Plan projektu

*Pharmacy*

Wersja v\_31\_03\_23\_1

Data 31.03.2023

Spis treści

[1. Wstęp 3](#_Toc130745239)

[2. Cel projektu 3](#_Toc130745240)

[3. Organizacja projektu 3](#_Toc130745241)

[4. Struktura organizacyjna 3](#_Toc130745242)

[5. Role i odpowiedzialność 3](#_Toc130745243)

[6. Harmonogram projektu 3](#_Toc130745244)

[7. Kosztorys projektu 3](#_Toc130745245)

[8. Zasoby projektu 3](#_Toc130745246)

[9. Standardy i narzędzia w projekcie 3](#_Toc130745247)

[10. Procesy zarządzania 3](#_Toc130745248)

[10.1 Plan zarządzania konfiguracją 3](#_Toc130745249)

[10.2 Plan zarządzania ryzykiem 3](#_Toc130745250)

[10.3 Plan zarządzania testami 3](#_Toc130745251)

# 1. Wstęp

Projekt dotyczy stworzenia aplikacji dla pracowników apteki umożliwiającej logowanie, rejestrację, dodawanie antybiotyków i suplementów oraz przeglądanie ich.

# 2. Cel projektu

Celem projektu jest dostarczenie kompletnego systemu informatycznego dla pracowników apteki, który umożliwi zarządzanie danymi dotyczącymi antybiotyków, suplementów oraz pracowników, którym zapewnia funkcje logowania i rejestracji.

# 3. Organizacja projektu

Projekt będzie organizowany i zarządzany przez zespół składający się z Business analyst, Change Manager, Configuration Manager, Database developer, Developer, Instalation engineer, Project Manager, QA Manager, Risk Analyst, Risk Manager, System Analyst, System Architect, Test Designer, Test Engineer, Test Manager

