

Dokumentacja projektu - aplikacja do nauki fiszek

Learnwords

Opis tematu

Zakres projektu

Motywacja

Cel biznesowy

Założenia projektowe

Funkcjonalne

Niefunkcjonalne

Schemat systemu i opis implementacji

Diagram systemu

Opis

Wykorzystane technologie i biblioteki

Instrukcja wdrożeniowa

Instrukcja użytkownika

Podsumowanie i wnioski

Learnwords

Tytuł projektu : Learnwords

Autor : Alicja Przeździecka

Data rozpoczęcia : 16.12.2024

Data zakończenia : 26.01.2025

Uczelnia: Politechnika Warszawska

Prowadzący: dr inż. Janusz Rafałko

Opis tematu

Zakres projektu

Learnwords to aplikacja okienkowa umożliwiająca systematyczną naukę za pomocą fiszek. Użytkownik może dodawać fiszki, uczyć się ich w trybie wpisywania, a także śledzić swoje postępy dzięki statystykom wyświetlanym po każdej sesji nauki oraz w panelu konta. Dodatkowo w celu poprawy

systematyczności nauki użytkownik ma możliwość dodania przypomnienia o następnej sesji nauki do swojego kalendarza Google.

Motywacja

Aplikacja ma na celu ułatwienie systematyczności nauki i wytrwałości podczas nauki np. języków obcych.

Aplikacja przeznaczona jest dla osób uczących się języków obcych, studentów oraz wszystkich, którzy chcą szybko przyswoić nowe informacje.

Cel biznesowy

Celem jest zaprojektowanie aplikacji, która może być rozwijana o dodatkowe funkcje, np. integrację z innymi platformami oraz zapis użytkowników do bazy danych.

Założenia projektowe

Funkcjonalne

- Logowanie i tworzenie konta w celu oddzielenia użytkowników
- Dodawanie fiszek na trzy sposoby: w aplikacji, z pliku TXT, z pliku CSV
- Usuwanie całych zestawów jak i pojedynczych fiszek
- Tryb nauki z informacją zwrotną o poprawnych i błędnych odpowiedziach występujących podczas zakończonej sesji nauki
- Statystyki podsumowujące ogólne opanowanie zestawu oraz czas utworzenia i wielkość zestawu
- Dodawanie przypomnień do kalendarza Google w wybranym terminie

Niefunkcjonalne

- Kompatybilność: Aplikacja wymaga połączenia z Internetem w celu autoryzacji dostępu do konta Google
- Zapis danych: Po każdym wprowadzeniu zmian przez użytkownika dane są zapisywane lokalnie w stworzonym na początku pliku. W przypadku braku pliku tworzony jest nowy plik na podstawie domyślnego pliku zapisanego w projekcie.
- Szybka odpowiedź aplikacji na interakcje

- Możliwość rozszerzania i modyfikowania aplikacji w przyszłości

Schemat systemu i opis implementacji

Diagram systemu



Opis

1. **User, SetOfCards, Flashcard** – klasy definiujące podstawowe obiekty aplikacji, przechowujące dane dotyczące użytkowników, zestawów fiszek i pojedynczych fiszek. Każda fiszka przechowuje dane takie jak słowo, definicja oraz odpowiedzi użytkownika (poprawne/błędne).

