

Dokumentacja Pakietu Frog – Sklep Żabka Online

1. Wstęp

Pakiet **Frog** to aplikacja desktopowa napisana w Pythonie, służąca do zarządzania sklepem internetowym Żabka. Umożliwia administrację zasobami produktów spożywczych oraz obsługę bazy klientów poprzez interfejs graficzny oparty na tkinter.

2. Struktura Projektu

```
frog/
├── data/
│   ├── products.xlsx      # Zasoby produktów spożywczych
│   └── customer.csv       # Dane klientów ze statusem stałego klienta
├── DATABASE/
│   └── <ID_klienta>.txt   # Indywidualne pliki historii zakupów klientów
├── add_delete_products.py # Moduł dodawania i usuwania produktów
├── add_delete_user.py     # Moduł rejestracji i usuwania użytkowników
├── main.py                # Główny moduł uruchamiający GUI i łączący
├── funkcje
│   ├── gui.py             # Definicje okien i handlerów GUI
│   ├── requirements.txt   # Lista zależności
│   └── README.md         # Instrukcja i opis projektu
```

3. Instalacja i Konfiguracja

1. Sklonuj repozytorium:

```
https://github.com/bAleksiejczuk/Zabka.git
```

2. Utwórz wirtualne środowisko i zainstaluj zależności :

```
``bash
python3 -m venv venv
source venv/bin/activate      # Linux/MacOS
venv\Scripts\activate        # Windows
pip install -r requirements.txt
```

1. Upewnij się, że folder `data/` zawiera plik `products.xlsx` i `customer.csv` .

4. Zależności

- Python 3.7
- pandas
- openpyxl
- pytz
- tkinter (w zestawie standardowym)
- csv (moduł standardowy)

Zainstaluj wszystkie pakiety poleceniem:

```
pip install pandas openpyxl pytz
```

5. Uruchomienie i Przykłady Użycia

Aby uruchomić aplikację:

```
python -m main
```

Klient lub administrator zobaczy okno logowania, a po poprawnym zalogowaniu główne okno sklepu.

Przykład: Dodawanie produktu

1. W polach „Nazwa produktu”, „Ilość” i „Cena” wpisz wartości.
2. Kliknij „Dodaj Produkt”.

Przykład: Rejestracja użytkownika

1. Wprowadź email, hasło, imię i telefon w sekcji **Zarządzanie Użytkownikami**.
2. Kliknij „Dodaj Użytkownika”.

6. Dokumentacja API

Poniżej opisano kluczowe moduły oraz wybrane funkcje.

6.1 Moduł `add_delete_products.py`

Zawiera obsługę pliku `data/products.xls` .

```
add_product(name_entry: tk.Entry, count_entry: tk.Entry, price_entry: tk.Entry) -> None
```

- **Cel:** Dodaje nowy wiersz do arkusza produktów.
- **Parametry:**
 - `name_entry` – widget wprowadzający nazwę (max 20 znaków).
 - `count_entry` – widget wprowadzający liczbę sztuk (1–99999).
 - `price_entry` – widget wprowadzający cenę w formacie `XX.XX` .

Działanie:

- Odczytuje wartość z każdego pola.
- Waliduje długość nazwy, liczbę i format ceny (regex `^\d+\.\d{2}$`).
- Generuje nowe ID (najwyższe istniejące + 1).
- Dopisuje wiersz do pliku przy użyciu `openpyxl` .
- **Wyjątki:**
 - `ValueError` przy niepoprawnym `int()` / `float()`.
 - `Ogólne Exception` przy błędach I/O lub `openpyxl`.

```
delete_product(delete_entry: tk.Entry) -> None
```

- **Cel:** Usuwa wszystkie wiersze o podanej nazwie.

-
- **Parametry:**
- `delete_entry` – widget wprowadzający nazwę do usunięcia.
- **Działanie:**
- Wczytuje dane arkusza jako `DataFrame` .
- Filtruje wiersze nierównające się nazwie (case-insensitive).
- Zapisuje wynik z powrotem do pliku.
- **Wyjątki:**
- Ogólne `Exception` przy błędach I/O.