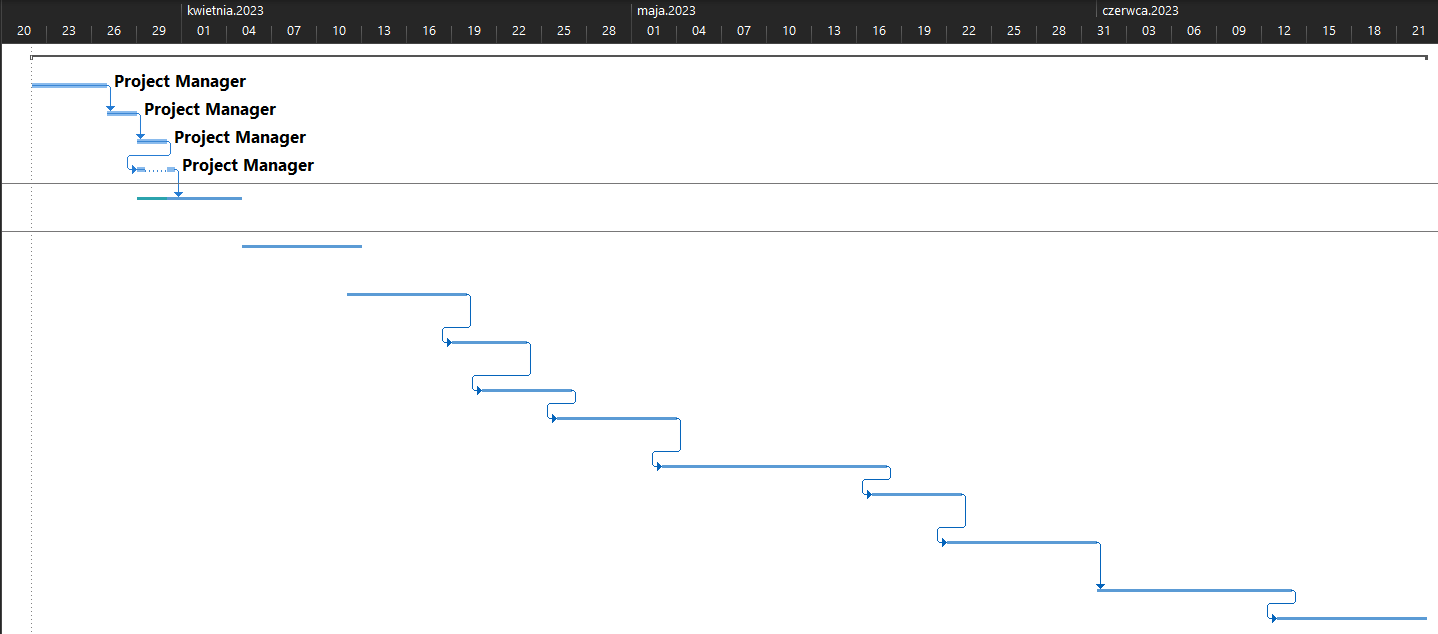
# 4. Struktura organizacyjna

* Business analyst - osoba odpowiedzialna za analizę wymagań jakości i opracowywanie specyfikacji funkcjonalnych i niefunkcjonalnych.
* Change Manager - osoba odpowiedzialna za zarządzanie procesem zmian w projekcie.
* Configuration Manager - osoba odpowiedzialna za zarządzanie konfiguracją aplikacji i kontrolę wersji.
* Database developer;Developer - osoby odpowiedzialne za projektowanie i rozwijanie bazy danych oraz implementację funkcjonalności systemu.
* Instalation engineer - osoba odpowiedzialna za instalację i konfigurację systemu na serwerach i urządzeniach klienta.
* Project Manager - osoba odpowiedzialna za ogólne zarządzanie projektem, koordynację zespołów i monitorowanie postępu prac.
* QA Manager - osoba odpowiedzialna za planowanie, koordynację i monitorowanie działań związanych z kontrolą jakości.
* Risk Analyst - osoba odpowiedzialna za identyfikację, analizę i ocenę ryzyka w projekcie.
* Risk Manager - osoba odpowiedzialna za stworzenie planu zarządzania ryzykiem i implementację strategii zarządzania ryzykiem.
* System Analyst - osoba odpowiedzialna za analizę systemową, identyfikację wymagań systemowych i opracowanie specyfikacji funkcjonalnych.
* System Architect - osoba odpowiedzialna za projektowanie architektury systemu i zapewnienie integralności systemu.
* Test Designer - osoba odpowiedzialna za projektowanie przypadków testowych i tworzenie planów testowych.
* Test Engineer - osoba odpowiedzialna za przeprowadzanie testów, raportowanie błędów i weryfikację zgodności z wymaganiami.
* Test Manager - osoba odpowiedzialna za zarządzanie procesem testowania, planowanie zasobów testowych i monitorowanie jakości testów.

