Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: System wspomagający pracę recepcji w hotelu

Autorzy: Grzegorz Bączek, Patryk Kosut

Grupa: I1-210B

Kierunek: informatyka

Rok akademicki: 2015/2016

Poziom i semestr: I/4

Tryb studiów: stacjonarne

Spis treści

[Dokumentacja projektu zaliczeniowego 1](#_Toc449192390)

[Przedmiot: Inżynieria oprogramowania 1](#_Toc449192391)

[2 Odnośniki do innych źródeł 4](#_Toc449192392)

[3 Słownik pojęć 5](#_Toc449192393)

[4 Specyfikacja wymagań 6](#_Toc449192394)

[4.1 Charakterystyka ogólna 6](#_Toc449192395)

[4.1.1 Definicja produktu 6](#_Toc449192396)

[4.1.2 Użytkownicy 6](#_Toc449192397)

[4.1.3 Korzyści z systemu 6](#_Toc449192398)

[4.1.4 Ograniczenia projektowe i wdrożeniowe 6](#_Toc449192399)

[4.2 Wymagania funkcjonalne 7](#_Toc449192400)

[4.2.1 Lista wymagań 7](#_Toc449192401)

[4.2.2 Diagramy przypadków użycia 8](#_Toc449192402)

[4.2.3 Szczegółowy opis wymagań 10](#_Toc449192403)

[4.3 Wymagania niefunkcjonalne 11](#_Toc449192404)

[5 Zarządzanie projektem 12](#_Toc449192405)

[5.1 Zasoby ludzkie 12](#_Toc449192406)

[5.2 Etapy/kamienie milowe projektu 12](#_Toc449192407)

[5.3 Harmonogram prac 12](#_Toc449192408)

[6 Zarządzanie ryzykiem 13](#_Toc449192409)

[6.1 Lista czynników ryzyka 13](#_Toc449192410)

[6.2 Ocena ryzyka 13](#_Toc449192411)

[6.3 Plan reakcji na ryzyko 13](#_Toc449192412)

[7 Zarządzanie jakością 14](#_Toc449192413)

[7.1 Definicje 14](#_Toc449192414)

[7.1.1 Priorytety defektów/awarii 14](#_Toc449192415)

[7.1.2 Istotność/znaczenie problemu 14](#_Toc449192416)

[7.2 Scenariusze testowe 14](#_Toc449192417)

[7.3 Proces obsługi defektów/awarii 14](#_Toc449192418)

[8 Projekt techniczny 15](#_Toc449192419)

[8.1 Opis architektury systemu 15](#_Toc449192420)

[8.2 Technologie implementacji systemu 15](#_Toc449192421)

[8.3 Diagramy UML 15](#_Toc449192422)

[8.3.1 Diagram(-y) klas 16](#_Toc449192423)

[8.3.2 Diagram(-y) czynności 16](#_Toc449192424)

[8.3.3 Diagramy sekwencji 16](#_Toc449192425)

[8.3.4 Inne diagramy 17](#_Toc449192426)

[8.4 Charakterystyka zastosowanych wzorców projektowych 17](#_Toc449192427)

[8.5 Projekt bazy danych 17](#_Toc449192428)

[8.5.1 Schemat 17](#_Toc449192429)

[8.5.2 Projekty szczegółowe tabel 17](#_Toc449192430)

[8.6 Projekt interfejsu użytkownika 17](#_Toc449192431)

[8.6.1 Lista głównych elementów interfejsu 17](#_Toc449192432)

[8.6.2 Projekty szczegółowe poszczególnych elementów 17](#_Toc449192433)

[8.7 Procedura wdrożenia 17](#_Toc449192434)

[9 Dokumentacja dla użytkownika 18](#_Toc449192435)

[10 Podsumowanie 19](#_Toc449192436)

[10.1 Szczegółowe nakłady projektowe członków zespołu 19](#_Toc449192437)

[11 Inne informacje 20](#_Toc449192438)