6.2 Moduł `add_delete_user.py`

Obsługuje plik `data/customer.csv` i folder `DATABASE/` .

`add_user(email_entry: tk.Entry, password_entry: tk.Entry, name_entry: tk.Entry, phone_entry: tk.Entry) -> None``

- **Cel:** Rejestruje użytkownika z unikalnym 4-cyfrowym ID.
- **Parametry:**
- `email_entry` – widget e-mail.
- `password_entry` – widget hasło.
- `name_entry` – widget imię i nazwisko.
- `phone_entry` – widget numer telefonu.
- **Działanie:**
- Waliduje format e-mail (regex).
- Sprawdza istnienie pliku CSV.
- Generuje unikalne ID (random 1000–9999).
- Dopisuje nowy rekord do CSV z timestampem (`datetime.now(pytz.timezone("Europe/Warsaw"))`)).
- Tworzy pusty plik `DATABASE/{ID}.txt` .
- **Wyjątki:**
- `IOError` przy błędach odczytu/zapisu pliku.
- `messagebox.showerror` dla nieprawidłowego e-maila lub braku pliku.

`delete_user(delete_entry: tk.Entry) -> None`

- **Cel:** Usuwa użytkownika i archiwizuje dane.

- **Parametry:**
- `delete_entry` – widget e-mail do usunięcia.
- **Działanie:**
- Wczytuje wszystkie rekordy z CSV.
- Filtruje użytkownika według e-maila.
- Dopisuje szczegółowe dane do `DATABASE/ARCHIVUM/Archivum.txt` .
- Usuwa plik `DATABASE/{ID}.txt` .
- Zapisuje zaktualizowaną listę klientów.
- **Wyjątki:**
- `FileNotFoundError` przy braku CSV.
- Ogólne `Exception` przy problemach plikowych.

6.3 Moduł `main.py`

Łączy funkcjonalności i inicjuje GUI.

`validate_length(text: str) -> bool`

- **Cel:** Sprawdza długość tekstu (50 znaków).
- **Parametry:** `text` – ciąg do walidacji.
- **Zwraca:** `True / False` . `show_users()` -> `None`
- **Cel:** Otwiera nowe okno z listą klientów.
- **Parametry:** Brak.
- **Działanie:** Wczytuje `data/customer.csv` przez `pandas.read_csv()` , generuje widgety dla każdego rekordu.
- **Wyjątki:**
- `FileNotFoundError` – wyświetla komunikat o braku pliku.
- Ogólne `Exception` – komunikat o błędzie wczytywania.

`main()` -> `None`

- **Cel:** Główna pętla aplikacji.
- **Działanie:**
- Tworzy okno `Tk()` .
- Podłącza przyciski do `add_product` , `delete_product` , `add_user` , `delete_user` , `show_users`
- Wywołuje `root.mainloop()` .

7. Działanie Komponentów

- **`GUI(`**)``:** definiuje okna logowania, koszyka i płatności, wykorzystując funkcję wyższego rzędu `create_message_handler``.
- **Dane:** `products.xlsx` trzymane jako arkusz Excel z arkuszem zawierającym rekordy CSV w pojedynczej kolumnie.
- **Baza Klientów:** `customer.csv` z nagłówkami
`ID, NAME, EMAIL, PASSWORD, PHONE, POINTS, CREATED, UPDATED` .

8. Obsługa Wyjątków

W krytycznych funkcjach użyto bloków `try/except` do:

- Walidacji danych (`ValueError`).
- Operacji I/O (`IOError` , `Exception`).
- Braku plików (`FileNotFoundError`).

9. Plany Rozszerzeń

- Moduł raportów sprzedaży (`analytics.py`).
- Integracja z bazą SQL (`db_connector.py`).
- Autoryzacja JWT dla API.

10. Licencja

Projekt rozpowszechniany na licencji MIT.