

Laboratorium 1

**Wykonanie opisu biznesowego „świata rzeczywistego” projektowanego oprogramowania,
definicja wymagań funkcjonalnych i нефункциональных projektowanego oprogramowania**

Spis treści

1.	Inżynieria oprogramowania – podstawowe informacje.....	2
2.	Cel laboratorium	3
3.	Materiały pomocnicze	6
3.1.	Przykłady opisu świata rzeczywistego wraz z definicją wymagań funkcjonalnych i нефункциональных.....	9

1. Inżynieria oprogramowania – podstawowe informacje

Inżynieria oprogramowania (ang. Software Engineering) to dziedzina zajmująca się systematycznym, zorganizowanym i efektywnym podejściem do tworzenia, rozwoju oraz utrzymania oprogramowania. Obejmuje cały cykl życia oprogramowania, od analizy wymagań, poprzez projektowanie, implementację, testowanie, wdrażanie, aż po konserwację i rozwój. Celem inżynierii oprogramowania jest tworzenie oprogramowania wysokiej jakości, które jest zgodne z wymaganiami użytkownika, efektywne, łatwe w utrzymaniu oraz niezawodne.

Kluczowe pojęcia w inżynierii oprogramowania:

1. **Cykl życia oprogramowania (SDLC)** – proces składający się z kolejnych etapów tworzenia oprogramowania:
 - Analiza wymagań – zrozumienie potrzeb klienta i stworzenie dokumentacji wymagań.
 - Projektowanie systemu – opracowanie architektury i struktury systemu.
 - Implementacja – kodowanie zgodne z projektami.
 - Testowanie – weryfikacja, czy oprogramowanie działa zgodnie z założeniami.
 - Wdrożenie – instalacja i konfiguracja oprogramowania u klienta.
 - Utrzymanie – poprawianie błędów i rozwijanie funkcjonalności po wdrożeniu.
2. **Modele procesów tworzenia oprogramowania:**
 - Kaskadowy – model, w którym każdy etap następuje po poprzednim, bez możliwości powrotu do wcześniejszych faz.
 - Spiralny – model iteracyjny, w którym projektowanie i implementacja są powtarzane, uwzględniając wyniki poprzednich faz.
 - Agile (zwinne) – model iteracyjny i przyrostowy, gdzie praca jest podzielona na krótkie sprinty, a zmiany są łatwo wprowadzane.
3. **Zarządzanie projektem** – proces planowania, kontrolowania i realizacji projektu oprogramowania, z uwzględnieniem zarządzania zasobami, czasem, zakresem i kosztami.
4. **Architektura oprogramowania** – ogólna struktura systemu, w tym podział na moduły, komponenty oraz sposób, w jaki one ze sobą współpracują.
5. **Kontrola jakości oprogramowania (QA)** – procesy i narzędzia służące do zapewnienia, że oprogramowanie spełnia oczekiwania jakościowe. Obejmuje testowanie jednostkowe, integracyjne, systemowe i akceptacyjne.
6. **Inżynieria wymagań** – proces identyfikacji, analizy, dokumentowania i zarządzania wymaganiami funkcjonalnymi i нефункциональными oprogramowania.
7. **Zarządzanie konfiguracją** – śledzenie zmian w kodzie źródłowym, dokumentacji oraz innych artefaktach projektu. Narzędzia takie jak Git są powszechnie używane w tym celu.

Inżynieria oprogramowania jest interdyscyplinarną dziedziną, która łączy wiedzę z zakresu programowania, zarządzania projektami, testowania oraz zarządzania zespołem.

2. Cel laboratorium

Opis najważniejszych procesów biznesowych występujących w tzw. „świecie rzeczywistym”, które należy zautomatyzować za pomocą projektowanej aplikacji oraz specyfikacja wymagań funkcjonalnych i нефункциональных tej aplikacji.

Zadanie do samodzielnej realizacji

Zgodnie z materiałami, załączonymi na końcu niniejszej instrukcji, przygotuj dokumentację wstępną do wybranego zestawu wymagań biznesowych (lista przypadków poniżej). Spróbuj postawić się w pozycji inżyniera, który odpowiada za przygotowanie systemu do analizy danych zgodnie z wymaganiami klienta. W wyniku pracy w obrębie laboratorium powinien powstać dokument, podobny do tego, zamieszczonego w materiałach (co do struktury i rodzaju zawartości).