# 5. Role i odpowiedzialność

1. Business analyst:
   * Rola i zakres obowiązków: sporządza specyfikacje wymagań funkcjonalnych, wymagań dotyczących jakości oprogramowania, wymagań niefunkcjonalnych i więzów, sporządza umowy oraz akceptuje specyfikacje wymagań.
   * Kompetencje: Doskonała komunikacja, umiejętność analizy biznesowej, znajomość technik zbierania wymagań.
2. Change Manager
   * Rola i zakres obowiązków: sporządza plan zarządzania zmianami oraz formularz żądania zmiany.
   * Kompetencje: Umiejętność zarządzania zmianami, analiza ryzyka, doskonała komunikacja.
3. Configuration Manager:
   * Rola i zakres obowiązków: sporządza plan zarządzania konfiguracja, tworzy system zarządzania konfiguracją oraz zarządzania wersjami i tworzy zbiór ‘baseline’ dla projektu.
   * Kompetencje: Znajomość narzędzi do kontroli wersji, umiejętność zarządzania konfiguracją.
4. Database developer, Developer
   * Rola i zakres obowiązków: Pisze kod aplikacji, przygotowuje opis wersji aplikacji, tworzy dokumentacje użytkownika, wykonuje unit testy.
   * Kompetencje: Znajomość języków zapytań SQL, umiejętność projektowania baz danych, znajomość odpowiedniego języka programowania, umiejętność rozwiązywania problemów programistycznych.
5. Instalation engineer:
   * Rola i zakres obowiązków: przygotowuje system klienta
   * Kompetencje: Znajomość systemów operacyjnych, umiejętność instalacji i konfiguracji aplikacji.
6. Project Manager:
   * Rola i zakres obowiązków: przygotowuje plan projektu, harmonogram projektu, kosztorys projektu oraz akceptuje plan projektu i opis wersji aplikacji.
   * Kompetencje: Umiejętność zarządzania projektem, przywództwo, planowanie zasobów.
7. QA Manager:
   * Rola i zakres obowiązków: sporządza plan zapewnienia jakości, kontroluje i dokumentuj jakość, sporządza plan zarządzania jakością i akceptuje plan zarządzania jakością
   * Kompetencje: Znajomość metod testowania, analiza ryzyka, umiejętność zarządzania jakością.
8. Risk Analyst:
   * Rola i zakres obowiązków: sporządza raport zarządzania ryzykiem.
   * Kompetencje: Umiejętność analizy ryzyka, ocena skutków ryzyka.
9. Risk Manager
   * Rola i zakres obowiązków: sporządza plan zarządzania ryzykiem.
   * Kompetencje: Umiejętność zarządzania ryzykiem, planowanie i podejmowanie działań zaradczych.
10. System Analyst:
    * Rola i zakres obowiązków: sporządza model use case’u, sporządza analizę specyfikacji wymagań, tworzy statyczny model systemu, tworzy dynamiczny model systemu, akceptuje analizę specyfikacji wymagań.
    * Kompetencje: Znajomość technik analizy systemowej, umiejętność modelowania systemów.
11. System Architect:
    * Rola i zakres obowiązków: tworzy prototyp interfejsów użytkownika, sporządza specyfikację interfejsów systemowych.
    * Kompetencje: Znajomość wzorców architektonicznych, umiejętność projektowania systemów.
12. Test Designer:
    * Rola i zakres obowiązków: przygotowuje testy.
    * Kompetencje: Znajomość technik testowania, umiejętność projektowania testów.
13. Test Engineer:
    * Rola i zakres obowiązków: tworzy zbiór test case’ów.
    * Kompetencje: Umiejętność przeprowadzania testów, analiza błędów, znajomość narzędzi do testowania.
14. Test Manager:
    * Rola i zakres obowiązków: sporządza plan zarządzania testami oraz harmonogram testów i tworzy bazę błędów.
    * Kompetencje: Umiejętność zarządzania testami, ocena skuteczności testów.

# 6. Harmonogram projektu



Pełny harmonogram projektu znajduje się w pliku „Harmonogram Projektu v\_29\_03\_23\_1”

# 7. Kosztorys projektu

|  |  |
| --- | --- |
| **Nazwy zasobów** | **Godzinowy nakład pracy** |
| Business analyst | 40 godz. |
| Change Manager | 24 godz. |
| Configuration Manager | 64 godz. |
| Database developer;Developer | 112 godz. |
| Instalation engineer | 8 godz. |
| Project Manager | 76 godz. |
| QA Manager | 40 godz. |
| Risk Analyst | 6 godz. |
| Risk Manager | 18 godz. |
| System Analyst | 40 godz. |
| System Architect | 16 godz. |
| Test Designer | 8 godz. |
| Test Engineer | 48 godz. |
| Test Manager | 40 godz. |

# 8. Zasoby projektu

* Dwa komputery ze systemem operacyjnym minimum Windows 10.
* Urządzenia wejścia:
  + Mysz
  + Klawiatura
* Urządzenia wyjścia:
  + Monitor
* Program XAMPP(Wersja 8.0.28) który służy do tworzenia bazy danych oraz zarządzania nią za pomocą phpMyAdmin.
* Środowisko programistyczne: Visual Studio Community 2022 – 17.4.4
* Wybrane program z pakietu Microsoft Office 2022:
  + Word
  + Excel
  + MsProject
* Narzędzia do kontroli wersji: Git (Wersja 2.41.0)

# 9. Standardy i narzędzia w projekcie

* Język programowania: C#
* Framework: .NET 4.8 Windows Forms
* Środowisko programistyczne: Visual Studio Community 2022 – 17.4.4
* Baza danych: MySQL, XAMPP(Wersja 8.0.28)
* Metodologia zarządzania projektem: Model kaskadowy
* Narzędzia do kontroli wersji: Git (Wersja 2.41.0)

