Dokumentacja projektu zaliczeniowego

Przedmiot: Inżynieria oprogramowania

Temat: **Oprogramowanie do wspierania placówki medycznej.**

Autorzy: **Sanocki Maciej, Deluga Bartosz**

Grupa: I1-211B

Kierunek: informatyka

Rok akademicki: 2

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](#_Toc1976793)

[3 Słownik pojęć 5](#_Toc1976794)

[4 Wprowadzenie 6](#_Toc1976795)

[4.1 Cel dokumentacji 6](#_Toc1976796)

[4.2 Przeznaczenie dokumentacji 6](#_Toc1976797)

[4.3 Opis organizacji lub analiza rynku 6](#_Toc1976798)

[4.4 Analiza SWOT organizacji 6](#_Toc1976799)

[5 Specyfikacja wymagań 7](#_Toc1976800)

[5.1 Charakterystyka ogólna 7](#_Toc1976801)

[5.2 Wymagania funkcjonalne 7](#_Toc1976802)

[5.3 Wymagania niefunkcjonalne 8](#_Toc1976803)

[6 Zarządzanie projektem 9](#_Toc1976804)

[6.1 Zasoby ludzkie 9](#_Toc1976805)

[6.2 Harmonogram prac 9](#_Toc1976806)

[6.3 Etapy/kamienie milowe projektu 9](#_Toc1976807)

[7 Zarządzanie ryzykiem 10](#_Toc1976808)

[7.1 Lista czynników ryzyka 10](#_Toc1976809)

[7.2 Ocena ryzyka 10](#_Toc1976810)

[7.3 Plan reakcji na ryzyko 10](#_Toc1976811)

[8 Zarządzanie jakością 11](#_Toc1976812)

[8.1 Scenariusze i przypadki testowe 11](#_Toc1976813)

[9 Projekt techniczny 12](#_Toc1976814)

[9.1 Opis architektury systemu 12](#_Toc1976815)

[9.2 Technologie implementacji systemu 12](#_Toc1976816)

[9.3 Diagramy UML 12](#_Toc1976817)

[9.4 Charakterystyka zastosowanych wzorców projektowych 12](#_Toc1976818)

[9.5 Projekt bazy danych 12](#_Toc1976819)

[9.6 Projekt interfejsu użytkownika 13](#_Toc1976820)

[9.7 Procedura wdrożenia 13](#_Toc1976821)

[10 Dokumentacja dla użytkownika 14](#_Toc1976822)

[11 Podsumowanie 15](#_Toc1976823)

[11.1 Szczegółowe nakłady projektowe członków zespołu 15](#_Toc1976824)

[12 Inne informacje 16](#_Toc1976825)

# Odnośniki do innych źródeł

tj. do wykorzystywanych narzędzi / projektów w tych narzędziach

* + Zarządzania projektem – Jira, Trello, itp.
  + Wersjonowanie kodu – sugerowany Git (hosting np. na Bitbucket lub GitHub), ew. SVN
  + System obsługi defektów – np. Bitbucket, GitHub, Bugzilla.

# 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

Dokumentacja zawiera informacje dotyczące placówek medycznych. Zaczynając od Analizy SWOT, aż przez opis organizacji. Oprócz tego dokumentacja posiada również stronę techniczną, która przedstawia zasadę działania oraz opis aplikacji, która jest głównym celem całej pracy. Sama dokumentacja ma na celu zobrazowanie zasady działania takiego programu oraz przedstawienie możliwości jakie placówki medyczne będą mogły osiągnąć używając tej właśnie aplikacji.

## Przeznaczenie dokumentacji

Dokumentacja przeznaczona jest dla pracowników, jednostki medycznej używających oprogramowania.

## Opis organizacji lub analiza rynku

Oprogramowanie jest zorientowane na rynek masowy.

Oprogramowanie jest przeznaczone dla placówek wielopowierzchniowych zatrudniających powyżej 50 lekarzy i posiadającą bazę pacjentów nie mniejszą niż 10 tyś.

Organizacje te posiadają zróżnicowane oddziały wraz z wyspecjalizowanym personelem.

Celem działań organizacji jest udzielanie świadczeń zdrowotnych w warunkach zamkniętych, a samo przyjęcie pacjenta następuje po wcześniejszym otrzymaniu skierowania od lekarza rodzinnego. Następnie pacjent kierowany jest na wybrany oddział, gdzie otrzymuje zakwaterowanie, a także lekarza z danego oddziału. Określana jest metoda leczenia, następnie po stwierdzeniu, że pacjent jest gotowy do wykwaterowania – otrzymuje on wypis, natomiast historia leczenia jest archiwizowana przez 20 lat.

Organizacje takie są rozproszone po całej Polsce, przy czym nie powinno to stanowić problemu dla samego wdrożenia oprogramowania, które jest zorientowane na uniwersalność.