Poniżej zamieszczono przykładowe tematy do zaprojektowania systemu/aplikacji w ramach zajęć z Inżynierii Oprogramowania. Przykłady te obejmują różnorodne dziedziny i mogą być dostosowane do różnych poziomów zaawansowania projektów, od prostych aplikacji mobilnych po złożone systemy korporacyjne.

Można wybrać jeden z poniższych tematów lub zaproponować swój własny. **UWAGA!!!** Temat projektowanej aplikacji wybiera się jeden raz i realizuje się go przez cały okres zajęć laboratoryjnych. Tematu projektu nie można zmienić.

1. Systemy Zarządzania

- a) System Zarządzania Relacjami z Klientami (CRM) – aplikacja umożliwiająca firmom zarządzanie interakcjami z klientami, śledzenie sprzedaży, analizę danych klientów oraz automatyzację marketingu.
- b) System Zarządzania Zasobami Ludzkimi (HRM) – system wspomagający zarządzanie pracownikami, rekrutację, ocenę wydajności oraz zarządzanie płacami.
- c) System Zarządzania Projektami – narzędzie do planowania, monitorowania i zarządzania projektami, umożliwiające zespołom współpracę i śledzenie postępów.

2. Aplikacje Mobilne

- a) Aplikacja Fitness – aplikacja mobilna śledząca aktywność fizyczną użytkownika, oferująca plany treningowe i monitorująca postępy.
- b) Aplikacja do Zarządzania Finansami Osobistymi – narzędzie pomagające użytkownikom zarządzać budżetem, śledzić wydatki i oszczędności.
- c) Aplikacja do Nauki Języków Obcych – interaktywna aplikacja edukacyjna wspomagająca naukę nowych języków poprzez gry, lekcje i ćwiczenia.

3. E – commerce i Platformy Online

- a) Sklep Internetowy – platforma umożliwiająca sprzedaż produktów online, zarządzanie katalogiem produktów, zamówieniami i płatnościami.
- b) Platforma Marketplace – miejsce, gdzie sprzedawcy mogą oferować swoje produkty lub usługi, a kupujący mogą je przeglądać i kupować.

- c) Aplikacja Subskrypcyjna – usługa oferująca treści lub produkty na zasadzie subskrypcji, np. streaming muzyki, wideo, dostawy jedzenia.

4. Systemy Informacyjne

- a) System Zarządzania Szkołą – oprogramowanie wspomagające zarządzanie szkołą, uczniami, nauczycielami i harmonogramami zajęć.
- b) System Biblioteczny – system umożliwiający zarządzanie zasobami biblioteki, wypożyczeniami i katalogowaniem książek.
- c) System Zarządzania Opieką Zdrowotną – oprogramowanie wspomagające zarządzanie pacjentami, historią medyczną, wizytami i zasobami medycznymi.

5. Aplikacje Społecznościowe i Komunikacyjne

- a) Platforma Społecznościowa – serwis umożliwiający użytkownikom tworzenie profili, dodawanie znajomych, publikowanie postów i interakcję z innymi użytkownikami.
- b) Aplikacja do Komunikacji Zespołowej – narzędzie wspomagające komunikację i współpracę w zespołach, z funkcjami czatu, wideokonferencji i zarządzania zadaniami.
- c) Aplikacja do Udostępniania Zdjęć i Wideo – platforma umożliwiająca użytkownikom przesyłanie, edytowanie i udostępnianie zdjęć oraz filmów.

6. Aplikacje Specjalistyczne

- a) System Zarządzania Produkcją (Manufacturing Execution System, MES) – system wspomagający zarządzanie procesami produkcyjnymi, kontrolę jakości i śledzenie produkcji w czasie rzeczywistym.
- b) Aplikacja do Analizy Danych (Business Intelligence) – narzędzie umożliwiające analizę dużych zbiorów danych, tworzenie wizualizacji i raportów wspomagających podejmowanie decyzji biznesowych.
- c) System Zarządzania Energią (Energy Management System) – system monitorujący i optymalizujący zużycie energii w przedsiębiorstwach lub budynkach.

