Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: Hotel

Autorzy: Beniamin Bochnacki, Michał Bortnowski

Grupa: I1-210C

Kierunek: informatyka

Rok akademicki: 2015/2016

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](#_Toc413831237)

[3 Słownik pojęć 5](#_Toc413831238)

[4 Wprowadzenie 6](#_Toc413831239)

[4.1 Cel dokumentacji 6](#_Toc413831240)

[4.2 Przeznaczenie dokumentacji 6](#_Toc413831241)

[4.3 Opis organizacji 6](#_Toc413831242)

[4.4 Analiza SWOT organizacji 8](#_Toc413831243)

[5 Specyfikacja wymagań 9](#_Toc413831244)

[5.1 Charakterystyka ogólna 9](#_Toc413831245)

[5.2 Wymagania funkcjonalne 10](#_Toc413831246)

[5.3 Wymagania niefunkcjonalne 11](#_Toc413831247)

[6 Zarządzanie projektem 13](#_Toc413831248)

[6.1 Zasoby ludzkie 13](#_Toc413831249)

[6.2 Etapy/kamienie milowe projektu 14](#_Toc413831250)

[6.3 Harmonogram prac 14](#_Toc413831251)

[7 Zarządzanie ryzykiem 15](#_Toc413831252)

[7.1 Lista czynników ryzyka 15](#_Toc413831253)

[7.2 Ocena ryzyka 15](#_Toc413831254)

[7.3 Plan reakcji na ryzyko 15](#_Toc413831255)

[8 Zarządzanie jakością 16](#_Toc413831256)

[8.1 Definicje 16](#_Toc413831257)

[8.2 Scenariusze testowe 16](#_Toc413831258)

[8.3 Proces obsługi defektów/awarii 16](#_Toc413831259)

[9 Projekt techniczny 17](#_Toc413831260)

[9.1 Opis architektury systemu 17](#_Toc413831261)

[9.2 Technologie implementacji systemu 17](#_Toc413831262)

[9.3 Diagramy UML 17](#_Toc413831263)

[9.4 Charakterystyka zastosowanych wzorców projektowych 17](#_Toc413831264)

[9.5 Projekt bazy danych 17](#_Toc413831265)

[9.6 Projekt interfejsu użytkownika 17](#_Toc413831266)

[9.7 Procedura wdrożenia 18](#_Toc413831267)

[10 Dokumentacja dla użytkownika 19](#_Toc413831268)

[11 Podsumowanie 20](#_Toc413831269)

[11.1 Szczegółowe nakłady projektowe członków zespołu 20](#_Toc413831270)

[12 Inne informacje 21](#_Toc413831271)

# Odnośniki do innych źródeł

* + Zarządzania projektem – JazzHub
  + Wersjonowanie kodu –Github
  + System obsługi defektów – 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

Niniejsza dokumentacja ma na celu ułatwienie stworzenia systemu, który ułatwi i usprawni pracę Hotelu poprzez dynamiczne zarządzanie obłożeniem pokoi hotelowych.

## Przeznaczenie dokumentacji

Niniejsza dokumentacja przeznaczona jest przede wszystkim dla zespołu programistów, tworzących aplikację, a także i dla użytkowników systemu. Zapewni to jednej i drugiej grupie przejrzysty i sprecyzowany wgląd w fundamentalne funkcje tworzonej aplikacji, zarówno przed jak i po jej debiucie. Dokument jest również w pełni stworzony dla innych zespołów biorących udział w procesie produkcji (testerzy, graficy) oraz osób spoza kręgu deweloperskiego (użytkownicy, niezwiązani z dziedziną informatyki).

## Opis organizacji

Organizacja, dla której realizowany jest system to hotel. Hotelarstwo to forma działalności gospodarczej charakteryzująca się:

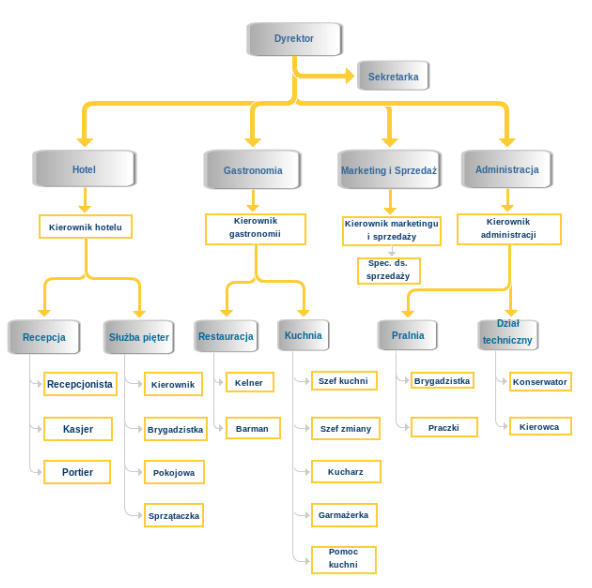
* Szczególnym rodzajem gościnności – gościnność za odpłatność,
* Różnego rodzaju działalnością usługową, główne z nich to usługi bytowe takie jak zapewnienie noclegu i wyżywienia,
* Określonym w czasie i z reguły krótkotrwałym pobytem gości
* Zakresem usług świadczonych przez hotelarzy rozszerzanym o innego rodzaju działalność turystyczną i rozrywkową

Sama organizacja zależna jest typu tego zakładu. Zupełnie inaczej wygląda struktura organizacyjna małego pensjonatu, prowadzonego przez kilka osób, a inaczej dużego luksusowego hotelu, nastawionego na klientów biznesowych. Naszym odbiorcą docelowym jest hotel „Dobosz”, mieszczący się przy ul. Jana Pawła II 21/40 w Szczecinie, który jest hotelem luksusowym, posiadającym cztery gwiazdki i obsługujący średnio 1200 klientów w ciągu roku.

