**1. Założenia projektowe na duży projekt w Javie:**

Zaprojektowany zostanie system zarządzający biblioteką książek. System będzie mieć funkcje CRUD (Create, Read, Update i Delete) dla zarządzania zasobami bibliotecznymi. Środowisko programistyczne to IntelliJ.

Założenia projektowe: Model Danych – Książka, Autor, Wydawnictwo, Gatunek. Każda z tych encji będzie reprezentowana jako tabela w mySQL. Interfejs klienta także zostanie napisany w Javie – dodawanie, przeglądanie, aktualizowanie i usuwanie informacji o książkach, wydawnictwach i gatunkach.

REST API – operacje CRUD będą udostępniane przez REST API wykorzystywane przez frontend do komunikacji z backendem.

Frontend: Aplikacja kliencka napisana w Javie komunikuje się z backendem dzięki REST API.

Backend: Aplikacja Spring Boot wraz z Hibernate zarządza operacjami na bazie danych. Serwer aplikacji napisany w Javie z wykorzystaniem Spring Boota. Hibernate będzie służyć do zarządzania encjami takimi, jak: Książka, Autor, Wydawnictwo i Gatunek. Hibernate umożliwia operacje CRUD na tych encjach, zarządzając transakcjami i sesjami z bazą danych.

Docker: Konteneryzacja aplikacji i bazy danych za pomocą Docker.

Unit Testing: Testowanie jednostkowe – Junit i Mockito.

Wymagania i zdefiniowanie celów projektu:

- Autentykacja i autoryzacja są niezbędnymi elementami systemu bibliotecznego i powinny zostać zaimplementowane priorytetowo. Podobnie, możliwość wyszukiwania książek według różnych kryteriów jest kluczowa dla komfortu użytkownika, więc powinna zostać uwzględniona w projekcie.

zwiększenie dostępności zasobów bibliotecznych, poprawa obsługi użytkowników, automatyzacja procesów zarządzania książkami.

- Zakres: Funkcjonalności CRUD dla książek, autorów i gatunków, zarządzanie wypożyczeniami, rejestracją użytkowników i wyszukiwaniem.

- Ograniczenie: Dostępność danych o książkach i autorach.

- Wymagania biznesowe:

- System dostępny 24/7.

- Bezpieczeństwo danych użytkownika, ochrona prywatności.

- Przyjazny dla użytkownika interfejs.

Role użytkowników:

- Administrator: zarządza użytkownikami, książkami, wypożyczeniami, rejestracja użytkowników.

- Pracownik biblioteki: obsługa książek i wypożyczeń.

- Czytelnik: rejestruje się, szuka książek i wypożycza je.

Testy jednostkowe: Sprawdzenie prawidłowości zapisu, czytania, aktualizacji i usuwania książek, autorów, wydawnictw i gatunków. Testowanie logiki wypożyczenia i zwrotu książek. Testowanie poprawności obsługi błędów i wyjątków.

**2. Założenia projektowe na średni projekt w Androidzie:**

- Jako środowisko programistyczne zostanie wykorzystane Android Studio oraz framework Jetpack Compose umożliwiający tworzenie interfejsów użytkownika. Dodatkowo zastosowana będzie biblioteka Material Components for Android (biblioteka UI Material Design) oraz narzędzie Android Gradle Plugin – wtyczka Gradle automatyzująca proces kompilacji oraz tworzenia pakietów APK.

- Android Emulator – emulator Androida do testowania aplikacji na różnych urządzeniach.

Rozwinięcia aplikacji:

- wyświetlanie aktualnej pogody, możliwość odświeżania danych pogodowych ręcznie lub automatycznie

- określenie okresu prognozy (3 dni, 7 dni, 14 dni)

- możliwość wyświetlania prognozy dla różnych parametrów (temperatura, opady, ciśnienie)

- wyszukiwanie pogody:

- określenie sposobu szukania lokalizacji ( wpisanie nazwy lub wybór z mapy)

**3. Założenia projektowe na mały projekt w iOS Xcode:**

Środowisko programistyczne:

- Xcode 12

- Swift

Funkcje aplikacji:

- pobieranie danych o wiadomościach z serwera w formacie JSON

- parsowanie danych JSON i wyświetlanie ich na liście

- możliwość wyszukiwania wiadomości

Architektura aplikacji:

- Model – View – Controller (MVC)

- Zostanie wykorzystany JSON, gdyż jest popularnym narzędziem do wymiany danych w systemach iOS oraz nadaje się do przesyłania i odbierania danych z serwera.