7. Aplikacje Edukacyjne

- a) Platforma E – learningowa – system umożliwiający tworzenie i udostępnianie kursów online, śledzenie postępów uczniów oraz interakcję z instruktorami.
- b) Aplikacja do Nauki Programowania – interaktywna aplikacja ucząca programowania poprzez zadania, projekty i natychmiastowe sprawdzanie kodu.
- c) Aplikacja do Nauki Matematyki – aplikacja edukacyjna wspomagająca naukę matematyki poprzez interaktywne lekcje, ćwiczenia i gry edukacyjne.

8. Aplikacje Rozrywkowe

- a) Gra Mobilna – interaktywna gra na urządzenia mobilne, może być w różnych gatunkach, takich jak puzzle, strategia, RPG.
- b) Aplikacja Streamingowa dla Gier (Game Streaming) – platforma umożliwiająca strumieniowanie gier na żywo, interakcję z widzami i zarządzanie treściami.

- c) Aplikacja do Tworzenia i Udostępniania Memów – narzędzie umożliwiające tworzenie, edytowanie i udostępnianie memów oraz śledzenie popularności.

9. Aplikacje dla IoT (Internet of Things)

- a) System Inteligentnego Domu – Aplikacja do zarządzania urządzeniami inteligentnego domu, takimi jak oświetlenie, termostaty, kamery bezpieczeństwa.
- b) Aplikacja do Monitorowania Środowiska – system monitorujący parametry środowiskowe, takie jak temperatura, wilgotność, jakość powietrza, i prezentujący dane użytkownikom.
- c) Aplikacja do Zarządzania Flotą Pojazdów – system śledzący lokalizację pojazdów, monitorujący ich stan techniczny oraz optymalizujący trasy.

10. Aplikacje Finansowe

- a) Aplikacja do Inwestowania – platforma umożliwiająca użytkownikom inwestowanie w akcje, obligacje, kryptowaluty i inne instrumenty finansowe.
- b) Aplikacja do Kredytów i Pożyczek – system zarządzający procesem ubiegania się o kredyty, ich zatwierdzaniem oraz monitorowaniem spłat.
- c) Aplikacja do Zarządzania Budżetem Firmowym – narzędzie wspomagające firmy w zarządzaniu finansami, śledzeniu wydatków i planowaniu budżetu.

Powyższe przykłady pokazują szeroki zakres możliwych projektów w ramach inżynierii oprogramowania. Wybór odpowiedniego tematu zależy od wielu czynników, takich jak zainteresowania zespołu, dostępność zasobów, cel projektu oraz wymagania użytkowników końcowych. Każdy z wymienionych systemów może być rozwijany z zastosowaniem różnych technologii, metodologii i najlepszych praktyk inżynierii oprogramowania, co pozwala na stworzenie wartościowego i funkcjonalnego rozwiązania.

3. Materiały pomocnicze

Wykonanie opisu biznesowego „świata rzeczywistego” oraz definicja wymagań funkcjonalnych i нефункциональных to kluczowe kroki w procesie tworzenia oprogramowania, szczególnie na etapie analizy wymagań. Są to działania mające na celu zrozumienie kontekstu biznesowego i oczekiwań wobec systemu, który będzie tworzony. Przyjrzyjmy się bliżej każdemu z tych elementów.

1. Opis biznesowy „świata rzeczywistego”

To szczegółowa analiza otoczenia, w którym będzie funkcjonować projektowane oprogramowanie, uwzględniająca procesy biznesowe, interesariuszy i cele organizacyjne. Jego celem jest zrozumienie, jak oprogramowanie wpisuje się w działalność przedsiębiorstwa i jakie problemy ma rozwiązywać.

Kluczowe kroki w opisie biznesowym:

1. Zrozumienie procesów biznesowych:

- Przeanalizowanie istniejących procesów i ich kluczowych elementów (np. przepływu informacji, decyzyjności, wykorzystywanych narzędzi).
- Zidentyfikowanie obszarów, które mogą zostać usprawnione przez nowe oprogramowanie (np. automatyzacja, integracja różnych systemów, poprawa wydajności).

2. Zidentyfikowanie interesariuszy:

- Ustalenie, kto będzie korzystał z systemu (np. pracownicy, klienci, partnerzy biznesowi) i jak system wpłynie na ich pracę.
- Ustalenie celów różnych grup interesariuszy – co chcą osiągnąć dzięki oprogramowaniu.