Głównym zwierzchnikiem organizacji hotelu „Dobosz” jest dyrektor, któremu bezpośrednio podlegają Sekretarka i sztab kierowników odpowiedzialnych za administrację, hotel, gastronomię i marketing. Każdemu kierownikowi podlegają z kolei pracownicy poszczególnych działów.

Obowiązkami recepcji, pracę której usprawni system, będą następujące procesy biznesowe:

* Przyjmowanie i realizacja zleceń na rezerwację miejsc noclegowych,
* Przyjmowanie gości,
* Prowadzenie ewidencji gości, zgodnie z obowiązującymi zasadami,
* Organizowanie pomocy przy wyjazdach i przyjazdach,
* Sporządzanie codziennych grafików wykorzystania pokoi,
* Wydawanie kluczy do pokoi,
* Sprawne działanie w przypadkach losowych,
* Przyjmowanie reklamacji gości.



*Organizacja hotelu „Dobosz”*

## Analiza SWOT organizacji

|  |  |
| --- | --- |
| Silne strony:   * Przejrzysty model procesów biznesowych hotelu, * Wysoka jakość świadczonych usług hotelowych, * Konkurencyjny koszt obsługi klienta detalicznego, * Jednostka działająca z wieloma scenariuszami i organizacjami | Słabe strony:   * Brak dostatecznie wykwalifikowanej kadry pracowniczej w dziedzinie IT, * Brak rozbudowanej struktury sieciowej hotelu, * Powolny czas obsługi, |
| Szanse:   * Możliwość rozwoju, * Możliwość obsługi lokalnych przedsiębiorców, * Możliwość zwiększenia szybkości obsługi klientów, | Zagrożenia:   * Konieczność szybkiego reagowania na sytuacje losowe, * Uzależnienie jakości świadczenia usług od czynnika ludzkiego (margines błędu pracownika), * Dalszy odpływ klientów |

# Specyfikacja wymagań

## Charakterystyka ogólna

### Definicja produktu

System zarządzania rezerwacjami hotelowymi – typu BMP.

### Cel biznesowy

Projektowany system ma ułatwić pracę i zwiększyć produktywność pracowników recepcji. Recepcjoniści powinni być wspierani przez system podczas obsługi gościa w najwyższym możliwym stopniu. System zarządzania procesami biznesowymi opiera się głównie na zwiększeniu efektywności procesów biznesowych w Hotelu Dobosz.

### Użytkownicy

**Recepcjonista** – podstawowy użytkownik systemu, recepcjonista będzie korzystał z większości funkcjonalności systemu.

**Pracownik** służby pięter – drugi w kolejności użytkownik systemu. Jego działalność w systemie leży tylko w zakresie zarządzania

**Kierownik recepcji** – użytkownik mający większe prawa niż recepcjonista. Może mieć wgląd w takie rzeczy jak np. historia rachunku, karty kredytowe.

**Administrator** – użytkownik z pełnią praw w systemie.

**Klient** – dokonuje rezerwacji i płatności w systemie.

**Bramka** **płatności** – Powiadamia system o dokonanej płatności przez klienta.

**System** **fakturowania** – wystawia faktury za pobyty.

### Korzyści z systemu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Korzyść | Klient | Recepcjonista | Kierownik | Pr.Sł.Pięter |
| 1. Dokonywanie rezerwacji | X | X | X |  |
| 1. Odwoływanie rezerwacji | X | X | X |  |
| 1. Edycja rezerwacji |  | X | X |  |
| 1. Przegląd rezerwacji |  | X | X |  |
| 1. Potwierdzenie rezerwacji | X | X | X |  |
| 1. Przegląd dostępnych pokoi | X |  |  |  |
| 1. Obliczanie kosztów rezerwacji 2. Zmiana statusu pokoju |  | X | X | X |
| 1. Anulowanie rezerwacji |  | X | X |  |
| 1. Sprawdzenie stanu płatności |  |  | X |  |
| 1. Dokonanie płatności | X |  |  |  |
| 1. Przegląd klientów |  | X | X |  |
| 1. Wystawienie faktury |  |  | X |  |
|  |  |  |  |  |

**//Klient**: 1,2,5,11,6

**//Recepcjonista**: 5,1,2,3,4,7,12

**//Kierownik**: 5,1,2,3,4,7,12, 13,10

**//Pracownik** **służby** **pięter**: 8

### Ograniczenia projektowe i wdrożeniowe

System powinien być ograniczony niżej wymienionymi parametrami:

* Baza danych powinna mieścić się w granicach rozmiaru dysku sieciowego o pojemności 300GB, (ponieważ takim limitem hostingu dysponuje klient)
* System kompatybilny z Windows XP, (życzenie klienta)
* Przechowywane powinny być dane klientów do 5 lat od ostatniej rezerwacji. (życzenie klienta)
* //wkleić przypisy prawne
* Baza danych powinna mieścić się na serwerze zewnętrznym i regularnie (T=10 dni) tworzyć backup danych, aby zapobiec utracie danych.
* //internetowa aplikacja czy stacjonarna?
* System powinien działać płynnie na komputerze o specyfikacje równej lub lepszej niż: Pentium IV 1GHz, 512 MB RAM (życzenie klienta)
* Czas odpowiedzi aplikacji powinien być krótszy niż 2 sekundy, (czasy dłuższe skutecznie wpłyną na niekorzyść naszego systemu, a tym samym zadowolenie klientów)
* Rezerwacja powinna być widoczna w systemie w czasie mniejszym niż 1 minuta, (nie możemy sobie pozwolić na dwukrotną rezerwację tego samego miejsca, sytuacje tego typu doprowadziłyby do katastrofy w systemie rezerwacji)

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!

