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. Aplikacja będzie stworzona przy użyciu języka obiektowego C++ oraz narzędzia Visual Paradigm do tworzenia modeli UML.

## 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:

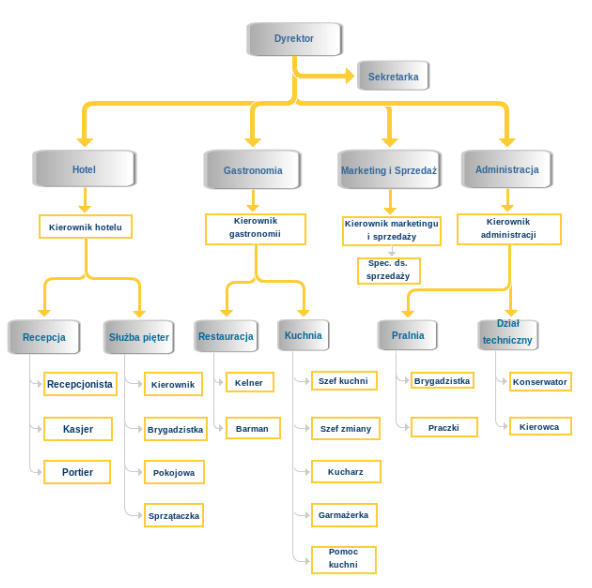
* Szczególnym rodzajem gościnności – gościnność za odpłatność,
* Różnego rodzaju działalnością usługową, główne z nich to usługi bytowe takie jak zapewnienie noclegu i wyżywienia,
* Określonym w czasie i z reguły krótkotrwałym pobytem gości
* Zakresem usług świadczonych przez hotelarzy rozszerzanym o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

1. Dokonywanie rezerwacji
2. Odwoływanie rezerwacji
3. Edycja rezerwacji
4. Przegląd rezerwacji
5. Potwierdzenie rezerwacji
6. Przegląd dostępnych pokoi
7. Obliczanie kosztów rezerwacji
8. Zmiana statusu pokoju
9. Anulowanie rezerwacji
10. Sprawdzenie stanu płatności
11. Dokonanie płatności
12. Przegląd klientów
13. Wystawienie faktury

**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
* Projekt powinien zostać wykonany w języku obiektowym C++, natomiast aplikacja powinna używać protokołów szyfrujących typu TSL, (bezpieczeństwo danych jest w tym wypadku priorytetem)
* 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

każde na nowej stronie wg następujących punktów:

* Numer – jako ID
* Nazwa
* Uzasadnienie biznesowe – odwołanie (-a) do elementów wymienionych w 5.1.4. (id i treść elementu, do którego się odwołujemy)
* Użytkownicy
* Warunki początkowe
* Przebieg działań
* 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

***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

* Działanie systemu na systemach operacyjnych: Windows XP lub nowszy, OS X, Linux.
* Czas szkolenia pracowników – maksymalnie 3 dni.
* 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.
* Praca w czasie rzeczywistym ( jak najmniejsze opóźnienie pomiędzy wprowadzeniem zmiany i wprowadzeniem jej do bazy danych).
* 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.
* Zabezpieczenie danych (haseł, identyfikatorów) dotyczących kont użytkowników.
* Szybka reakcja na działanie użytkownika.
* Automatyczne wykonywanie backup’u systemu.
* Niezawodność- nie więcej niż 1 błąd krytyczny tygodniowo i aktualizacje systemu eliminujące powstawanie błędu w ciągu 2 dni roboczych od momentu jego zgłoszenia.
* Prawdopodobieństwo zniszczenia danych w awarii poniżej 1%.
* Szybkie (odpowiednio do sprzętu) grupowanie dokumentów i danych, ich katalogowani, wyszukiwanie i analizowanie.
* Łatwość implementacji systemu.
* Zgodność systemu z przepisami prawa.
* Proces zakupu produktu trwa nie dłużej niż 2 minuty.
* Zabezpieczenie systemu przed atakami hakerów, wirusami.
* 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

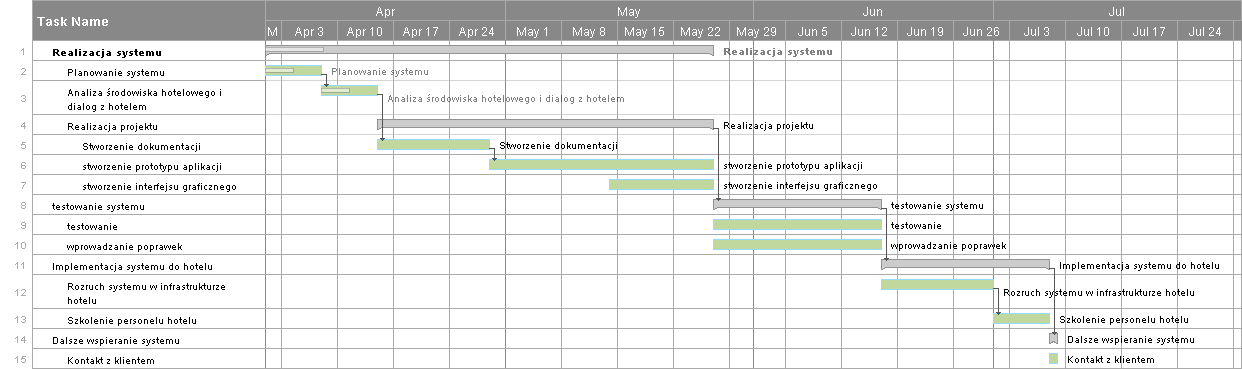
* Kierownik wykonawczy (szef wszystkich działów, na bieżąco egzekwujący efekty pracy danych działów i pracowników, zatwierdza kolejne kamienie milowe i wyznacza oczekiwaną jakość projektu jak i jego podzespołów)
* Szef działu programistów (głównodowodzący zespołem programistów, projektujący konkretne 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

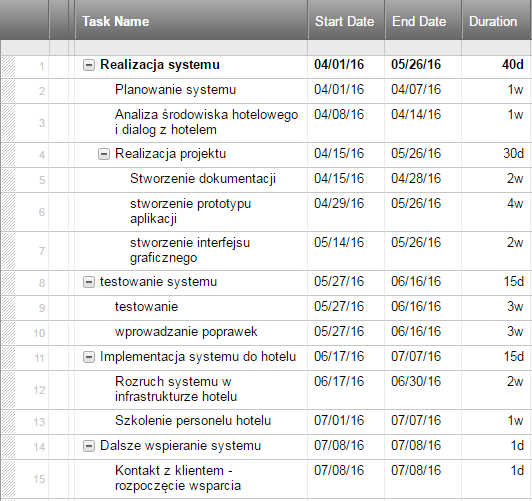
(rzeczywiste lub hipotetyczne) – przy realizacji projektu

## Etapy/kamienie milowe projektu

wg kaskadowego, przyrostowego lub innego cyklu życia

## Harmonogram prac





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

## 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 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

### 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