

MemoLang - koncepcja wykonania systemu

Wojciech Ganobis, Tomasz Woszczyński



1. Scenariusze przypadków użycia:


1. Logowanie lub rejestracja [nowy lub zarejestrowany użytkownik]
 - 1.1. Użytkownik uruchamia aplikację MemoLang
 - 1.2. Użytkownik ma do wyboru dwie opcje “Zarejestruj się” oraz “Zaloguj się”
 - 1.2.1. Po wybraniu “Zarejestruj się” otwiera się formularz, użytkownik musi uzupełnić następujące dane:
 - Login użytkownika (*maks. 20 znaków alfanumerycznych*)
 - Adres email (*maks. 70 znaków alfanumerycznych i specjalnych*)
 - Hasło (*maks. 50 znaków alfanumerycznych i specjalnych*)
 - Imię i nazwisko (*maks. 60 znaków alfanumerycznych*)
 - Data urodzenia (*do wyboru z pól DD-MM-RRRR*)Poniższe pola są opcjonalne i nowy użytkownik nie musi ich wypełniać:
 - Kraj zamieszkania (*maks. 30 znaków alfanumerycznych*)
 - Szkoła (*maks. 100 znaków alfanumerycznych*)Użytkownik wybiera pole “Zarejestruj”, sprawdzana jest poprawność wprowadzonych danych, jeżeli są poprawne, użytkownik jest dodawany do bazy danych i jest automatycznie zalogowany do aplikacji
 - 1.2.2. Użytkownik wybiera opcję “Zaloguj się”, po czym wypełnia pola:
 - Login lub adres email
 - HasłoPo podaniu prawidłowych danych użytkownik przechodzi do aplikacji MemoLang i może korzystać z jej funkcjonalności.
2. Dodanie nowego słowa [zalogowany użytkownik]
 - 2.1. Zalogowany użytkownik chce dodać nowe słówko do swoich słówek
 - 2.2. Użytkownik uruchamia aplikację
 - 2.3. Użytkownik wybiera opcję dodania słówka
 - 2.4. Aplikacja wyświetla okno dialogowe w którym widoczne są:


- 2.4.1. Pole do wpisania słówka w aktualnym języku (max 100 znaków alfanumerycznych)
 - 2.4.2. Wybór języka (domyślnie najczęściej używany język)
 - 2.4.3. Pole do wpisania tłumaczenia (1-100 znaków alfanumerycznych)
 - 2.4.4. Wybór grupy do której ma zostać dodane słówko z możliwością stworzenia nowej grupy (domyślnie brak grupy)
 - W przypadku utworzenia nowej grupy pojawia się pole do wpisania nazwy grupy (1-100 znaków alfanumerycznych)
 - 2.4.5. Przycisk “akceptuj”
 - Przycisk będzie szary i niemożliwy do kliknięcia jeśli wszystkie warunki nie zostaną spełnione
 - 2.5. Wyskakuje nowe okienko dialogowe z informacją, że słowo zostało dodane oraz z następującymi przyciskami:
 - 2.5.1. “OK”, który po wciśnięciu przeniesie nas do menu głównego
 - 2.5.2. “Dodaj następne słówko”, który po wciśnięciu uruchamia ponownie proces dodawania nowego słówka
3. Pobieranie gotowych zbiorów [zalogowany użytkownik]
 - 3.1. Użytkownik uruchamia aplikację MemoLang
 - 3.2. Użytkownik wybiera pole “Nauka” w menu głównym
 - 3.3. Użytkownik wybiera jeden z istniejących zbiorów
 - 3.4. Użytkownik klika “Dodaj do konta”
 - 3.5. Dodany zbiór jest przypisywany do konta użytkownika oraz tworzona jest jego lokalna kopia, aby użytkownik mógł przeglądać go będąc offline
4. Nauka słów [zalogowany użytkownik]
 - 4.1. Użytkownik uruchamia aplikację MemoLang
 - 4.2. Użytkownik wybiera pole “Nauka” w menu głównym
 - 4.3. Użytkownik wybiera jeden z istniejących zbiorów
 - 4.4. Użytkownik zostaje przekierowany do listy słów należących do danego zbioru. Po nauczaniu się słowa, użytkownik zaznacza to słowo kliknięciem, dzięki czemu słowo zostaje zapamiętane (implementowana jest metoda nauki *spaced repetition*), wraz z licznikiem etapów
5. Testowanie znajomości słów [zalogowany użytkownik]
 - 5.1. Użytkownik uruchamia aplikację MemoLang
 - 5.2. Użytkownik wybiera pole “Sprawdź się” w menu głównym