3. Zdefiniowanie problemów i potrzeb:

- Jakie są obecne wyzwania w świecie rzeczywistym, które oprogramowanie ma rozwiązać? Mogą to być problemy takie jak czasochłonność pewnych procesów, trudności w komunikacji, brak dostępu do kluczowych informacji w czasie rzeczywistym.

4. Cele biznesowe:

- Co organizacja chce osiągnąć dzięki wdrożeniu oprogramowania? Cele mogą obejmować zwiększenie wydajności, redukcję kosztów, poprawę jakości usług, wzrost konkurencyjności.

2. Definicja wymagań funkcjonalnych

Wymagania funkcjonalne opisują konkretne funkcje i zadania, jakie oprogramowanie ma realizować. Odpowiadają na pytanie „co” system ma robić. Często stosowany sposób opisu wymagań to język naturalny. Liczba wymagań funkcjonalnych może być bardzo duża; konieczne jest pewnego rodzaju uporządkowanie tych wymagań, które ułatwi pracę nad nimi (złożoność!).

Wymagania funkcjonalne:

- opisują, jak funkcja powinna działać,
- skupiają się na wyniku działania użytkownika,
- definiują wymagania użytkownika,
- weryfikują funkcjonalność system

Przykłady wymagań funkcjonalnych:

- Rejestracja i logowanie użytkowników: System musi umożliwiać użytkownikom tworzenie kont i logowanie się.
- Przetwarzanie transakcji: System ma umożliwiać użytkownikom dokonywanie zakupów online.
- Generowanie raportów: System ma generować raporty o sprzedaży na podstawie danych z bazy.
- Wyszukiwanie i filtrowanie danych: Użytkownicy mają możliwość wyszukiwania produktów w systemie z użyciem różnych filtrów.
- Powiadomienia: System ma automatycznie wysyłać powiadomienia o ważnych wydarzeniach (np. o statusie zamówienia).

Wymagania funkcjonalne zazwyczaj są przedstawiane w formie scenariuszy lub przypadków użycia (ang. use cases), które opisują interakcję użytkownika z systemem.

3. Definicja wymagań niefunkcjonalnych

Wymagania niefunkcjonalne odnoszą się do tego, **jak** system ma działać. Obejmują one kryteria jakościowe, które nie odnoszą się bezpośrednio do konkretnych funkcji, ale są równie ważne dla zapewnienia odpowiedniej jakości oprogramowania.

Przykłady wymagań niefunkcjonalnych:

- Wydajność: System powinien obsługiwać minimum 1000 użytkowników jednocześnie bez spadku wydajności.
- Bezpieczeństwo: Dane użytkowników muszą być przechowywane w sposób bezpieczny, z zastosowaniem szyfrowania i ochrony przed nieautoryzowanym dostępem.
- Skalowalność: System musi być zaprojektowany tak, aby można go było łatwo rozbudować w przyszłości, np. poprzez dodanie nowych modułów.
- Dostępność: System musi być dostępny 24/7, z maksymalnym czasem przestoju nieprzekraczającym 1 godziny miesięcznie.
- Łatwość użycia: Interfejs użytkownika musi być intuicyjny i prosty w obsłudze dla osób bez specjalistycznej wiedzy technicznej.
- Przenośność: System musi być kompatybilny z różnymi platformami (np. Windows, Linux, macOS).

Rola dokumentacji wymagań

Dokumentowanie wymagań funkcjonalnych i niefunkcjonalnych jest kluczowym elementem procesu tworzenia oprogramowania. Umożliwia ono precyzyjne przekazanie oczekiwań wszystkich interesariuszy zespołowi programistów i testerów. Dobrze przygotowane wymagania:

- Minimalizują ryzyko nieporozumień między zespołem a klientem.
- Pomagają w ocenie postępu prac.
- Służą jako punkt odniesienia przy testowaniu oprogramowania.

Typowe formaty dokumentacji to:

- Dokument Specyfikacji Wymagań (ang. Software Requirements Specification, SRS): Zawiera szczegółowy opis wymagań funkcjonalnych i niefunkcjonalnych.
- Modelowanie przypadków użycia (ang. Use Case Diagrams): Wykorzystuje diagramy UML do przedstawienia interakcji użytkownika z systemem.