**Poprawki dla 1. punktu:**

a) Historia wypożyczeń będzie dostępna zarówno stacjonarnie, jak i zdalnie.

Zdalnie: Użytkownicy mogą zalogować się do swojego konta w systemie bibliotecznym za pomocą przeglądarki internetowej. Po zalogowaniu się użytkownik powinien dodatkowo podać swoje dane identyfikacyjne ( imię, nazwisko oraz unikalny token, który zostanie wygenerowany dla niego podczas rejestracji). Po podaniu tychże informacji czytelnik będzie mieć dostęp do swojej historii wypożyczeń (należy wybrać ją z menu) zawierającej:

- Tytuł książki

- Autor

- Data wypożyczenia

- Data zwrotu

- Status wypożyczenia (wypożyczona, przetrzymana).

Stacjonarnie: Kiosk samoobsługowy (książkomat) – Biblioteka może zainstalować kiosk samoobsługowy umożliwiający samodzielne sprawdzanie historii wypożyczeń. Użytkownicy mogą logować się do kiosku za pomocą tych samych danych identyfikacyjnych, co do przeglądarki internetowej (imię, nazwisko i token). Zaletą książkomatu jest to, że użytkownicy mogą wypożyczać i zwracać książki bez konieczności kontaktowania się z pracownikiem biblioteki.

b) Druga ważna rzecz to powiadomienia o terminach zwrotu książek. Optymalnym rozwiązaniem jest połączenie kilku sposobów wysyłania powiadomień w celu zapewnienia użytkownikom elastyczności oraz większych szans na dotarcie do nich z informacją o terminie zwrotu. E-mail: System wysyła wiadomość e-mail na 3 dni przed terminem zwrotu, SMS: system wysyła SMS na 1 dzień przed terminem zwrotu, Powiadomienie systemowe: System wyświetla powiadomienie o zbliżającym się terminie zwrotu po zalogowaniu się użytkownika do konta.

c) System biblioteczny jest zintegrowany z systemem płatności online, aby automatycznie przypominać o płatnościach za przetrzymane książki. Użytkownicy będą mogli opłacić kary online za pomocą karty płatniczej lub przelewu bankowego. Za każdy dzień przetrzymania książki naliczana jest kara. W przypadku notorycznego przetrzymywania książek biblioteka blokuje użytkownikowi możliwość wypożyczania książek, dopóki nie opłaci zaległych kar. Użytkownicy mogą sprawdzić aktualny stan swoich kar w systemie bibliotecznym. System blokuje dostęp do wypożyczania książek użytkownikom z zaległymi karami. Kara za dzień przetrzymania wynosi 2 zł. Przykład: Użytkownik przetrzymuje książkę przez 10 dni. Wysokość kary za dzień przetrzymania to 2 zł. Łączna kwota kary to 20 zł. Użytkownik otrzymuje przypomnienie o karze na adres e-mail. Jest tam link do strony internetowej systemu płatności online. Wybiera płatność kartą i podaje dane do płatności. Po dokonaniu płatności, użytkownik otrzymuje potwierdzenie na adres e-mail. System odblokuje mu wtedy możliwość wypożyczania książek.

d) Rezerwacja książki w systemie bibliotecznym: System biblioteczny pozwala użytkownikom na rezerwację książek aktualnie wypożyczonych przez innych użytkowników lub niedostępnych z innych przyczyn.

Proces rezerwacji:

1. Wyszukiwanie książki: Użytkownik wyszukuje książkę, którą chce zarezerwować w katalogu bibliotecznym. Katalog zawiera różne kryteria wyszukiwania, takie jak: autor, tytuł, gatunek itp.

2. Dostępność: Po znalezieniu książki można sprawdzić jej dostępność. System wyświetli informację, czy książka jest dostępna, wypożyczona, czy zarezerwowana.

3. Złożenie rezerwacji: Gdy książka jest wypożyczona lub niedostępna, czytelnik może dokonać rezerwacji. System prosi go o podanie danych osobowych: imię i nazwisko oraz token.

4. Kolejka rezerwacji: Rezerwacje realizowane są w kolejności chronologicznej. Użytkownik może sprawdzić swoją pozycję w kolejce rezerwacji.

5. Powiadomienie: Gdy książka będzie dostępna, system powiadomi użytkownika mailowo lub opcjonalnie SMS-em. Czytelnik ma 7 dni na odbiór książki.