2. **GoogleAuth** – klasa odpowiedzialna za logikę obsługi Google Authentication (autoryzacji użytkownika) oraz integrację z kalendarzem Google. Umożliwia autoryzację użytkownika oraz dodawanie wydarzeń do kalendarza powiązanego z kontem Google.
3. **PanelManager** – klasa zarządzająca logiką wyświetlania odpowiednich paneli w zależności od wybranego elementu w menu. Koordynuje zmiany widoków i przekazuje informacje o aktywnym panelu, odpowiedzialna za organizację interfejsu użytkownika.
4. **MyMenuBar** – klasa odpowiedzialna za wygląd i funkcjonalność paska menu aplikacji, który dostosowuje się do stanu użytkownika (np. pokazuje opcje logowania, rejestracji lub dostęp do profilu użytkownika w zależności od tego, czy użytkownik jest zalogowany).
5. **Session i klasy pokrewne** – klasa **Session** pełni rolę pośrednika między interfejsem użytkownika (panelami) a danymi aplikacji. Zarządza sesją aktywnego użytkownika, przechowując podstawowe informacje takie jak nazwa użytkownika oraz dane uwierzytelniające, ale nie angażuje się bezpośrednio w weryfikację poprawności danych ani operacje na nich. **Session** udostępnia interfejs do odczytu i zapisu danych użytkownika, jednak cała logika dotycząca poprawności tych danych jest zlecona innym klasom (w tym **Manager**). **Session** może przechowywać dane dotyczące postępu w nauce, ale sama nie wykonuje szczegółowych operacji weryfikacyjnych.
6. **Manager i klasy pokrewne** – klasa **Manager** pełni bardziej zaawansowaną rolę w systemie. Odpowiada za głębszy dostęp do danych użytkowników i zarządzanie nimi. **Manager** zarządza zapisem i odczytem danych, sprawdzaniem poprawności danych (np. przy wczytywaniu plików, zapisie postępu nauki) oraz wykonuje operacje na bazie danych użytkowników. **Manager** współpracuje z innymi klasami odpowiedzialnymi za dane użytkowników, takimi jak **User** czy **SetOfCards**, by przeprowadzać bardziej złożone operacje, takie jak weryfikacja poprawności danych wejściowych, zapis wyników do plików czy odczyt danych z różnych źródeł (np. CSV, TXT). **Manager** może mieć dostęp do pełnej bazy użytkowników i przechowywać informacje o wszystkich kontach, podczas gdy **Session** skupia się tylko na danych bieżącego, zalogowanego użytkownika.
7. **Panele (LoginPanel, RegisterPanel, LearnPanel, AddSetPanel itp.)** – klasy rozszerzające JPanel, odpowiedzialne za wyświetlanie i zarządzanie

interfejsem użytkownika. Każdy panel jest odpowiedzialny za przedstawienie określonego widoku oraz umożliwienie interakcji użytkownika z aplikacją.

8. **LoadFlashcardsCSV i LoadFlashcardsTXT** – klasy obsługujące import fiszek z plików CSV i TXT. Przetwarzają dane z pliku i dodają je do odpowiednich zestawów fiszek (SetSession), umożliwiając użytkownikowi łatwe załadowanie nowych fiszek z zewnętrznych plików.
 9. **RoundedPanel, CircularProgressbar** – klasy odpowiedzialne za tworzenie niestandardowych komponentów interfejsu użytkownika. RoundedPanel tworzy panele o zaokrąglonych rogach, natomiast CircularProgressbar reprezentuje postęp w nauce w formie kołowego paska postępu.
 10. **UIComponentAdder** – interfejs definiujący metody do dodawania komponentów do interfejsu użytkownika za pomocą GridBagLayout. Umożliwia łatwe dodawanie i rozmieszczanie elementów na panelach zgodnie z siatką, zapewniając estetyczny układ aplikacji.
-

Wykorzystane technologie i biblioteki

Projekt "Learnwords" to aplikacja do nauki słówek, wykorzystująca:

- **Maven** do zarządzania zależnościami.
 - **Google Calendar API** do autoryzacji i dodawania wydarzeń.
 - **JFreeChart** do generowania wykresów.
 - **Swing** do budowy interfejsu użytkownika.
 - **FlatLaf** do zmiany wyglądu interfejsu.
 - **JCalendar** do wyboru daty z kalendarza.
 - **IntelliJ IDEA** jako środowisko programistyczne.
-

Instrukcja wdrożeniowa

1. Pobranie projektu z repozytorium Bitbucket
2. Wymagania wstępne: Java Development Kit w wersji 22, Maven, środowisko programistyczne wspierające Maven i Java
3. Uruchomienie aplikacji:

- a. Otworzenie pobranego projektu
 - b. Uruchomienie klasy MainApp znajdującej się w pakiecie pl.edu.pw.mini
 4. Interfejs jest dostępny dla użytkownika. Podczas logowania nastąpi przekierowanie do konfiguracji dostępu do konta Google.
 5. Aplikacja jest niezatwierdzona przez google więc tylko przypisane maile do trybu testu mają dostęp do autoryzacji
-

Instrukcja użytkownika

1. **Wybór zakładki z paska menu** – wybierz interesującą cię opcję z głównego menu aplikacji.
2. **Panel logowania** – jeśli masz już konto, przejdź do panelu logowania.
3. **Panel tworzenia konta** – jeśli nie masz konta, użyj opcji do stworzenia nowego konta.
4. **Po udanym logowaniu** – po zalogowaniu uzyskasz dostęp do wielu opcji w aplikacji.
5. Zakładka konto zawierająca:
 - a. Wybór zestawu fiszek przypisanego do konta.
 - b. Możliwość usunięcia zestawu lub pojedynczej fiszki.
 - c. Tabelę z wyświetlonymi fiszkami przypisanymi do wybranego zestawu.
 - d. Statystyki – przekierowanie do panelu z wykresami przedstawiającymi postępy w nauce.
6. **Zakładka "Dodawanie fiszek":**
 - a. Trzy opcje dodawania fiszek: poprzez plik TXT, plik CSV lub bezpośrednio w aplikacji.
 - b. Dodane zestawy będą widoczne w zakładce "Konto" po naciśnięciu przycisku "Aktualizuj".
7. Zakładka nauka:
 - a. Użytkownik ładuje dostępne zestawy fiszek, a następnie na panelu wybiera i zatwierdza interesujący go zestaw.