Inżynieria Oprogramowania – Laboratorium 1

Podsumowując, opis biznesowy „świata rzeczywistego” pozwala na głębokie zrozumienie kontekstu, w którym będzie działało oprogramowanie, natomiast definicja wymagań funkcjonalnych i нефункциональных precyzuje, co oprogramowanie ma robić i jakie kryteria jakościowe ma spełniać.

3.1. Przykłady opisu świata rzeczywistego wraz z definicją wymagań funkcjonalnych i niefunkcjonalnych

Poniżej przedstawiono przykłady pokazujące, jak istotne jest zrozumienie rzeczywistych potrzeb biznesowych, a także jak ważne jest precyzyjne zdefiniowanie zarówno wymagań funkcjonalnych, jak i niefunkcjonalnych. Dzięki takim wymaganiom zespół projektowy jest w stanie stworzyć oprogramowanie, które nie tylko spełnia swoje funkcje, ale również odpowiada na kluczowe wyzwania w zakresie wydajności, bezpieczeństwa, łatwości obsługi i skalowalności.

Przykład 1. Poniżej przedstawiono dokument, który opisuje kluczowe aspekty projektowanego systemu, umożliwiając jego precyzyjne wdrożenie zgodnie z założeniami biznesowymi i technicznymi.

1. Opis świata rzeczywistego

Firma XYZ to przedsiębiorstwo zajmujące się sprzedażą detaliczną i hurtową produktów elektronicznych. Obecnie firma posiada sieć sklepów stacjonarnych, a jej pracownicy zarządzają zamówieniami i stanami magazynowymi ręcznie, co powoduje liczne opóźnienia i błędy w dostawach. Klienci często narzekają na brak informacji o dostępności produktów w czasie rzeczywistym, a pracownicy muszą dzwonić do różnych magazynów, aby sprawdzić stany magazynowe.

Firma XYZ zdecydowała się na wdrożenie systemu zarządzania zamówieniami i stanami magazynowymi, który umożliwi automatyzację procesów, poprawi jakość obsługi klienta oraz zapewni dostęp do aktualnych informacji o produktach. System ma działać w modelu online, z dostępem przez przeglądarkę internetową, co umożliwi łatwe korzystanie zarówno przez pracowników, jak i klientów.

Celem wdrożenia nowego systemu jest:

- Zautomatyzowanie zarządzania zamówieniami i stanami magazynowymi.
- Zwiększenie szybkości obsługi klientów.
- Usprawnienie komunikacji między sklepami a magazynami.
- Zwiększenie sprzedaży poprzez lepszą dostępność produktów.

Wymagania funkcjonalne

1. Rejestracja użytkowników:

- System musi umożliwiać rejestrację pracowników firmy (adminów, menedżerów magazynów, sprzedawców) oraz klientów detalicznych.
- Rejestracja musi wymagać podania podstawowych danych (imię, nazwisko, adres e – mail, numer telefonu, hasło).

2. Logowanie i zarządzanie kontem:

- Użytkownicy muszą mieć możliwość logowania się do systemu przy użyciu zarejestrowanego adresu e – mail i hasła.
- System powinien umożliwiać użytkownikom zmianę danych konta (np. zmiana hasła, aktualizacja danych osobowych).

3. Zarządzanie zamówieniami:

- System musi umożliwiać tworzenie, edytowanie i usuwanie zamówień przez sprzedawców oraz klientów.

Inżynieria Oprogramowania – Laboratorium 1

- System musi wyświetlać szczegóły zamówienia, takie jak liczba produktów, cena, termin dostawy.
 - Klienci powinni móc monitorować status swojego zamówienia (np. "W trakcie realizacji", "Zrealizowane").
4. Aktualizacja stanów magazynowych:
- System musi umożliwiać pracownikom magazynu aktualizację stanów magazynowych w czasie rzeczywistym.
 - System powinien informować o brakach w magazynie oraz umożliwiać generowanie zamówień na dostawę do magazynów.
5. Wyszukiwanie i filtrowanie produktów:
- System musi umożliwiać wyszukiwanie produktów według różnych kryteriów (np. nazwa, kategoria, dostępność).
 - Użytkownicy powinni mieć możliwość filtrowania produktów według ceny, dostępności w magazynie, marki itp.
6. Powiadomienia:
- System powinien automatycznie wysyłać powiadomienia e – mail do klientów o zmianie statusu zamówienia.
 - System musi wysyłać przypomnienia o niskich stanach magazynowych dla wybranych produktów.
7. Raportowanie:
- System musi umożliwiać generowanie raportów dotyczących sprzedaży (dziennie, tygodniowe, miesięczne), stanu magazynów i liczby zamówień.
 - Menedżerowie muszą mieć możliwość eksportowania raportów do plików CSV lub PDF.