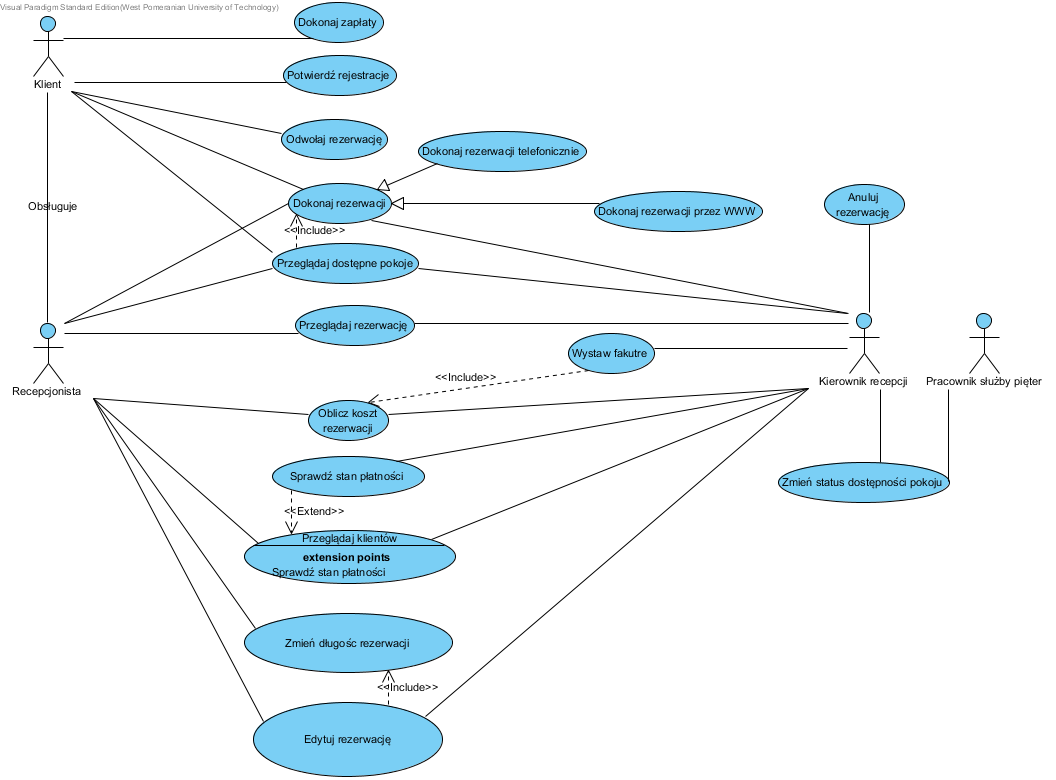
## Wymagania funkcjonalne

### Lista wymagań

lista numerowana – czyli lista przypadków użycia

### Diagramy przypadków użycia

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



### Szczegółowy opis wymagań

ID: 1

Nazwa: „Dokonywanie rezerwacji”

Głównym celem elementu jest wypełnienie korzyści systemu o ID „1”.

Użytkownikami korzystającymi z tej funkcji systemu będą klient, recepcjonista i kierownik recepcji.

Warunki początkowe: użytkownik zalogowany do systemu

Przebieg działań:

1. Klient/ recepcjonista/ kierownik recepcji wybiera numer interesującego go pokoju,
2. Klient / recepcjonista / kierownik recepcji uzupełnia formularz z danymi rezerwacji z polami: imię, nazwisko, adres email, telefon kontaktowy, data początku i końca pobytu
3. System sprawdza poprawność danych
4. Klient zostaje dodany do bazy danych z unikalnym identyfikatorem ID
5. Następuje generowanie faktury, która będzie opłacona na koniec pobytu lub poprzez serwis typu przelewy24.pl w terminie wcześniejszym
6. Dany pokój zyskuje status „zajęty” w danym przedziale czasowym

Alternatywny przebieg działań:

1. Klient/ recepcjonista/ kierownik recepcji wybiera numer interesującego go pokoju,
2. Klient / recepcjonista / kierownik recepcji uzupełnia formularz z danymi rezerwacji z polami: imię, nazwisko, adres email, telefon kontaktowy, data początku i końca pobytu
3. System sprawdza poprawność danych
4. W przypadku podania złych danych system wyświetla informacje „Podano nieprawidłowe dane!”

Efekty:

* Następuje generowanie faktury
* Dany pokój zyskuje status „zajęty” w danym przedziale czasowym

Wymagania niefunkcjonalne:

Brak

Częstotliwość: 5

Istotność: 5

Prawdopodobieństwo defektów: 1

ID: 2

Nazwa: „Edycja rezerwacji”

Głównym celem elementu jest wypełnienie korzyści systemu o ID „2” oraz „3”.

Użytkownikami korzystającymi z tej funkcji systemu będą klient, recepcjonista i kierownik recepcji.

Warunki początkowe: użytkownik zalogowany do systemu, rezerwacja widoczna w systemie i poprawnie skonfigurowana wcześniej

Przebieg działań:

1. Klient wysyła formularz z edytowaną rezerwacją
2. System sprawdza poprawność danych
3. recepcjonista/ kierownik recepcji wprowadza nowe dane rezerwacji
4. System sprawdza poprawność danych i możliwość edycji rezerwacji
5. Do klienta zostaje wysłany email z linkiem do potwierdzenia dokonanych zmian
6. Następuje edycja zamówienia lub usunięcie rezerwacji
7. Edytowana rezerwacja widnieje pod tym samym numerem ID lub jest usunięta

Alternatywny przebieg działań 1:

1. Klient wysyła formularz z edytowaną rezerwacją
2. System sprawdza poprawność danych
3. Wysyłany jest komunikat o błędnie wprowadzonych danych

Alternatywny przebieg działań 2:

* Klient wysyła formularz z edytowaną rezerwacją
* System sprawdza poprawność danych
* recepcjonista/ kierownik recepcji wprowadza nowe dane rezerwacji
* System sprawdza poprawność danych i możliwość edycji rezerwacji
* W przypadku niepowodzenia system odsyła klientowi informacje o niemożliwości wprowadzenia zmian

Efekty:

1. Edycja rezerwacji
2. Lub usunięcie rezerwacji

Wymagania niefunkcjonalne:

Brak

Częstotliwość: 4

Istotność: 5

Prawdopodobieństwo defektów: 2

ID: 3

Nazwa: „Przegląd rezerwacji”

Głównym celem elementu jest wypełnienie korzyści systemu o ID „4”.