- b. Poniżej, w wydzielonej strefie, wyświetlane są pola: słówko do przetłumaczenia oraz pole do wpisania definicji.
 - c. Po zatwierdzeniu przyciskiem "Sprawdź", aplikacja wyświetla okno potwierdzające poprawność odpowiedzi użytkownika. Jeśli definicja jest błędna, wyświetlana jest poprawna definicja. W tym etapie użytkownik ma możliwość uznania swojej odpowiedzi za poprawną, co skutkuje usunięciem słówka z bieżącej rundy oraz zapisaniem odpowiedzi jako poprawnej w statystykach. W przeciwnym razie, jeśli odpowiedź jest błędna, słówko zostaje ponownie dodane do listy jako błędnie odpowiedziane.
 - d. Po zakończeniu rundy nauki pojawia się okno z opcją rozpoczęcia kolejnej rundy lub zakończenia sesji.
 - e. Po zakończeniu nauki, za pomocą wyżej opisanego okna lub przycisku znajdującego się na dole strony, wyświetlany jest wykres kołowy, który przedstawia stosunek poprawnych i błędnych odpowiedzi w danej sesji nauki.
8. **Zakładka "Przypomnienia"** – umożliwia wybór daty, godziny i nazwy przypomnienia o następnej sesji nauki, które zostanie zapisane w kalendarzu Google.
9. **Zakładka "Wyloguj"** – po naciśnięciu tego przycisku użytkownik zostaje wylogowany, a wszystkie zmiany zostają zapisane. Po wylogowaniu pojawi się panel startowy. Aby ponownie uzyskać dostęp do aplikacji, należy powtórzyć kroki od 1.
-

Podsumowanie i wnioski

Projekt udało się zrealizować zgodnie z kluczowymi założeniami. Aplikacja jest łatwa w użyciu i skutecznie wspiera użytkowników w nauce poprzez oferowanie funkcji takich jak dodawanie fiszek, nauka za pomocą zestawów, przypomnienia w kalendarzu Google oraz możliwość śledzenia postępów. Systematyczność w nauce została wspierana przez funkcje przypomnień, co przyczynia się do lepszych wyników użytkowników.

Jednakże istnieje kilka obszarów, które mogą zostać usprawnione i rozwinięte w przyszłości:

1. **Autoryzacja Google** – Dodanie funkcji zapamiętywania autoryzacji konta Google dla konkretnego użytkownika mogłoby przyspieszyć proces

logowania, eliminując konieczność każdorazowego logowania się. Taka funkcjonalność poprawiłaby wygodę użytkownika aplikacji.

2. **Użytkownik bez dostępu do konta Google** - Zablokowanie dostępu do panelu Przypomnienia w przypadku, gdy użytkownik loguje się bez dostępu do konta Google
3. **Przechowywanie danych użytkowników** – Zapis danych użytkowników w systemie lokalnym jest rozwiązaniem tymczasowym. W przyszłości warto rozważyć przejście na bazę danych, co umożliwiłoby przechowywanie kont użytkowników w chmurze i dostęp do nich na różnych urządzeniach.
4. **Rozdzielanie kont użytkowników** – Jednym z wyzwań było prawidłowe zarządzanie kontami użytkowników. W przyszłości warto wprowadzić bardziej zaawansowane mechanizmy zarządzania użytkownikami, np. umożliwić wielokrotne logowanie lub wprowadzenie bardziej złożonego systemu uprawnień.
5. **Rozmieszczenie komponentów** – W projekcie rozmieszczenie elementów było wykonane ręcznie za pomocą różnych narzędzi layout. Chociaż ten sposób dał dużą kontrolę nad wyglądem aplikacji, w przyszłości warto rozważyć użycie bardziej zaawansowanych narzędzi do layoutu, co mogłoby uprościć ten proces i poprawić efektywność.

Podsumowując, projekt zrealizował swoje cele, jednak istnieje wiele możliwości rozwoju aplikacji, które mogą poprawić jej funkcjonalność, wygodę użytkownika i skalowalność w przyszłości.