- 5.3. Użytkownik wybiera jeden ze zbiorów
- 5.4. Losowane jest 80% wszystkich słów ze zbioru, test wiedzy polega na zaznaczeniu prawidłowego tłumaczenia z czterech wypisanych słów
- 5.5. Na końcu testu wypisywany jest wynik, na podstawie którego użytkownik zdobywa punkty w rankingu ogólnym
6. Sprawdzenie pożądanego rankingu (na podstawie testów) [zalogowany użytkownik]
 - 6.1. Użytkownik uruchamia aplikację
 - 6.2. Użytkownik wybiera opcję "Ranking"
 - 6.3. Uruchamia się ranking globalny wszystkich użytkowników z ostatniego tygodnia
 - 6.3.1. Użytkownik ma możliwość zmiany "wszyscy użytkownicy" na:
 - Wszyscy użytkownicy
 - Użytkownicy z danego kraju
 - Znajomi
 - Aktualnie wpisana szkoła
 - 6.3.2. Użytkownik ma również możliwość zmiany "ostatni tydzień" na:
 - Ostatni tydzień
 - Ostatni miesiąc
 - Ostatni rok
 - Cały czas
 - 6.4. Po wyborze wyświetla się aktualny ranking z kontem użytkownika


2. Projekty ekranów

adam2137


Nauka


Sprawdź się


Słówka


Ranking

Nauka

☐ Wszystkie zbiory☒ Zapisane zbiory

Obsługa procesów

Struktury danych


Owoce i warzywa

Piłka nożna

Grafika komputerowa

Mechanika kwantowa

Fizyka niskich temperatur

Obsługa procesów
Zapisz zbiór

processproces

jobzadanie

to inheritodziedziczyć

to terminatetakończyć, przerwać

to deferodroczyć, wstrzymać

parent (process)proces rodzica

child (process)proces dziecka

signal handlingobsługa sygnałów

Test: Obsługa procesów