Wymagania niefunkcjonalne

1. Wydajność:
- System musi obsługiwać jednoczesne logowanie co najmniej 1000 użytkowników bez zauważalnych opóźnień w działaniu.
 - Operacje na dużych bazach danych, np. wyszukiwanie produktów, muszą być wykonywane w czasie nie dłuższym niż 3 sekundy.
2. Bezpieczeństwo:
- Dane użytkowników (np. dane osobowe, dane logowania) muszą być szyfrowane w bazie danych oraz podczas transmisji (SSL/TLS).
 - System musi obsługiwać dwuskładnikowe uwierzytelnianie (2FA) dla pracowników i adminów systemu.
3. Skalowalność:

- System musi być skalowalny, aby w przyszłości obsłużyć wzrost liczby użytkowników i zamówień o 200% bez potrzeby przebudowy architektury.

4. Dostępność:

- System musi być dostępny 24/7, z dopuszczalnym czasem przestoju nieprzekraczającym 1 godziny miesięcznie (99,9% dostępności).
- System powinien mieć wbudowane mechanizmy automatycznego tworzenia kopii zapasowych co 24 godziny.

5. Łatwość użycia:

- Interfejs systemu musi być intuicyjny i łatwy w obsłudze, umożliwiający użytkownikom bez specjalistycznej wiedzy technicznej szybkie nauczanie się obsługi.
- System powinien zapewniać pomoc online (np. poprzez sekcję FAQ i samouczki).

6. Kompatybilność:

- System musi być dostępny z poziomu popularnych przeglądarek internetowych (Chrome, Firefox, Safari, Edge) oraz kompatybilny z urządzeniami mobilnymi (responsywny design).

7. Przenośność:

- System musi działać na różnych platformach serwerowych (Windows Server, Linux) i umożliwiać łatwe przeniesienie danych na inne systemy, jeśli zajdzie taka potrzeba.

Przykład 2. Poniżej przedstawiono dokument, który opisuje kluczowe aspekty projektowanego systemu, umożliwiając jego precyzyjne wdrożenie zgodnie z założeniami biznesowymi i technicznymi.

Opis świata rzeczywistego oraz dokumentacja specyfikacji wymagań dla Systemu Zarządzania Zamówieniami.

Celem niniejszego dokumentu jest zdefiniowanie wymagań dla nowego Systemu Zarządzania Zamówieniami, który zostanie wdrożony w firmie XYZ. Dokument ten określa wymagania funkcjonalne i нефункционалне, jakie system musi spełniać, aby skutecznie wspierać procesy biznesowe firmy. Zakres dokumentu obejmuje opis procesów biznesowych, które system ma wspierać, wymagania funkcjonalne, które system musi spełniać, oraz wymagania нефункционалне, takie jak bezpieczeństwo, wydajność i dostępność. Dokument jest przeznaczony dla zespołów: programistycznych, testerów, menedżerów projektu, kluczowych interesariuszy (klientów wewnętrznych). Założenia i ograniczenia systemu związane z działaniem na platformach Windows i Linux. Zakłada się, że integracja z istniejącymi systemami ERP firmy zostanie zapewniona, natomiast pełna migracja danych nie jest przedmiotem tego projektu.

1. Opis świata rzeczywistego (kontekst biznesowy)

1.1. Opis firmy i kontekst operacyjny

Firma XYZ jest średniej wielkości przedsiębiorstwem działającym w branży detalicznej, które zarządza siecią sklepów oraz magazynów na terenie kraju. Obecny system zarządzania zamówieniami jest przestarzały, oparty na ręcznych procesach i nie pozwala na sprawną obsługę klientów online.