6. Odbiór książki: Użytkownik może odebrać zarezerwowaną książkę w bibliotece. System wyświetli informację o jego rezerwacji.

7. Anulowanie rezerwacji: Użytkownik może anulować rezerwację w dowolnym momencie.

- Dodatkowe funkcje:

1. Automatyczne anulowanie rezerwacji: System może automatycznie anulować rezerwację, jeśli użytkownik nie odbierze książki w określonym czasie.

2. Powiadomienia o zbliżającym się terminie odbioru: System może wysyłać użytkownikom powiadomienia o zbliżającym się terminie odbioru zarezerwowanej książki.

Stacjonarna rezerwacja książki: Czytelnik może zarezerwować książkę stacjonarnie w bibliotece, korzystając z kiosku samoobsługowego (książkomatu).

Proces rezerwacji:

1. Użytkownik loguje się do kiosku samoobsługowego za pomocą imienia, nazwiska i tokenu.

2. Wybiera opcję „Rezerwuj książkę”.

3. Wyszukuje książkę, którą chce zarezerwować, korzystając z ekranu dotykowego.

4. Sprawdza dostępność książki.

5. Zatwierdza rezerwację.

Odbiór książki – Użytkownik odbiera zarezerwowaną książkę w bibliotece: Użytkownik wybiera opcję „Odbierz książkę” w książkomacie, podaje kod rezerwacji ( w momencie składania rezerwacji system generuje unikalny kod rezerwacji, kod ten jest niezbędny do odbioru zarezerwowanej książki) i odbiera książkę.

- Anulowanie rezerwacji: Użytkownik może anulować rezerwację stacjonarnie w bibliotece, korzystając z kiosku samoobsługowego.

- Kurier włączony jest również w dostawę książek do użytkowników – Dostarcza książki do użytkowników, którzy złożyli zamówienie online lub zarezerwowali książkę. Czytelnik może wybrać opcję dostawy kurierem podczas składania zamówienia lub rezerwacji.

System będzie mieć funkcje CRUD (Create, Read, Update i Delete) dla zarządzania zasobami bibliotecznymi. Środowisko programistyczne to IntelliJ.

Założenia projektowe: Model Danych – Książka, Autor, Wydawnictwo, Gatunek. Każda z tych encji będzie reprezentowana jako tabela w mySQL. Interfejs klienta także zostanie napisany w Javie – dodawanie, przeglądanie, aktualizowanie i usuwanie informacji o książkach, wydawnictwach i gatunkach.

REST API – operacje CRUD będą udostępniane przez REST API wykorzystywane przez frontend do komunikacji z backendem.

Frontend: Aplikacja kliencka napisana w Javie komunikuje się z backendem dzięki REST API.

Backend: Aplikacja Spring Boot wraz z Hibernate zarządza operacjami na bazie danych. Serwer aplikacji napisany w Javie z wykorzystaniem Spring Boota. Hibernate będzie służyć do zarządzania encjami takimi, jak: Książka, Autor, Wydawnictwo i Gatunek. Hibernate umożliwia operacje CRUD na tych encjach, zarządzając transakcjami i sesjami z bazą danych.

Docker: Konteneryzacja aplikacji i bazy danych za pomocą Docker.

Unit Testing: Testowanie jednostkowe – Junit i Mockito.

Wymagania:

- autentykacja i autoryzacja

- możliwość wyszukiwania książek według różnych kryteriów

Cel:

- zwiększenie dostępności zasobów bibliotecznych

- poprawa obsługi użytkowników

- automatyzacja procesów zarządzania książkami.

Zakres:

- dodawanie, usuwanie, edycja książek, autorów i gatunków

- zarządzanie wypożyczeniami

- rejestracja użytkowników

- wyszukiwanie użytkowników

Ograniczenie:

Dostępność danych o książkach i autorach.

Wymagania biznesowe:

- System dostępny 24/7 (95% - przerwa tech. niedziele 3:00 – 3:30)

- Bezpieczeństwo danych użytkownika, ochrona prywatności.

- Przyjazny dla użytkownika interfejs.

Role użytkowników:

- Administrator: (dziedziczy po pracownik biblioteki) + zarządza użytkownikami, książkami, rejestracja użytkowników.

- Pracownik biblioteki: obsługa książek i wypożyczeń.

- Czytelnik: rejestruje się, szuka książek i wypożycza je.

Testy jednostkowe (60% pokrycia kodu)