## Analiza SWOT organizacji

|  |  |
| --- | --- |
| **SZANSE** | **ZAGROŻENIA** |
| * Inwestycje w najnowszą technologie za sprawą dofinansowań ze strony państwa lub prywatnych jednostek * Promocja na skalę światową * Napływ propozycji współpracy z uczelniami medycznymi * Promocja zdrowia | * Zagrożenia biologiczne powodujące przeciążenie danych jednostek szpitalnych * Rezygnacja z profilaktyki oraz leczenia publicznego, spowodowana długim czasem oczekiwania (korzystanie z prywatnych placówek medycznych) * Redukcja poziomu zatrudnienia spowodowana szpitalną siatką płac |

# Specyfikacja wymagań

## Charakterystyka ogólna

### Definicja produktu

AID System ™ – System wspomagania zarządzania.

### Podstawowe założenia

AID System jest to oprogramowanie do wspomagania pracy szpitala.

System jest odpowiedzialny za:

* Ewidencję godzin pracy,
* Zarządzanie bazą pacjentów,
* Zarządzaniem bazą pracowników,
* Zarządzaniem asortymentem szpitala,
* Umożliwienie komunikacji w ramach jednego oddziału jak i komunikacji międzyoddziałowej.

Oprogramowanie jest formą nakładki na system Windows 10, która wymaga zalogowania się swoim unikalnym kodem UUID i hasłem. Po zalogowaniu się dany użytkownik ma dostęp do konta, które posiada tylko przyznane mu uprawnienia w ramach, których może się on poruszać po systemie. Historia leczenia pacjentów jest utrzymywana przez 20 lat i może być ona udostępniona tylko osobom do tego upoważnionym ( np. Policja).

### Cel biznesowy

Decydując się na wdrożenie systemu, organizacja szuka sposobu na usprawnienie pracy całej jednostki, a tym samym stać się bardziej atrakcyjną dla swoich pacjentów.

### Użytkownicy

* Kierownik Szpitala
* Lekarz
* Pielęgniarka
* Woźny
* Administrator Systemu

### Korzyści z systemu

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | Korzyść | Kierownik Szpitala | Lekarz | Pielęgniarka | Woźny | Administrator Systemu |
| 1 | Ewidencja Pracy | X | X | X | X | X |
| 2 | Dostęp do bazy pacjentów | X | X | X |  | X |
| 3 | Dostęp do bazy asortymentów | X | X | X | X | X |
| 4 | Dostęp do bazy pracowników | X |  |  |  | X |
| 5 | Dostęp do komunikacji | X | X | X |  | X |

### Ograniczenia projektowe i wdrożeniowe

W razie potrzeby udostępnienia danych dla innego wewnętrznego systemu,

AID System umożliwia udostępnienie:

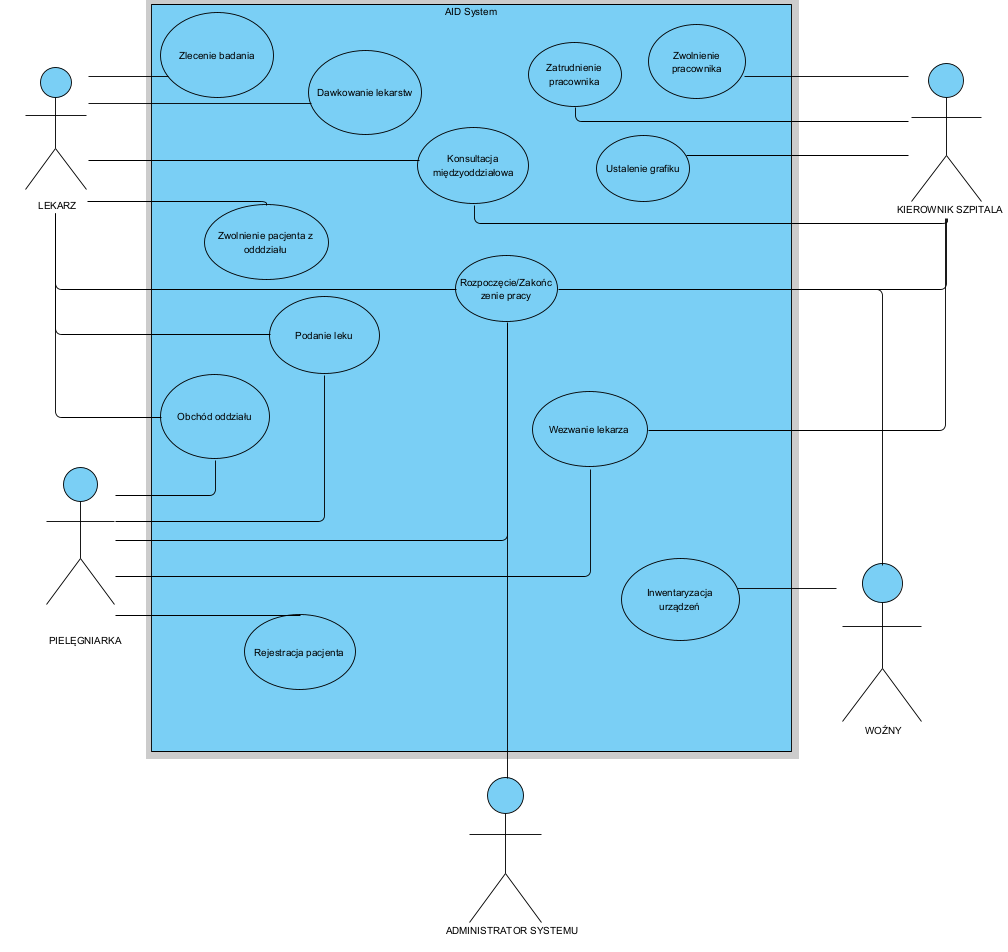
* Ewidencji pracy
* Bazy użytkowników
* Dostępu do asortymentów