# Odnośniki do innych źródeł

* + Zarządzania projektem – Visual Paradigm
  + Wersjonowanie kodu – https://github.com/Bumbl3Bee/IO-Hotel

# Słownik pojęć

Tabela lub lista z pojęciami, które wymagają wyjaśnienia, wraz z tymi wyjaśnieniami – w szczególności synonimy różnych pojęć używanych w dokumentacji.

|  |  |
| --- | --- |
| Pojęcie | Wytłumaczenie |
| Administrator | Posiada najwyższe uprawnienia, może tworzyć konta, przyznawać uprawnienia itp., |
| Recepcjonista | Może zarezerwować pokoje, meldować i wymeldowywać, przyjmować płatności, udzielać informacji odnoście usług |
| Kierownik recepcji | Może edytować informacje o pokojach, dodawać i usuwać pokoje, zmieniać ich status |
| Obsługa | Może edytować status pokoi |
| Klient | Gość hotelu |

# Specyfikacja wymagań

## Charakterystyka ogólna

### Definicja produktu

System ma za zadanie ułatwić pracę recepcjonisty w hotelu, tj. zarządzanie pokojami, rezerwacją, regulacją płatnościami, dostarczaniem informacji.

### Użytkownicy

|  |  |
| --- | --- |
| Użytkownik | Opis |
| Administrator | Osoba, która dba o poprawność działania systemu. |
| Recepcjonista | Może zarezerwować pokoje, meldować i wymeldowywać, przyjmować płatności, udzielać informacji odnoście usług |
| Kierownik recepcji | Może edytować informacje o pokojach, dodawać i usuwać pokoje, zmieniać ich status |
| Obsługa | Może edytować status pokoi |

### Korzyści z systemu

Administrator

* Podgląd wszystkich zasobów systemu,
* Możliwość pełnego zarządzania kontami, pokojami,
* Zmienianie danych osobowych,

Recepcjonista

* Sprawność zarządzania obsługą rezerwacji
* Szybkość przyjmowania i egzekwowania płatności
* Łatwość dostępu do informacji na temat stanu pokoi

Kierownik recepcji

* Możliwość edycji parametrów pokoi
* Łatwość w zarządzaniu recepcją
* Intuicyjność w wyszukiwaniu i pozyskiwaniu informacji

### Ograniczenia projektowe i wdrożeniowe

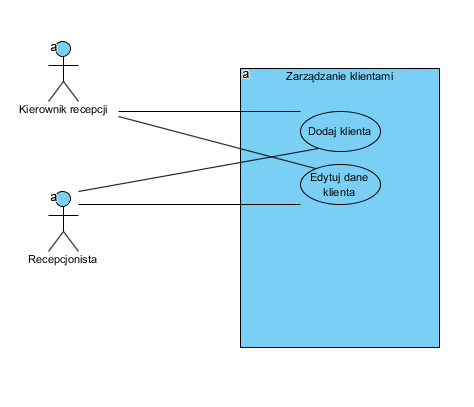
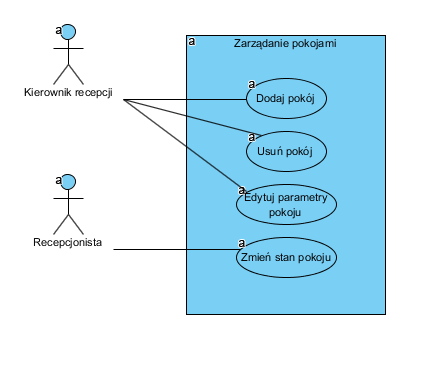
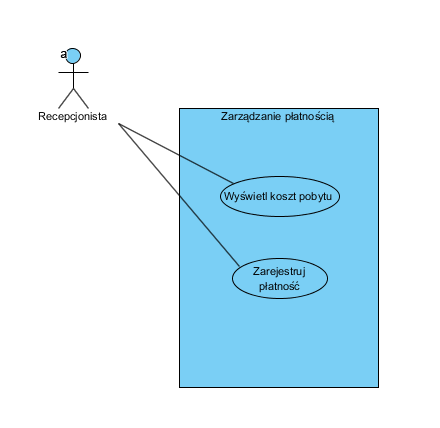
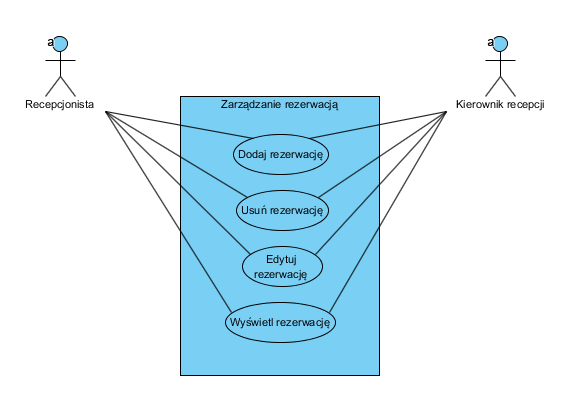
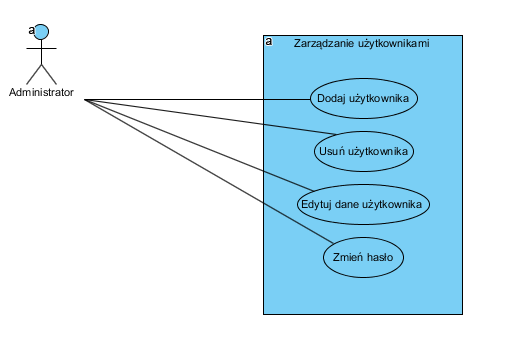
* Niechęć pracowników firmy do używania nowego systemu,
* System jest niezgodny z faktycznymi wymaganiami klienta,
* Niezadowalająca jakość systemu,
* System może posiadać luki w zabezpieczeniach danych klientów,
* Zbyt duża konkurencja na rynku,
* Możliwość nie wykonania projektu w terminie,

## Wymagania funkcjonalne

### Lista wymagań

1. Dodaj użytkownika
2. Usuń użytkownika
3. Edytuj dane użytkownika
4. Dodaj klienta
5. Edytuj dane klienta
6. Zmień hasło
7. Dodaj pokój
8. Usuń pokój
9. Edytuj parametry pokoju
10. Wyświetl stan pokoju
11. Wyświetl rezerwacje
12. Dodaj rezerwację
13. Usuń rezerwację
14. Edytuj rezerwację
15. Wyświetl koszt pobytu
16. Zarejestruj płatność

### Diagramy przypadków użycia

### 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

Wymagania produktowe

Wymagania organizacyjne

Wymagania zewnętrzne

# 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

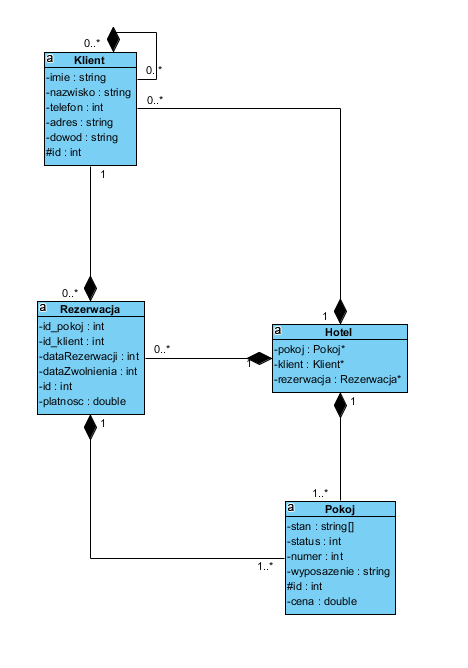
tabela z listą wykorzystanych technologii, każda z uzasadnieniem

## Diagramy UML

każdy diagram ma mieć tytuł 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 pokazującam.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