Użytkownikami korzystającymi z tej funkcji systemu będą recepcjonista i kierownik recepcji.

Warunki początkowe: użytkownik zalogowany do systemu, rezerwacja widoczna w systemie i poprawnie skonfigurowana wcześniej

Przebieg działań:

1. Recepcjonista lub kierownik recepcji wybiera daną rezerwację
2. System sprawdza ID rezerwacji
3. Rezerwacja jest wyświetlona wraz ze wszelkimi informacjami o niej

Alternatywny przebieg działań 1:

1. Recepcjonista lub kierownik recepcji wybiera daną rezerwację
2. System sprawdza ID rezerwacji
3. System wyświetla błąd o braku rezerwacji w bazie danych

Efekty:

* Wyświetlenie danych rezerwacji o danym numerze ID

Wymagania niefunkcjonalne:

Brak

Częstotliwość: 4

Istotność: 4

Prawdopodobieństwo defektów: 1

ID: 4

Nazwa: „Potwierdzenie rezerwacji”

Głównym celem elementu jest wypełnienie korzyści systemu o ID „5”.

Użytkownikami korzystającymi z tej funkcji systemu będzie grupa klientów, recepcjoniści i kierownicy recepcji.

Warunki początkowe: użytkownik wprowadził poprawne dane rezerwacji,

Przebieg działań:

1. System sprawdza dane wprowadzone przez klienta pod kątem poprawności informacji i dostępności
2. Recepcjonista/Kierownik recepcji dostaje komunikat o poprawności danych
3. Recepcjonista/kierownik recepcji potwierdza dane rezerwacji
4. Odnotowanie rezerwacji w bazie danych
5. Generowany jest mail z informacją o prawidłowej rezerwacji
6. Mail jest wysyłany do klienta

Alternatywny przebieg działań 1:

1. System sprawdza dane wprowadzone przez klienta pod kątem poprawności informacji i dostępności
2. System generuje komunikat o niepoprawnych danych
3. Wszelkie dane zamówienia nie zostają zapisane do bazy danych

Alternatywny przebieg działań 2:

1. System sprawdza dane wprowadzone przez klienta pod kątem poprawności informacji i dostępności
2. System wyświetla komunikat, ze rezerwacja już była potwierdzona

Efekty:

* W bazie danych zmienia się status rezerwacji
* Wygenerowanie wiadomości email

Wymagania niefunkcjonalne:

Brak

Częstotliwość: 5

Istotność: 5

Prawdopodobieństwo defektów: 4

ID: 5

Nazwa: „Przegląd dostępnych pokoi”

Głównym celem elementu jest wypełnienie korzyści systemu o ID „6”.

Użytkownikami korzystającymi z tej funkcji systemu będzie grupa klientów, recepcjoniści i kierownicy recepcji.

Warunki początkowe: użytkownik z dostępem do Internetu

Przebieg działań:

1. Recepcjonista/kierownik recepcji lub klient wybiera opcję przeglądania wolnych pokoi
2. Wyświetla się plan budynku z wolnymi pokojami zaznaczonymi na zielono, a zajętymi na czerwono (informacja ta jest generowana na podstawie dostępności na obecny dzień – po najechaniu na dany „czerwony” pokój myszką wyświetlają się daty z dostępnością pokoju)

Efekty:

* Wyświetlenie danych z dostępnościami pokojów

Wymagania niefunkcjonalne:

Brak

Częstotliwość: 5

Istotność: 5

Prawdopodobieństwo defektów: 1

ID: 6

Nazwa: „Obliczanie kosztów rezerwacji”

Głównym celem elementu jest wypełnienie korzyści systemu o ID „7”.

Użytkownikami korzystającymi z tej funkcji systemu będą, recepcjoniści i kierownicy recepcji.

Warunki początkowe: użytkownik wprowadził poprawne dane rezerwacji, rezerwacja jest widoczna w bazie danych

Przebieg działań:

1. System oblicza ilość dób, które spędzie gość w hotelu
2. Ilość dób jest mnożona przez odpowiedni przelicznik cenowy
3. Wartości są sumowane
4. Obliczony jest koszt rezerwacji

Alternatywny przebieg działań 1:

* Rezerwacja nie zostaje znaleziona, wyświetlony zostaje błąd.

Efekty:

* Obliczona jest wartość rezerwacji

Wymagania niefunkcjonalne:

Brak

Częstotliwość: 5

Istotność: 5

Prawdopodobieństwo defektów: 4

ID: 7

Nazwa: „Zmiana statusu pokoju”

Głównym celem elementu jest wypełnienie korzyści systemu o ID „8”.

Użytkownikami korzystającymi z tej funkcji systemu będą kierownicy recepcji i pracownicy służby pięter.

Warunki początkowe: użytkownik jest zalogowany do systemu

Przebieg działań:

1. Kierownik recepcji lub pracownik służby pięter wybiera pokój, któremu chce zmienić status z planszy wyboru pokojów
2. System sprawdza obecny status pokoju i wyświetla kolor zielony gdy pokój jest wolny, czerwony gdy jest zajęty
3. System podaje możliwą opcję zmiany statusu
4. Status jest zmieniony

Efekty:

* Status pokoju jest zmieniony

Wymagania niefunkcjonalne:

?