## Wymagania funkcjonalne

### Lista wymagań

1. Rozpoczęcie/Zakończenie pracy.
2. Rejestracja pacjenta.
3. Zlecenie badania.
4. Zwolnienie pacjenta z oddziału.
5. Inwentaryzacja urządzeń.
6. Podanie leku.
7. Dawkowanie lekarstw.
8. Konsultacja międzyoddziałowa.
9. Wezwanie lekarza.
10. Zwolnienie pracownika.
11. Zatrudnienie pracownika.
12. Ustalanie grafiku.
13. Obchód oddziału.

### Diagramy przypadków użycia



### Szczegółowy opis wymagań

dla 5-7 wybranych najważniejszych przypadków użycia

każde na nowej stronie wg następujących punktów:

* Numer – jako ID
* Nazwa
* Uzasadnienie biznesowe – odwołanie (-a) do elementów wymienionych w 5.1.5. (id i treść elementu, do którego się odwołujemy)
* Użytkownicy
* Scenariusze, dla każdego z nich:
* Nazwa scenariusza
* Warunki początkowe
* **Przebieg działań –** numerowana lista kroków, ze wskazaniem, kto realizuje dany krok
* 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

***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

W odniesieniu do całego systemu, modułów lub innych składowych 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

Należy założyć, że projekt byłby realizowany w całości jako projekt komercyjny a nie tylko częściowo w ramach zajęć na uczelni

## Harmonogram prac

Etapy mogą się składać z zadań.

Wskazać czasy trwania poszczególnych etapów i zadań – wykres Gantta.

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.)

## Etapy/kamienie milowe projektu

dla głównych etapów projektu

# Zarządzanie ryzykiem

## Lista czynników ryzyka

Wypełniona lista kontrolna

## Ocena ryzyka

prawdopodobieństwo i wpływ

## Plan reakcji na ryzyko

Działania w odniesieniu do poszczególnych ryzyk.

Mogą być wg różnych strategii, tj. kilka strategii dla pojedynczego czynnika ryzyka

# Zarządzanie jakością

## Scenariusze i przypadki testowe

szczegółowy plan testowania systemu – głównie testowanie funkcjonalności; każdy scenariusz od nowej strony, musi zawierać co najmniej następujące informacje (sugerowany układ tabelaryczny, np. wg szablonu podanego w osobnym pliku lub na wykładzie):

* numer – jako ID
* nazwa scenariusza – co test w nim testowane (max kilka wyrazów)
* kategoria – poziom/kategoria testów
* 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
* założenia, środowisko, warunki wstępne, dane wejściowe – przygotowanie przed uruchomieniem testów
* 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*

# 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

1 lub więcej

### Diagram(-y) czynności

1 lub więcej

### Diagramy sekwencji

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

### 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

w zależności, czy następujące elementy są widoczne na schemacie b.d.: nazwa tabeli, nazwy pól, typ danych, wartości NULL, klucz główny, klucz obcy –

- jeśli TAK: i nie ma potrzeby pokazania dodatkowych elementów b.d., to ten punkt może być pusty,

- jeśli NIE: to podać te elementy, których nie widać na schemacie.

dodatkowymi elementami mogą być np. triggery, procedury, funkcje, indeksy, użytkownicy, role.

## Projekt interfejsu użytkownika

co najmniej dla głównej funkcjonalności programu – w razie wątpliwości, uzgodnić z prowadzącym zajęcia

### Lista głównych elementów interfejsu

okien, stron, aktywności (Android)

### Przejścia między głównymi elementami

np. storyboard, schemat blokowy lub inna notacja

### Projekty szczegółowe poszczególnych elementów

dla 5-7 głównych elementów

każdy element od nowej strony z następującą minimalną zawartością:

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

Opcjonalnie – dla chętnych

Na podstawie projektu docelowej aplikacji, a nie zaimplementowanego prototypu architektury

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 oraz wiersz podsumowania – procentowy udział każdej osoby w skali całego projektu

# Inne informacje

przydatne informacje, które nie zostały ujęte we wcześniejszych punktach