# 10. Procesy zarządzania

## 10.1 Plan zarządzania konfiguracją

* Zarządzanie wersjami: Wykorzystanie systemu kontroli wersji Git do śledzenia zmian w kodzie źródłowym, utrzymywania historii zmian i umożliwienia współpracy między członkami zespołu. Wersja aplikacji jest nadawana w formacie 1.0 gdzie mało znaczące aktualizacje zwiększają liczbę po kropce o jeden w górę (np. 1.1), a bardzo znaczące aktualizację zwiększają o jeden liczbę przed kropką i zerują liczbę po kropce (np. 2.0). Wersja dokumentu jest nadawana na podstawie daty oraz kolejnego numeru który następuje po poprzedniej wersji w danym dniu (np. v\_24\_06\_23\_1 lub v\_24\_06\_23\_2). Na koniec dnia należy zrobić „commit” dokonanych zmian.
* Hierarchia uprawnień: Developerzy mają pełny dostęp do kodu źródłowego, a członkowie zespołu testowego mogą tylko odczytywać kod.
* Weryfikacja tożsamości: Każdy członek zespołu ma przypisaną unikalną tożsamość użytkownika – login i hasło, aby śledzić ich działania w systemie kontroli wersji.
* Monitorowanie dostępu: Regularne monitorowanie działań w systemie kontroli wersji w celu wykrywania nieprawidłowości lub podejrzanej aktywności. To pozwoli na szybkie reagowanie na ewentualne naruszenia zabezpieczeń.
* Zarządzanie konfiguracją oprogramowania: Dokładne rejestrowanie i kontrola zmian w konfiguracji oprogramowania, w tym uaktualnień, poprawek i dodatkowych modułów.

Używamy numeracji wersji i wykorzystujemy narzędzia do zarządzania konfiguracją.

* Przewidziane są szkolenia dotyczące konfiguracji dla członków zespołu oraz plany komunikacji dotyczące wprowadzanych zmian w konfiguracji aplikacji.

## 10.2 Plan zarządzania ryzykiem

* Organizacja procesu zarządzania ryzykiem:

W procesie zarządzania ryzykiem najpierw identyfikujemy potencjalne ryzyko, które może wystąpić w projekcie. Konsultujemy się z członkami zespołu projektowego i przeprowadzamy burzę mózgów.

* Ocena ryzyka: Zidentyfikowane ryzyko jest oceniane pod względem prawdopodobieństwa jego wystąpienia oraz jego wagi (szkody jaką może wyrządzić w projekcie wystąpienie danego ryzyka).
* Minimalizacja ryzyka: Określenie czynności, które mają na celu zminimalizować prawdopodobieństwo wystąpienia ryzyka oraz zmniejszyć jego wagę. W celu realizacji działań minimalizujących ryzyko, konieczne jest określenie odpowiednich kosztów związanych z ich wykonaniem.
* Po minimalizacji ryzyka należy ponownie przeprowadzić ocenę ryzyka.
* Kontrola ryzyka: Definicja miar ryzyka
  + Prawdopodobieństwo ryzyka:
    - Bardzo mało prawdopodobne (25%), wartość: 1.
    - Mało prawdopodobne (50%), wartość: 2.
    - Prawdopodobne (75%), wartość: 3.
    - Bardzo prawdopodobne (99%), wartość: 4.
  + Waga ryzyka:
    - Mała szkoda, wartość: 1.
    - Średnia szkoda, wartość: 2.
    - Poważna szkoda, wartość: 4.
    - Krytyczna szkoda, wartość: 8.
* Dokumentacja ryzyka:
  + Ryzyka projektu znajdują się w pliku „Ryzyka projektu v\_04\_04\_23\_1.xmls”.

## 10.3 Plan zarządzania testami

* Strategia testowania: Wykonanie testów jednostkowych
* Planowanie testów:
  + Harmonogram testów: Harmonogram testów znajduję się w pliku „Harmonogram Projektu v\_29\_03\_23\_1”
  + Środowisko:
    - Nazwa: Bugzilla
    - Firma: Mozilla
    - Opis: Narzędzie wsparcia procesu wytwarzania oprogramowania, testów, w tym zarządzania błędami.
    - Platforma: Windows
    - Adres: http://www.bugzilla.org/
* Przewidziane są szkolenia dotyczące testów dla członków zespołu testowego.
* Raportowanie i monitorowanie: Przesyłanie raportu z wykonanego testu trafia do przełożonego. Szablon raportu znajduje się „Test case - szablon”.