Częstotliwość: 4

Istotność: 5

Prawdopodobieństwo defektów: 1

ID: 8

Nazwa: „Sprawdzenie stanu płatności”

Głównym celem elementu jest wypełnienie korzyści systemu o ID „10”.

Użytkownikami korzystającymi z tej funkcji systemu będą kierownicy recepcji.

Warunki początkowe: rezerwacja przeprowadzona poprawnie, koszt rezerwacji obliczony, faktura wysłana klientowi

Przebieg działań:

1. Kierownik recepcji wybiera daną rezerwację lub ID/imię i nazwisko klienta
2. System weryfikuje poprawność danych wejściowych
3. Wyświetlana jest informacja o stanie płatności danej rezerwacji

Alternatywny przebieg działań 1:

1. Kierownik recepcji wybiera daną rezerwację lub ID/imię i nazwisko klienta
2. System weryfikuje poprawność danych wejściowych
3. Wyświetlony jest błąd dot. błędnie wprowadzonych danych

Alternatywny przebieg działań 2:

1. Kierownik recepcji wybiera daną rezerwację lub ID/imię i nazwisko klienta
2. System weryfikuje poprawność danych wejściowych
3. Wyświetlone jest powiadomienie o braku takiego klienta/rezerwacji

Efekty:

* Wyświetlony jest status płatności za daną rezerwację

Wymagania niefunkcjonalne:

Brak

Częstotliwość: 5

Istotność: 5

Prawdopodobieństwo defektów: 4

ID: 9

Nazwa: „Dokonanie płatności”

Głównym celem elementu jest wypełnienie korzyści systemu o ID „11”.

Użytkownikami korzystającymi z tej funkcji systemu będą klienci.

Warunki początkowe: rezerwacja przeprowadzona poprawnie, koszt rezerwacji obliczony, faktura wysłana klientowi

Przebieg działań:

1. System sprawdza dane wejściowe z warunków początkowych
2. Użytkownik przekierowany jest z linku z wiadomości email z potwierdzeniem rezerwacji do serwisu przelewy24.pl lub payu.pl z podanymi gotowymi danymi przelewu
3. System weryfikuje przelew
4. Po otrzymaniu informacji z payu/przelewy 24 zmieniany jest status płatności zamówienia
5. Wyświetlana jest informacja o stanie płatności danej rezerwacji

Alternatywny przebieg działań 1:

1. System sprawdza dane wejściowe z warunków początkowych
2. Wyświetlany jest komunikat o braku możliwości przeprowadzenia transakcji

Alternatywny przebieg działań 2:

1. System sprawdza dane wejściowe z warunków początkowych
2. Użytkownik przekierowany jest z linku z wiadomości email z potwierdzeniem rezerwacji do serwisu przelewy24.pl lub payu.pl z podanymi gotowymi danymi przelewu
3. System weryfikuje przelew
4. System wyświetla informację o braku otrzymania informacji z banku

Alternatywny przebieg działań 3:

1. System sprawdza dane wejściowe z warunków początkowych
2. Użytkownik przekierowany jest z linku z wiadomości email z potwierdzeniem rezerwacji do serwisu przelewy24.pl lub payu.pl z podanymi gotowymi danymi przelewu
3. System weryfikuje przelew
4. System wyświetla informację o błędnym przelewie

Efekty:

* Wyświetlony jest status płatności za daną rezerwację

Wymagania niefunkcjonalne:

Na koncie bankowym klienta muszą znajdować się środki wyższe lub równe kwocie do zapłaty, niezbędny jest terminal do płatności kartą w przypadku takiej formy zapłaty.

Częstotliwość: 5

Istotność: 5

Prawdopodobieństwo defektów: 5

ID: 10

Nazwa: „Przegląd klientów”

Głównym celem elementu jest wypełnienie korzyści systemu o ID „12”.

Użytkownikami korzystającymi z tej funkcji systemu będą kierownicy recepcji i recepcjoniści.

Warunki początkowe: użytkownik z dostępem do Internetu, klient dokonywał wcześniej jakiejś rezerwacji w hotelu.

Przebieg działań:

1. Kierownik recepcji lub recepcjonista wybierają opcję wyświetlania klientów
2. Wyświetla się opcja podania Imienia I Nazwiska lub ID klienta
3. Dane są wyświetlone

Alternatywny przebieg działań 1:

1. Kierownik recepcji lub recepcjonista wybierają opcję wyświetlania klientów
2. Wyświetla się opcja podania Imienia I Nazwiska lub ID klienta
3. Wyświetlona jest informacja o braku takiego klienta

Efekty:

* Wyświetlenie klienta i szczegółowych informacji o nim

Wymagania niefunkcjonalne:

Brak

Częstotliwość: 3

Istotność: 5

Prawdopodobieństwo defektów: 2

ID: 11

Nazwa: „Wystawienie faktury”

Głównym celem elementu jest wypełnienie korzyści systemu o ID „13”.

Użytkownikami korzystającymi z tej funkcji systemu będą kierownicy recepcji.

