Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: Parkingomatr

Autorzy: Trawiński Rafał, Kowalewski Adrian

Grupa: I1-213A

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.

1. Spis treści
2. Odnośniki do innych źródeł
   * Zarządzania projektem – <https://hub.jazz.net/ccm02/>
   * Wersjonowanie kodu – https://github.com/adrianak/Parking.git
   * System obsługi defektów – <https://hub.jazz.net/ccm02/>
3. Wprowadzenie
   1. Cel dokumentacji

po co ją robimy i co zawiera (poziom szczegółowości)

Dokumentacja zawiera projekt systemu wspomagającego zarządzanie parkingiem. Zawiera on narzędzia umożliwiające sprawniejsze zarządzanie parkingiem oraz kontrolę jego działania

* 1. Przeznaczenie dokumentacji

dla kogo ona jest

Dokumentacja jest przeznaczona dla osób zainteresowanych systemem.

1. Specyfikacja wymagań
   1. Charakterystyka ogólna

Wymagany jest parking,

papier,

szlaban,

komputer,

* + 1. Definicja produktu

jedno zdanie o systemie

System dla parkingów.

* + 1. Cel biznesowy

co chcemy osiągnąć wdrażając system

Zwiększenie zysków z powadzenia parkingu oraz ułatwienie zarządzania nim.

* + 1. Użytkownicy

lista z ew. krótkim wyjaśnieniem

Właściciele parkingów

Klienci parkingów

Zarządcy parkingów

* + 1. Środowisko wdrożeniowe

platforma sprzętowa, system operacyjny, inne komponenty niezbędne do współpracy

Komputer PC

Windows XP

Bramka wjazdowa

Bramka wyjazdowa

Terminal myszą oraz klawiaturą.

* + 1. Ograniczenia projektowe i wdrożeniowe

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

* + 1. Korzyści z systemu

dla poszczególnych grup użytkowników

Dla właścicieli parkingów zwiększenie zysków i łatwiejsza kontrola nad obiektem.

Klienci parkingów zwiększenie bezpieczeństwa z korzystania z parkingu oraz korzyści z korzystania z karty stałego klienta

Zarządcy parkingów korzystanie ze statystyk użytkowania obiektu, ułatwienie pracy.

* + 1. Analiza SWOT organizacji

|  |  |  |
| --- | --- | --- |
|  | Pozytywne | Negatywne |
| Wewnętrzne | -Bezpieczeństwo  -Porządek | -Parkowanie zabiera wiecej czasu |
| Zewnętrzne |  |  |

* 1. Historie użytkownika

lista

* 1. Wymagania funkcjonalne
     1. Lista wymagań

lista numerowana

* + 1. 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
  1. 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
5. Zarządzanie projektem
   1. Zasoby ludzkie

(rzeczywiste lub hipotetyczne) – przy realizacji projektu

* 1. Etapy/kamienie milowe projektu

wg kaskadowego, przyrostowego lub innego cyklu życia

* 1. 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.)

1. Zarządzanie ryzykiem
   1. Lista czynników ryzyka
   2. Ocena ryzyka

prawdopodobieństwo i wpływ

* 1. Plan reakcji na ryzyko

1. Zarządzanie jakością
   1. Definicje
      1. Priorytety defektów/awarii

np. 1-5, co oznaczają wartości

* + 1. Istotność/znaczenie problemu

blocker, major, minor, … z wyjaśnieniem wartości skali

* 1. 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*
  1. 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

1. Projekt techniczny
   1. Opis architektury systemu

z ew. rysunkami pomocniczymi

* 1. Technologie implementacji systemu

tabela z listą wykorzystanych technologii, każda z uzasadnieniem

* 1. Diagramy UML

każdy diagram ma mieć tytuł!

* + 1. Diagram(-y) przypadków użycia
    2. Diagram(-y) klas
    3. Diagram(-y) czynności
    4. Diagramy sekwencji

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

* + 1. Inne diagramy

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

* 1. Charakterystyka zastosowanych wzorców projektowych

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

* 1. Projekt bazy danych
     1. Schemat

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

* + 1. Projekty szczegółowe tabel
  1. Projekt interfejsu użytkownika
     1. Lista głównych elementów interfejsu

okien, stron, aktywności (Android)

* + 1. 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.
  1. Procedura wdrożenia

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

1. 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.

1. Podsumowanie
   1. 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)
  1. 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

1. Inne informacje

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach