Dokumentation des Vereins-Kassen-Systems

Allgemeine Beschreibung

Das Programm simuliert ein Vereinskassen-System mit einer grafischen Benutzeroberfläche (Tkinter). Es ermöglicht die Verwaltung von Abteilungskonten für verschiedene Vereinsbereiche wie Fußball oder Tanzen. Nutzer haben unterschiedliche Rollen mit spezifischen Berechtigungen:

- Administratoren: Erstellen und verwalten Nutzer sowie Konten, speichern aktuelle Kontostände.
- Kassenwarte: Tätigen Einzahlungen, Auszahlungen und Transfers zwischen Konten.
- Finanzreferenten: Sehen Transaktionshistorien und Gesamtkontenübersichten ein.

Dateien und ihre Funktionen

1. login_gui.py (Login-Fenster)

Startpunkt der Anwendung. Ermöglicht Benutzern die Authentifizierung.

Funktionen:

- **login()**
  - **Beschreibung:** Führt den Login-Prozess durch, überprüft Benutzername und Passwort.
  - **Parameter:** Keine
  - **Rückgabewert:** None
  - **Testfall:** Benutzer gibt gültige Login-Daten ein.

- **open_admin_gui()**
  - **Beschreibung:** Öffnet die Admin-Oberfläche nach erfolgreichem Login.
  - **Parameter:** Keine
  - **Rückgabewert:** None

```
[DEBUG] Benutzername eingeben: Admin
[DEBUG] Passwort eingeben: admin
[DEBUG] Gefundener Benutzer: User(username='Admin', role='Administrator', accounts=[])
[DEBUG] Erwartetes Passwort: admin
öffne Admin-Bereich
```

- **open_kassenwart_gui()**
  - **Beschreibung:** Öffnet die Kassenwart-Oberfläche nach erfolgreichem Login.
  - **Parameter:** Keine
  - **Rückgabewert:** None

```
[DEBUG] Lade Konten für Kassenwart: Ayoub
[DEBUG] Erlaubte Konten für Ayoub: []
öffne Kassenwart-Bereich
```

- **open_finanzen_gui()**
  - **Beschreibung:** Öffnet die Finanzreferenten-Oberfläche nach erfolgreichem Login.
  - **Parameter:** Keine
  - **Rückgabewert:** None

```
[DEBUG] Gefundener Benutzer: User(username='Max', role='Finance-Referent', accounts=[])
[DEBUG] Erwartetes Passwort: newPassword
öffne Finanzen referent-Bereich
```

2. admin_gui.py (Admin-Oberfläche)

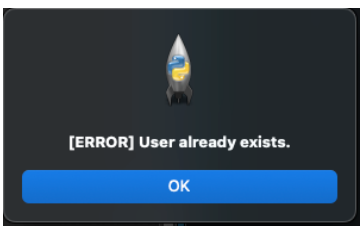
Ermöglicht Administratoren, neue Nutzer und Konten zu erstellen.

Funktionen:

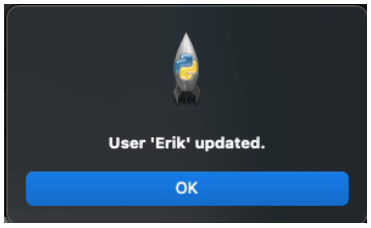
- **__init__()**
  - **Beschreibung:** Initialisiert das Admin-GUI-Fenster.
  - **Parameter:** Keine
  - **Rückgabewert:** None

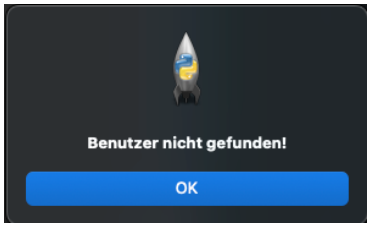
- **add_user()**
  - **Beschreibung:** Fügt ein neues Benutzerkonto hinzu.
  - **Parameter:** Keine
  - **Rückgabewert:** None

```
Adding user: Testname, Role: Administrator
Saving data to JSON file...
Benutzer erfolgreich hinzugefügt!
```

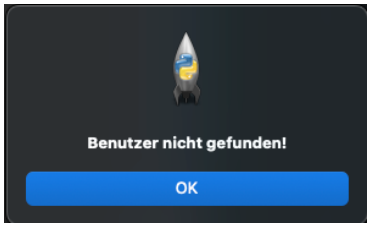


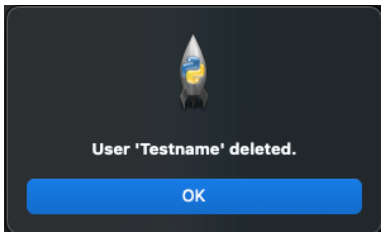
- **edit_user()**
  - **Beschreibung:** Bearbeitet ein Benutzerkonto
  - **Parameter:** Keine
  - **Rückgabewert:** None



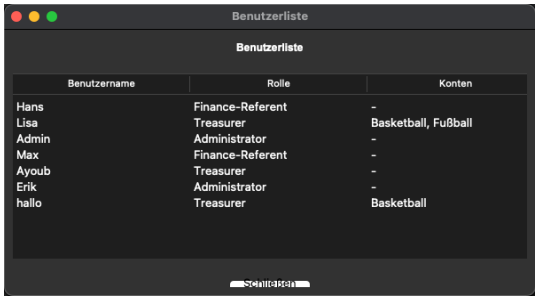


- **delete_user()**
  - **Beschreibung:** Löscht ein Benutzerkonto.
  - **Parameter:** Keine
  - **Rückgabewert:** None



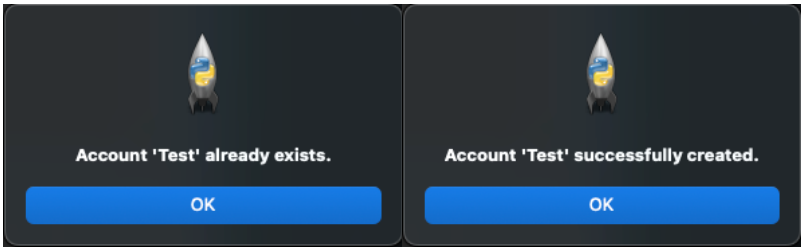


- **list_users()**
  - **Beschreibung:** Zeigt eine Liste aller Benutzer an.
  - **Parameter:** Keine
  - **Rückgabewert:** None

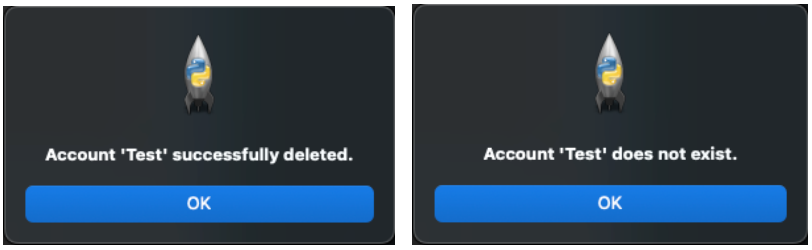


- **logout()**
  - **Beschreibung:** Schließt das aktuelle Fenster und zeigt das Login-Fenster wieder an..
  - **Parameter:** Keine
  - **Rückgabewert:** None

- **add_account()**
  - **Beschreibung:** Ermöglicht dem Admin, ein neues Vereinskonto zu erstellen.
  - **Parameter:** Keine
  - **Rückgabewert:** None



- **delete_account()**
  - **Beschreibung:** Ermöglicht dem Admin, ein Konto zu löschen, falls es ein Guthaben von 0€ hat.
  - **Parameter:** Keine
  - **Rückgabewert:** None

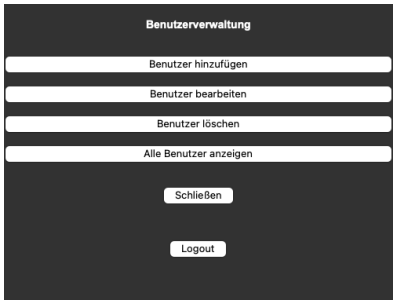


- **show_account_summary()**
  - **Beschreibung:** Zeigt eine Übersicht aller existierenden Konten mit Namen und Saldo in der GUI.
  - **Parameter:** Keine
  - **Rückgabewert:** None

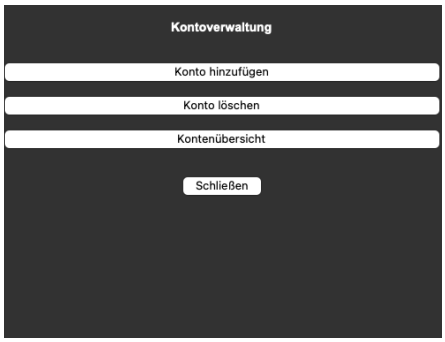
- **view_account_history()**
  - **Beschreibung:** Zeigt eine Übersicht aller existierenden Konten mit Namen und Saldo in der GUI.
  - **Parameter:** Keine
  - **Rückgabewert:** None

- **return_to_dashboard()**
    - **Beschreibung:** Schließt die aktuelle Ansicht und zeigt das Admin-Dashboard wieder an.
    - **Parameter:** Keine
    - **Rückgabewert:** None

- **open_user_managment()**
    - **Beschreibung:** Öffnet das Benutzerverwaltungs-Fenster.
    - **Parameter:** Keine
    - **Rückgabewert:** None



- **open_account_managment()**
    - **Beschreibung:** Öffnet das Kontoverwaltungs-Fenster.
    - **Parameter:** Keine
    - **Rückgabewert:** None



3. accounts_manager.py (Kontoverwaltung)

Steuert Einzahlungen, Auszahlungen und Umbuchungen.

Funktionen:

- **__init__()**
    - **Beschreibung:** Initialisiert die Kontoverwaltung.
    - **Parameter:** Keine
    - **Rückgabewert:** None

- **save_data()**
  - **Beschreibung:** Überschreibt die gespeicherten Daten in der JSON
  - **Parameter:** Keine
  - **Rückgabewert:** None

- **create_account()**
  - **Beschreibung:** Erstellt ein neues Vereinskonto.
  - **Parameter:** Keine
  - **Rückgabewert:** Dict

```
{'success': "Account 'Tennis' successfully created."}
{'error': "Account 'Tennis' already exists."}
```

- **delete_account()**
  - **Beschreibung:** Löscht ein Vereinskonto.
  - **Parameter:** name: String
  - **Rückgabewert:** Dict

```
{'success': "Account 'Tennis' successfully deleted."}
{'error': "Account 'Football' does not exist."}
```

- **deposit()**
  - **Beschreibung:** Einzahlen auf ein Vereinskonto.
  - **Parameter:** name: String, amount: float, source: String, note: string = „“
  - **Rückgabewert:** Dict

```
{'success': "100€ deposited into 'Tennis'."}
{'error': 'Amount must be positive.'}
{'error': "Account 'Unknown' does not exist."}
```

- **withdraw()**
  - **Beschreibung:** Auszahlen auf ein Vereinskonto.
  - **Parameter:** name: String, amount: float, note: string = „“
  - **Rückgabewert:** Dict

```
Withdrawal: 50€ from 'Basketball'
Saving data to JSON...
{'success': "50€ withdrawn from 'Basketball'.", 'transaction': {'date': '2025-01-31 21:59:06', 'account': 'Basketball', 'type': 'Withdrawal', 'amount': 50, 'source': 'Account', 'note': 'Purchase', 'target_account': None}}
Withdrawal: 1000€ from 'Basketball'
Saving data to JSON...
{'success': "1000€ withdrawn from 'Basketball'.", 'transaction': {'date': '2025-01-31 21:59:06', 'account': 'Basketball', 'type': 'Withdrawal', 'amount': 1000, 'source': 'Account', 'note': 'Overdraw Test', 'target_account': None}}
Withdrawal: 50€ from 'Unknown'
{'error': "Account 'Unknown' does not exist."}
```

4. kassenwart_gui.py (Kassenwart-Oberfläche)

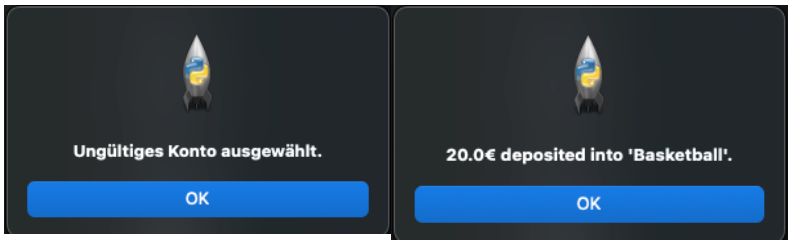
GUI für Kassenwarte zur Verwaltung ihrer Konten.

Funktionen:

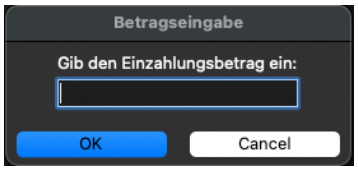
- **__init__()**
  - **Beschreibung:** Initialisiert das Kassenwart-GUI-Fenster.
  - **Parameter:** Keine
  - **Rückgabewert:** None

- **render_dashboard()**
  - **Beschreibung:** Zeigt das Dashboard mit Kontoinformationen.
  - **Parameter:** Keine
  - **Rückgabewert:** None

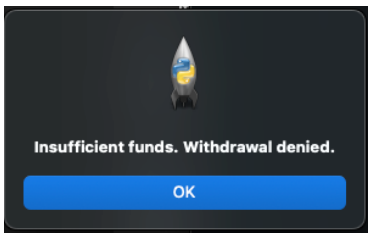
- **deposit_money()**
  - **Beschreibung:** Führt eine Einzahlung durch.
  - **Parameter:** Keine
  - **Rückgabewert:** None

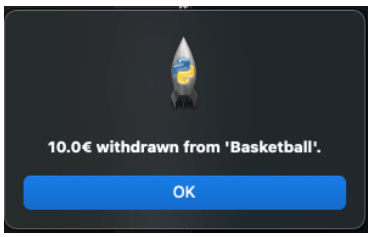


- **get_amount()**
  - **Beschreibung:** Zeigt einen Dialog zur Eingabe eines Betrags und validiert die Eingabe.
  - **Parameter:** prompt: String
  - **Rückgabewert:** None