Warunki początkowe: rezerwacja przeprowadzona poprawnie, koszt rezerwacji obliczony,

Przebieg działań:

1. Kierownik wybiera opcję wystawienia faktury
2. Kierownik wybiera rezerwację
3. Zostają sprawdzone warunki wejściowe
4. Faktura zostaje wystawiona
5. Faktura zostaje wysłana na adres email klienta

Alternatywny przebieg działań 1:

1. Kierownik wybiera opcję wystawienia faktury
2. Kierownik wybiera rezerwację
3. Funkcja zwraca błąd o braku występowania takiej rezerwacji

Alternatywny przebieg działań 2:

1. Kierownik wybiera opcję wystawienia faktury
2. Kierownik wybiera rezerwację
3. Zostają sprawdzone warunki wejściowe
4. Informacja o braku spełnienia warunków wejściowych

Efekty:

* Wystawienie faktury

Wymagania niefunkcjonalne:

Potrzebna jest drukarka w przypadku konieczności zażądania przez klienta papierowej wersji faktury.

Częstotliwość: 5

Istotność: 5

Prawdopodobieństwo defektów: 4

## Wymagania niefunkcjonalne

* Działanie systemu na systemach operacyjnych: Windows XP lub nowszy.
* Czas szkolenia pracowników – maksymalnie 3 dni.
* Projekt powinien zostać wykonany w języku obiektowym C++, natomiast aplikacja powinna używać protokołów szyfrujących typu TSL, (bezpieczeństwo danych jest w tym wypadku priorytetem)
* Dostępność instrukcji do systemu dla poszczególnych stanowisk w języku polskim.
* Wdrażanie systemu powinno trwać co najwyżej 30 dni.
* Dostęp do strony internetowej z poziomu popularnych przeglądarek internetowych (Chrome, IE, Mozilla Firefox).
* Intuicyjny interfejs (przejrzyste okna, mniej niż 3 kolory przewodnie, mniej niż 20 linii danych na ekranie).
* Jak najmniejsze opóźnienie pomiędzy wprowadzeniem zmiany i wprowadzeniem jej do bazy danych (T<0,5s).
* W razie awarii ponowne uruchomienie systemu w ciągu godziny
* Dodawanie lub usuwanie klientów, produktów itp. w czasie do dwóch minut.
* Zabezpieczenie przechowywanych danych o klientach (szyfrowanie typu TSL),
* Zabezpieczenie danych (haseł, identyfikatorów) dotyczących kont użytkowników, poprzez przechowywanie danych na szyfrowanym serwerze.
* Szybka reakcja na działanie użytkownika (T<0,1s).
* Automatyczne, cotygodniowe wykonywanie backup’u danych.
* Niezawodność- nie więcej niż 1 błąd krytyczny rocznie i aktualizacje systemu eliminujące powstawanie błędu w ciągu 2 dni roboczych od momentu jego zgłoszenia.
* Prawdopodobieństwo zniszczenia danych w awarii poniżej 0,001% wykonanych operacji.
* Łatwość implementacji system – czas trwania implemntacji nie dłuższy niż 3 tygodnie.
* Zgodność systemu z ustawą o ochronie danych osobowych z dnia 29. Sierpnia 1997 r.
* Zabezpieczenie systemu przed atakami hakerów, wirusami (zainstalowanie profesjonalnego oprogramowania antywirusowego firmy Norton i zabezpieczenie serwera przed fizycznym włamaniem).
* System pracuje z zaimplementowanym systemem płatności.
* Możliwość realizacji jednej transakcji na sekundę.

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

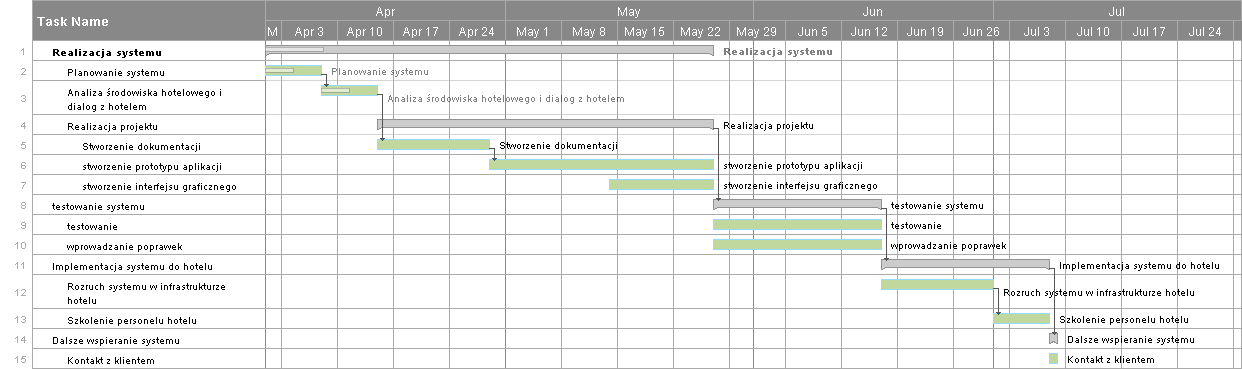
* Kierownik wykonawczy (szef wszystkich działów, na bieżąco egzekwujący efekty pracy danych działów i pracowników, zatwierdza kolejne kamienie milowe i wyznacza oczekiwaną jakość projektu jak i jego podzespołów)
* Szef działu programistów (głównodowodzący zespołem programistów, projektujący konkretne mechanizmy systemu i wyznaczający kolejne cele do realizacji, przypisuje je konkretnym grupą programistów, egzekwuje zlecone zadania i przedstawia je Kierownikowi wykonawczemu) –
* Szef działu grafików (głównodowodzący zespołem grafików, wyznaczający wizję wizualną projektu, a także kolejne projekty graficzne do realizacji, przypisuje je konkretnym grupą grafików, egzekwuje zlecone zadania i przedstawia je Kierownikowi wykonawczemu)
* Szef działu testerów (głównodowodzący zespołem testerów, wyznaczający jakość konkretnych mechanizmów systemu, przypisujący je konkretnym grupom testerów, egzekwuje zlecone zadania i przedstawia je Kierownikowi wykonawczemu)
* Programiści (ludzie bezpośrednio odpowiedzialni za programowanie konkretnych mechanizmów logicznych systemu, zleconych przez szefa działu, podzieleni na podgrupy wykonawcze) – zespół składający się z 15 osób
* Graficy (ludzie bezpośrednio odpowiedzialni za wykonywanie szaty graficznej systemu - ikon, elementów interfejsu itp, zleconych przez szefa działu, podzieleni na podgrupy wykonawcze) – zespół składający się z 7 osób
* Testerzy (grupy badające działanie, przebieg i wykonanie konkretnych mechanizmów systemu) – zespół składający się z 5 osób
* Konsultant/wdrożeniowiec IT (osoba odpowiedzialna za wdrożenie systemu i szkolenia z zakresu obsługi systemu). – zespół składający się z 3 osób

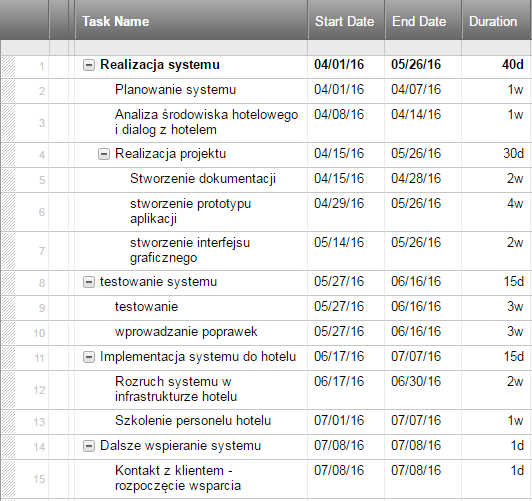
(rzeczywiste lub hipotetyczne) – przy realizacji projektu

## Etapy/kamienie milowe projektu

wg kaskadowego, przyrostowego lub innego cyklu życia

## Harmonogram prac





[*https://app.smartsheet.com/b/home?lx=0tMMLU0BHZZAR4LBYQvUVA*](https://app.smartsheet.com/b/home?lx=0tMMLU0BHZZAR4LBYQvUVA)

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 wersj

e (1.0, 1.5, itd.)

# Zarządzanie ryzykiem

## Lista czynników ryzyka

* Nowa technologia, (Intrinsic)
* Narzucony termin ukończenia (masymalnie rok – imposed),
* Narzucone kryteria niezawodności i bezpieczeństwa (imposed),
* Bardzo długi harmonogram,
* Założenia początkowe,
* Dopuszczenie problemów nieudokumentowanych lub nieprzypisanych.

## Ocena ryzyka

|  |  |  |  |
| --- | --- | --- | --- |
| Czynnik/zagrożenie | Kategoria | Prawdopodobieństwo | Wpływ |
| Nowa technologia | Klient | 20% | Marginalne |
| Narzucony termin ukończenia | Produkt | 40% | Marginalne |
| Narzucone kryteria niezawodności i bezpieczeństwa | Zespół | 50% | Krytyczne |
| Bardzo długi harmonogram | Proces | 20% | Marginalne |
| Założenia początkowe | Produkt | 40% | Krytyczne |
| Dopuszczenie problemów nieudokumentowanych lub nieprzypisanych. | Produkt | 70% | Krytyczne |

## Plan reakcji na ryzyko

- Problem:

* Nowa technologia, z którą mogą mieć problemy pracownicy hotelu

- Rozwiązanie:

* Prowadzenie szkoleń przez pracowników naszej firmy stworzonych dla przyszłych użytkowników systemu z zakresu jego obsługi

- Problem:

* Narzucony termin ukończenia projektu może sprzyjać niepotrzebnemu stresowi w zespole, a ponadto powodować przesunięcia czasowe w harmonogramie

- Rozwiązanie:

* Dokumentacja projektu musi być stworzona bardzo dokładnie już na etapie planowania systemu, a harmonogram prac ściśle konsultowany z zespołem wykonawczym

- Problem:

* Narzucone kryteria niezawodności i bezpieczeństwa

- Rozwiązanie:

* Tworzony system na każdym etapie jego kreacji, powinien być sprawdzany pod kątem zabezpieczeń i niezawodności, a wszelkie błędy natychmiastowo powinny być usuwane

- Problem:

* Bardzo długi harmonogram może być przyczyną niedomówień zespołowych, a w konsekwencji błędów w systemie. Ponadto, kontrola nad tak rozległym w czasie projektem może być ciężka.

- Rozwiązanie:

* Harmonogram w momencie jego tworzenia powinien być precyzyjnie omawiany z zespołem

- Problem:

* Założenia początkowe mogą nie być identyczne z finalnym projektem podczas tak długiej realizacji.

- Rozwiązanie:

* Zespół na bieżąco powinien konsultować wszelkie zmiany z klientem, a także być w granicach rozsądku plastyczny jeżeli chodzi o wprowadzanie zmian względem wcześniej ustalonych założeń początkowych.

- Problem:

* Dopuszczenie przypadków nieopisanych lub nieprzewidzianych może powodować poważne błędy w systemie.

- Rozwiązanie:

* System już we wstępnych etapach jego realizacji, powinien być precyzyjnie zaplanowany po konsultacji z klientem. Wszelkie wyjątkowe przypadki na przestrzeni lat powinny być wypunktowane zespołowi przez klienta i uwzględnione w projekcie jeżeli wynoszą one 5% wszystkich przypadków.