1.2. Problemy do rozwiązania

Obecnie firma boryka się z:

- Niewydajnym procesem zarządzania stanami magazynowymi.
- Opóźnieniami w przetwarzaniu zamówień.
- Brakiem dostępu do aktualnych informacji o produktach dla klientów online.
- Problemami komunikacyjnymi między magazynami a punktami sprzedaży.

1.3. Cele biznesowe nowego systemu

- Zautomatyzowanie procesów zarządzania magazynami i zamówieniami.
- Zwiększenie sprzedaży poprzez wdrożenie platformy online.
- Usprawnienie procesu obsługi klienta.
- Umożliwienie pracownikom sklepu oraz klientom dostępu do aktualnych stanów magazynowych w czasie rzeczywistym.

1.4. Kluczowi interesariusze

- Klienci: Oczekują szybkiej obsługi zamówień oraz dostępu do informacji o produktach.
- Pracownicy magazynu: Potrzebują narzędzi do sprawnego zarządzania stanami magazynowymi.
- Menedżerowie sklepów: Odpowiedzialni za kontrolę stanu produktów i analizę sprzedaży.

2. Wymagania funkcjonalne

2.1. Ogólne założenia funkcjonalne

System musi zapewniać możliwość zarządzania zamówieniami i stanami magazynowymi w czasie rzeczywistym, umożliwiać generowanie raportów sprzedaży oraz powiadamianie użytkowników o statusach zamówień.

2.2. Szczegółowe wymagania funkcjonalne

Wymaganie	Opis	Priorytet
Rejestracja użytkowników	System umożliwia rejestrację nowych użytkowników, w tym pracowników i klientów.	Wysoki
Logowanie	Użytkownicy mogą logować się do systemu, korzystając z adresu e – mail i hasła.	Wysoki
Zarządzanie zamówieniami	Użytkownicy mogą tworzyć, edytować i usuwać zamówienia.	Wysoki
Aktualizacja stanów magazynowych	System pozwala na bieżącą aktualizację stanów magazynowych.	Wysoki
Generowanie raportów	Menedżerowie mogą generować raporty sprzedaży i analizę stanów magazynowych.	Średni
Powiadomienia e – mail	Automatyczne wysyłanie powiadomień o zmianie statusu zamówienia.	Średni

3. Wymagania niefunkcjonalne

1. Wydajność:

- System musi obsługiwać do 1000 użytkowników jednocześnie, bez opóźnień większych niż 3 sekundy na operację.

2. Bezpieczeństwo:

- Dane użytkowników muszą być szyfrowane (SSL/TLS).
- System musi obsługiwać dwuskładnikowe uwierzytelnianie (2FA) dla pracowników i adminów.

3. Skalowalność

- System musi być skalowalny, aby w przyszłości obsłużyć wzrost o 200% w liczbie użytkowników.

4. Dostępność

- System musi być dostępny 24/7, z dopuszczalnym przestojem nie większym niż 1 godzina miesięcznie.

5. Użyteczność

- Interfejs użytkownika musi być intuicyjny i dostępny zarówno na komputerach stacjonarnych, jak i urządzeniach mobilnych.

6. Kompatybilność

- System musi działać na przeglądarkach (Chrome, Firefox, Safari, Edge) oraz urządzeniach mobilnych.

7. Przenośność

- System musi być przenoszalny na różne platformy serwerowe (Windows Server, Linux).

4. Priorytetyzacja wymagań

- Krytyczne wymagania: Rejestracja użytkowników, logowanie, zarządzanie zamówieniami, aktualizacja stanów magazynowych.
- Wysoki priorytet: Generowanie raportów, powiadomienia e – mail.
- Średni priorytet: Dodatkowe funkcjonalności analityczne, rozszerzenia systemu.

5. Kryteria akceptacji

- Definicja ukończenia projektu: Projekt uznaje się za ukończony, gdy wszystkie krytyczne wymagania funkcjonalne oraz нефункционалне są spełnione, a system pomyślnie przejdzie testy UAT.
- Testowanie i weryfikacja: Każda funkcjonalność musi być dokładnie przetestowana, w tym testy jednostkowe, integracyjne, wydajnościowe oraz testy bezpieczeństwa.