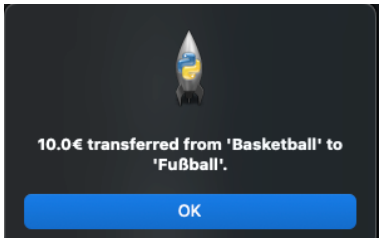


- **withdraw_money()**
  - **Beschreibung:** Führt eine Auszahlung durch.
  - **Parameter:** Keine
  - **Rückgabewert:** None





- **transfer_money()**
  - **Beschreibung:** Ermöglicht dem Kassenwart eine Umbuchung zwischen seinen Konten.
  - **Parameter:** Keine
  - **Rückgabewert:** None



- **show_transactions()**
  - **Beschreibung:** Ermöglicht dem Kassenwart eine Umbuchung zwischen seinen Konten.
  - **Parameter:** Keine
  - **Rückgabewert:** None

5. finanzen_gui.py (Finanzreferenten-Oberfläche)

Zeigt Finanzberichte und eine Übersicht aller Vereinskonten.

Funktionen:

- **__init__()**
  - **Beschreibung:** Initialisiert das Finanzreferenten-GUI-Fenster.
  - **Parameter:** Keine
  - **Rückgabewert:** None

- **show_dashboard()**
    - **Beschreibung:** Zeigt das Finanz-Dashboard.
    - **Parameter:** Keine
    - **Rückgabewert:** None

- **view_overview()**
    - **Beschreibung:** Zeigt eine Übersicht über alle Vereinskonten.
    - **Parameter:** Keine
    - **Rückgabewert:** None

- **view_account_history()**
    - **Beschreibung:** Zeigt die Transaktionshistorie eines Kontos.
    - **Parameter:** Keine
    - **Rückgabewert:** None

- **logout()**
    - **Beschreibung:** Meldet den Benutzer ab.
    - **Parameter:** Keine
    - **Rückgabewert:** None

6. user_manager.py (Benutzerverwaltung)

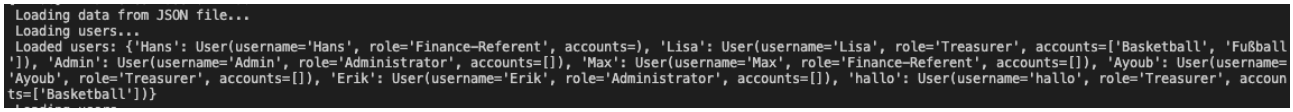
Verwaltet Benutzerrechte und Passwörter.

Funktionen:

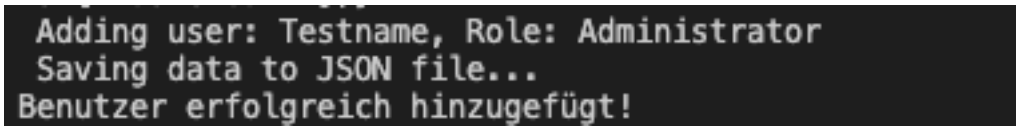
- **__init__()**
    - **Beschreibung:** Initialisiert das Benutzerverwaltungs-Modul.
    - **Parameter:** Keine
    - **Rückgabewert:** None

- **__repr__()**
    - **Beschreibung:** Gibt eine lesbare Repräsentation des Benutzers zurück.
    - **Parameter:** Keine
    - **Rückgabewert:** String

- **load_data()**
    - **Beschreibung:** Lädt Benutzerdaten aus der Datenbank.
    - **Parameter:** Keine
    - **Rückgabewert:** None



- **add_user()**
    - **Beschreibung:** Lädt Benutzerdaten aus der Datenbank.
    - **Parameter:** username: str, password: str, role: str, accounts: (list, optional)
    - **Rückgabewert:** String



- **edit_user()**
    - **Beschreibung:** Lädt Benutzerdaten aus der Datenbank.
    - **Parameter:** username: str, new_username: (str, optional), new_password: (str, optional), new_role: (str, optional), new_accounts: (list, optional)
    - **Rückgabewert:** String
- **delete_user()**
    - **Beschreibung:** Lädt Benutzerdaten aus der Datenbank.
    - **Parameter:** username: str
    - **Rückgabewert:** String
- **list_users()**
    - **Beschreibung:** Lädt Benutzerdaten aus der Datenbank.
    - **Parameter:** Keine
    - **Rückgabewert:** String
- **get_kassenwart_accounts()**
    - **Beschreibung:** Lädt Benutzerdaten aus der Datenbank.
    - **Parameter**: username: str
    - **Rückgabewert**: dict[str, str]