to terminate

☐ zakończyć☐ odziedziczyć

☐ zadanie☐ odroczyć

signal handling

☐ identyfikator☐ proces dziecka

☐ instancja☐ obsługa sygnałów

process state

☐ stan procesu☐ uruchomiony proces

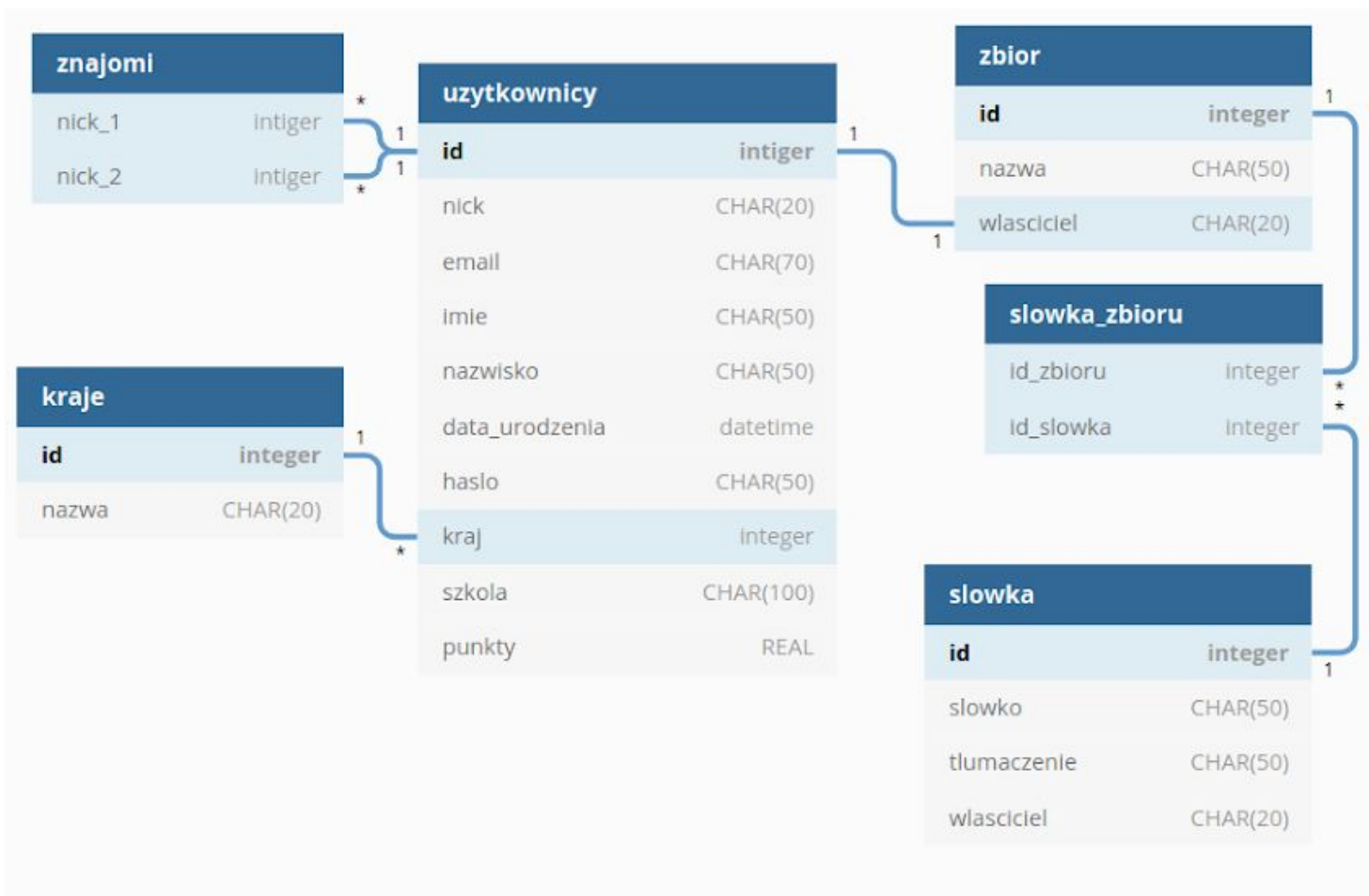
☐ stan zadania☐ czas wykonywania procesu



3. Projekt architektury

Aplikacja będzie napisana w języku C# przy użyciu środowiska Xamarin oraz frameworku .NET. Narzędzia programistyczne wykorzystywane do pracy to Visual Studio 2019 oraz zewnętrzne aplikacje testowe. Za frontend oraz wyświetlanie danych na ekranie urządzeń odpowiedzialny będzie Xamarin.Forms, który umożliwia tworzenie wspólnego UI dla sprzętu działającego na systemach Android oraz iOS. Backend aplikacji MemoLang zostanie napisany w języku C#.

Baza danych będzie wykorzystywała relacyjny server PostgreSQL w wersji 12.3. Programista aplikacji będzie mógł pisać wszelkie zapytania do bazy. Pośrednio do zapytań SELECT, INSERT oraz UPDATE będzie miał również użytkownik na przykład poprzez dodanie nowego słówka do zbioru i tym podobne. Poniżej widoczny jest diagram bazy danych



4. Główne zasady kodowania

Zgodnie z ogólnie przyjętymi zasadami programowania, kod, jak i wszystkie komentarze znajdujące się w plikach, muszą być pisane w języku angielskim. W razie potrzeby powiększenia zespołu nie chcemy ograniczać naszych możliwości, dlatego wybór języka angielskiego na język główny programowania będzie optymalny.

