

Projekt Dokumentation: bib_talk Chat-Anwendung

Projektstruktur

Das Projekt ist in verschiedene Dateien und Klassen unterteilt, die jeweils eine spezifische Funktion innerhalb der Anwendung erfüllen. Die Hauptverzeichnisse und Dateien sind:

- bib_talk
 - **Business:** Beinhaltet die Geschäftslogik, die zur Verarbeitung von Benutzeranforderungen erforderlich ist.
 - User.cs
 - Message.cs
 - MessageDto.cs
 - LoginManager.cs
 - RegisterManager.cs
 - MainWindowManager.cs
 - **Repository:** Enthält die Datenzugriffsschicht.
 - LoginDatabase.cs
 - RegisterDatabase.cs
 - MainWindowDatabase.cs
 - **UI Components:** WPF-basierte Fenster für Benutzerinteraktionen.
 - Login.xaml.cs
 - RegisterWindow.xaml.cs
 - MainWindow.xaml.cs

Hauptkomponenten

User-Management

Das User-Management umfasst alle Operationen, die für das Erstellen, Speichern und Verwalten von Benutzerinformationen erforderlich sind. Dies beinhaltet das Registrieren neuer Benutzer, das Anmelden und das Ausloggen von Benutzern.

- **Klassen und Methoden:**
 - User.cs: Definiert die Eigenschaften eines Benutzers, wie Id, Username, Password, Email, Birthday, IsOnline und ProfileImagePath.
 - RegisterManager.cs und RegisterDatabase.cs: Behandeln die Logik für das Registrieren neuer Benutzer, einschließlich der Validierung und Speicherung in der Datenbank.

Nachrichten-Management

Das Nachrichten-Management befasst sich mit dem Senden, Empfangen und Verwalten von Nachrichten zwischen Benutzern.

- **Klassen und Methoden:**
 - Message.cs: Repräsentiert eine Nachricht, die in der Datenbank gespeichert wird, mit Feldern wie Id, Username, Text und Timestamp.
 - MessageDto.cs: Eine vereinfachte DTO-Version (Data Transfer Object) von Message für die Übertragung zwischen Client und Server.
 - MainWindowManager.cs und MainWindowDatabase.cs: Behandeln das Laden von Nachrichten, das Senden neuer Nachrichten und das Löschen von Nachrichten im Chat.

GUI-Komponenten

Die Benutzeroberfläche besteht aus mehreren Fenstern, die es dem Benutzer ermöglichen, mit der Anwendung zu interagieren:

- **Login Window:** Ermöglicht es Benutzern, sich anzumelden oder zum Registrierungsfenster zu wechseln.
- **Register Window:** Ermöglicht es neuen Benutzern, sich zu registrieren und ein neues Konto zu erstellen.
- **Main Window:** Das Hauptfenster der Anwendung, in dem Benutzer Nachrichten senden und empfangen können.

Wichtige Klassen und Methoden

User.cs

Die User-Klasse repräsentiert die Benutzerdaten und ihre Eigenschaften. Sie ist entscheidend für die Verwaltung von Benutzerinformationen sowohl auf der Client-Seite als auch auf der Server-Seite.

- **Eigenschaften:**
 - Id: Eindeutige Kennung des Benutzers.
 - Username: Benutzername des Benutzers.
 - Password: Passwort des Benutzers.
 - Email: E-Mail-Adresse des Benutzers.
 - Birthday: Geburtsdatum des Benutzers.
 - IsOnline: Status, ob der Benutzer online ist.
 - ProfileImagePath: Pfad zum Profilbild des Benutzers.

Message.cs und MessageDto.cs

Diese Klassen verwalten die Nachrichten, die zwischen den Benutzern gesendet werden.

- **Message.cs:** Wird für die Speicherung von Nachrichten in der Datenbank verwendet.
- **MessageDto.cs:** Wird für die Datenübertragung im Netzwerk verwendet.

Login und Registrierungslogik

Die Login- und Registrierungslogik stellt sicher, dass Benutzer sicher ein Konto erstellen und sich anmelden können.

- **RegisterWindow.xaml.cs:** Handhabt die Benutzeroberfläche für die Registrierung neuer Benutzer. Validiert die Eingabe des Benutzers und sendet Registrierungsdaten an den Server.
- **RegisterManager.cs:** Vermittler zwischen der Benutzeroberfläche und der Datenbanklogik.
- **RegisterDatabase.cs:** Verwendet HTTP-Anfragen, um Benutzerregistrierungen auf dem Server zu speichern.

Hauptfenster-Logik (MainWindow)

Das Hauptfenster ermöglicht es dem Benutzer, Nachrichten zu senden und zu empfangen und zeigt eine Liste der aktuell online befindlichen Benutzer an.

- **MainWindow.xaml.cs:** Beinhaltet die Logik für das Hauptfenster, einschließlich des Timers, der Nachrichtenaktualisierungen und Benutzerinteraktionen.
- **MainWindowManager.cs:** Verwaltet die Hauptoperationen für das Senden und Empfangen von Nachrichten.
- **MainWindowDatabase.cs:** Handhabt die Kommunikation mit dem Server für Nachrichten und Benutzerinformationen.

Verwendete Technologien und Bibliotheken

- **C# und .NET Framework:** Hauptprogrammiersprache und Framework für die Anwendungslogik und GUI.
- **WPF (Windows Presentation Foundation):** Framework zur Entwicklung der grafischen Benutzeroberfläche.
- **Newtonsoft.Json:** Bibliothek zur Serialisierung und Deserialisierung von JSON-Daten.
- **HttpClient:** Verwendet für HTTP-Anfragen und -Antworten zur Kommunikation mit dem Backend-Server.