# 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

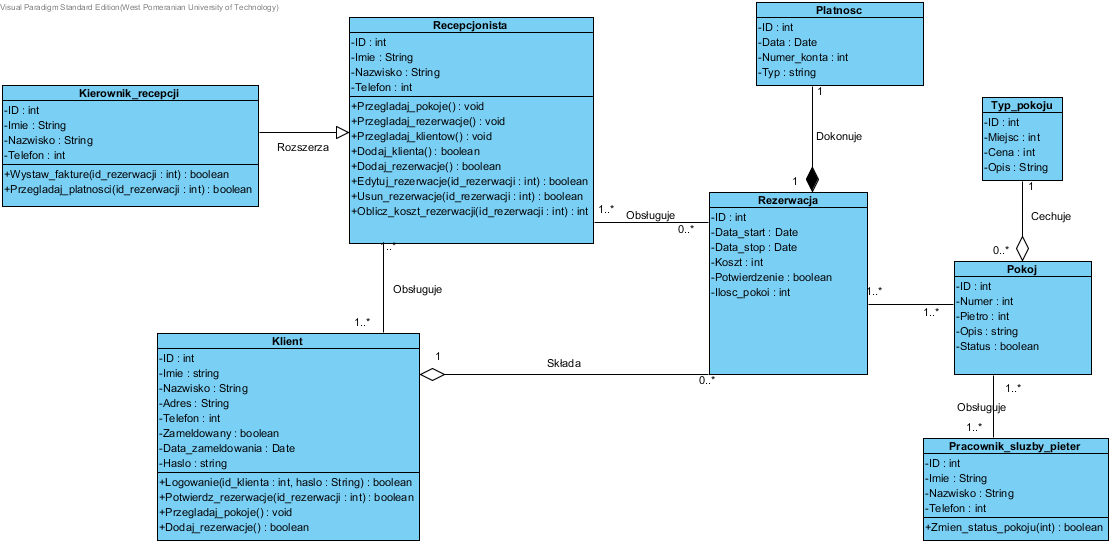
z ew. rysunkami pomocniczymi

## Technologie implementacji systemu

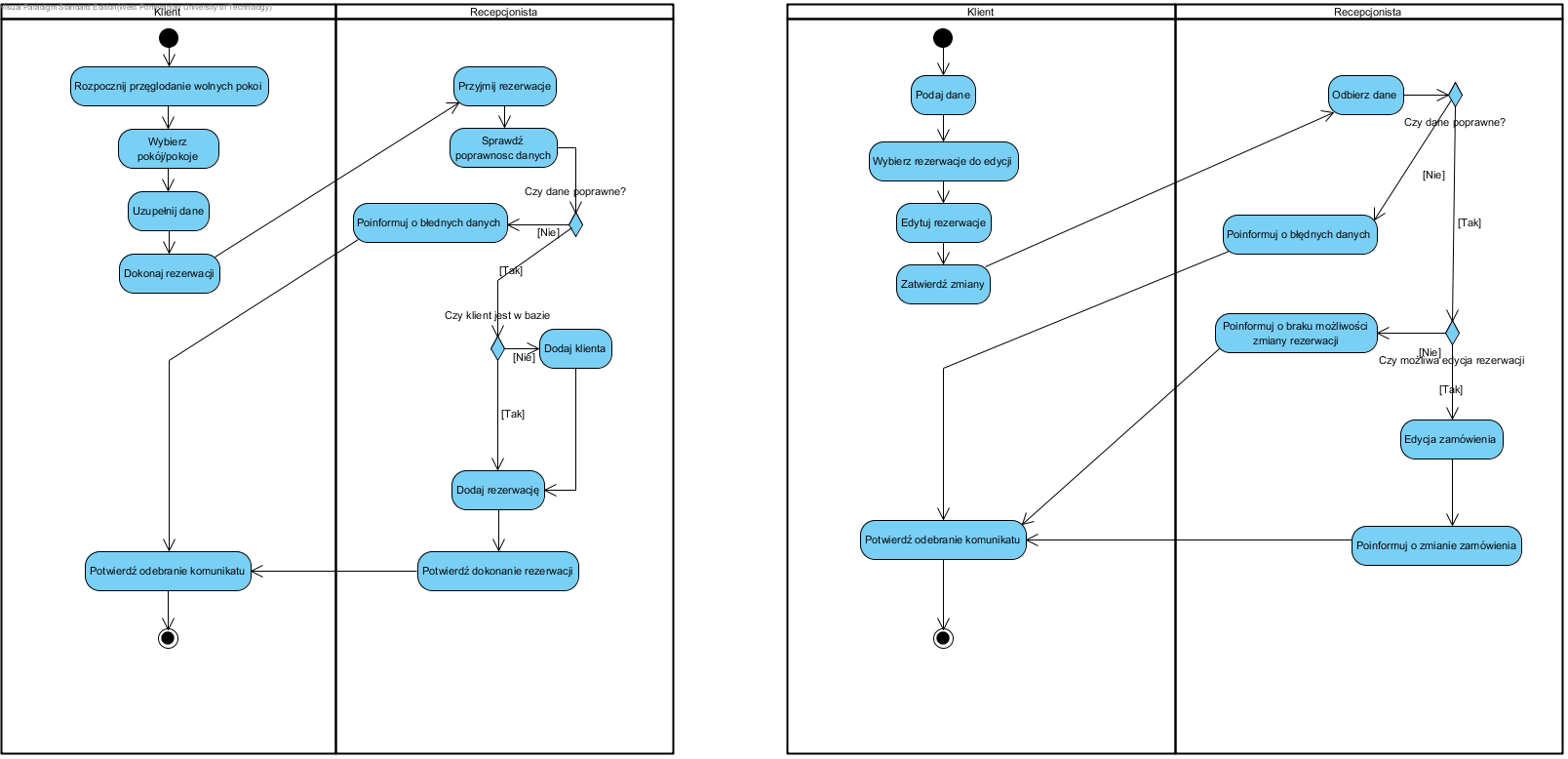
tabela z listą wykorzystanych technologii, każda z uzasadnieniem

## Diagramy UML

### 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

## 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