Aplikacja MemoLang będzie napisana w języku C#, dlatego przestrzegane będą podstawowe standardy kodowania, jakie obowiązują w tym języku. Nazwy metod oraz klas będą zawsze pisane w Pascal Case (np. `public class WordsSet`), a nazwy argumentów oraz zmiennych lokalnych w Camel Case (np. `date birthDate`). Ponadto preferowane będzie użycie zmiennych predefiniowanych, a nie systemowych (`int` zamiast `Int32`). Nazwę interfejsu powinna poprzedzać litera `I` (np. `IWord`). W przypadku użycia typów jednorazowo, preferowane jest użycie słowa kluczowego `using` (np. `using (var conn = new`

`SqlConnection(connectionString)`). Zmienne globalne muszą zawsze być pisane używając dużych liter.

Xamarin wykorzystuje wzorzec Model-View-ViewModel (MVVM), aby oddzielić logikę prezentacji i biznesową aplikacji od interfejsu użytkownika. Należy starać się nie przypisywać więcej niż jednej strony (np. test, nauka) do jednego widoku modelu. Widoki modeli nie powinny być przekazywane w metodach do innych stron. W plikach `name.xaml.cs` powinno znajdować się tylko zainicjalizowanie konstruktora, stworzenie modelu widoku, przypisanie kontekstu danych do modelu widoku, jak i wywołanie metody `Initialize`. Chcemy, aby nasz kod był zwięzły, dzięki czemu testowanie aplikacji będzie łatwiejsze.

Całość projektu będzie zamieszczona w prywatnym repozytorium w serwisie GitHub, do którego dostęp będą mieli programiści, testerzy oraz zarząd. Wszystkie zmiany powinny być tworzone w osobnych branchach, a następnie po rewizji kodu, mergeowane z gałęzią `master`. Wszystkie commity powinny dotyczyć jak najmniejszej ilości plików - przyspieszy to proces rewizji kodu, jak i ułatwi śledzenie i wprowadzanie zmian. Nazwy commitów powinny być pisane w języku angielskim w czasie teraźniejszym, muszą zawierać zwięzły opis wprowadzanych zmian, np. `"Introduce adding words to database"`.

5. Identyfikacja i zasady zarządzania ryzykiem

Ryzyka jakie mogą wystąpić to:

- Możliwość przeciążenia serwerów spowodowana zbyt dużą ilością użytkowników. Prawdopodobieństwo wystąpienia takiego zdarzenia jest stosunkowo niskie, ponieważ aplikacje mobilne nie cieszą się bardzo dużą ilością użytkowników na początku istnienia. Problem może nastąpić jednak po rozpoczęciu akcji reklamowej, kiedy dużo użytkowników będzie masowo pobierać aplikację i łączyć się z serwerami. Wtedy jednak będziemy się spodziewać takiego zjawiska, dzięki czemu będziemy mogli mu łatwiej zapobiec.
- Niedotrzymanie terminu ukończenia aplikacji. Prawdopodobieństwo tego ryzyka jest średnie/duże. Jeżeli nie będziemy na bieżąco sprawdzać postępów lub źle rozplanujemy czas na wykonywanie danego etapu będziemy musieli się rozprawić z brakującym czasem. Aby uniknąć tego problemu należy monitorować postępy na bieżąco. Jeśli problem wystąpi pracownicy będą musieli zmierzyć się z nadgodzinami lub opóźnieniem w realizacji projektu.

6. Zgodność prac z tablicą koncepcyjną i specyfikacją

Po przeanalizowaniu wykonanych prac i porównaniu ich z wizją z tablicy koncepcyjnej oraz specyfikacji wymagań, możemy stwierdzić, że otrzymane wyniki są zadowalające. Aktualnie nie przewidujemy większych zmian względem tego, co zostało opisane w tablicy koncepcyjnej, ani specyfikacji wymagań.