąć wdrażając system

### Użytkownicy

lista z ew. krótkim wyjaśnieniem

### Środowisko wdrożeniowe

platforma sprzętowa, system operacyjny, inne komponenty niezbędne do współpracy

### Ograniczenia projektowe i wdrożeniowe

przepisy prawne, specyficzne technologie, narzędzia, b.d., protokoły komunikacyjne, aspekty zabezpieczeń, zgodność ze standardami, powiązania z innymi aplikacjami

### Korzyści z systemu

dla poszczególnych grup użytkowników

### Analiza SWOT organizacji

## Historie użytkownika

lista

## Wymagania funkcjonalne

### Lista wymagań

lista numerowana

### Szczegółowy opis wymagań

każde na nowej stronie wg następujących punktów:

* Numer – jako ID
* Nazwa
* Użytkownicy
* Przebieg działań
* Warunki początkowe
* 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
* prawdopodobieństwo defektów - na skali 1-5 lub BN-BW

## 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

## Etapy/kamienie milowe projektu

wg kaskadowego, przyrostowego lub innego cyklu życia

## Harmonogram prac

wraz ze wskazaniem, co jest warunkiem odbioru danego etapu i przejścia do następnego

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.)

# Zarządzanie ryzykiem

## Lista czynników ryzyka

## Ocena ryzyka

prawdopodobieństwo i wpływ

## Plan reakcji na ryzyko

# Zarządzanie jakością

## Definicje

### Priorytety defektów/awarii

np. 1-5, co oznaczają wartości

### Istotność/znaczenie problemu

blocker, major, minor, … z wyjaśnieniem wartości skali

## Scenariusze testowe

szczegółowy plan testowania systemu – głównie testowanie funkcjonalności; każdy scenariusz od nowej strony wg następujących punktów:

* numer – jako ID
* nazwa scenariusza – co test w nim testowane (max kilka wyrazów)
* 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
* 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*

## Proces obsługi defektów/awarii

działania podejmowane w przypadku zgłoszenia defektu – może być w formie tabelarycznej

* kto i co ma robić po kolei, jaki czas reakcji
* być może zależy to od priorytetu/wpływu/lokalizacji defektu – wtedy kilka alternatywnych ścieżek obsługi – np. proces ogólny i kilka specyficznych w zależności od pewnych czynników

# 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!

### Diagram(-y) przypadków użycia

### 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

### Lista głównych elementów interfejsu

okien, stron, aktywności (Android)

### Projekty szczegółowe poszczególnych elementów

każdy element od nowej strony z następującą minimalną zawartością:

* numer – ID elementu
* nazwa – np. formularz danych produktu
* opis – dodatkowe opcjonalne informacje o przeznaczeniu, obsłudze – jeśli nazwa nie będzie wystarczająco czytelna
* projekt graficzny – wystarczy schemat w narzędziu graficznym lub zrzut ekranu – z przykładowymi informacjami (nie pusty!!!)
* 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

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

## Wycena prac

tabela

* na podstawie harmonogramu, szczegółowych przypadków użycia/wymagań, scenariuszy testowych, itp.
* wziąć pod uwagę zasoby potrzebne (liczba osób i ich zróżnicowanie pod względem wynagrodzeń)
* koszty ew. licencji do kupienia, sprzętu – te mogą być wykorzystane również w innych projektach, więc można przyjąć ułamek wartości (tj. wartość:liczba\_projektów uwzględniając wykorzystanie zasobu w innych projektach)

## 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

# Inne informacje

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach