

Mikołaj Rożek, nr indeksu: 325214

Cel projektu:

Celem projektu było stworzenie aplikacji obsługującej bibliotekę, umożliwiając pracę na stanowisku czytelnika oraz bibliotekarza.

Podział repozytorium

Repozytorium jest podzielone na następujące katalogi:

Katalog classes, w nim znajdują się wszystkie klasy użyte w projekcie:

- Moduł `book.py` - w nim napisany jest kod dla klasy `Book`, jedna instancja klasy `Book` reprezentuje jedną fizyczną książkę. Przechowuje ona podstawowe dane służące do opisu książki (tytuł, id, czy jest dostępna).
- Moduł `renting.py` - w nim napisany jest kod dla klasy `Renting`, która przedstawia jedno wypożyczenie książki przez czytelnika. Znajdują się w niej atrybuty które domyślnie nadają wypożyczeniu datę wygaśnięcia oraz datę wypożyczenia. Przechowuje ona również informację na temat tego jaka książka została wypożyczona. W klasie znajdują się metody odpowiadające za zwrócenie książki do biblioteki, zakończenie wypożyczenia.
- Moduł `member.py` - w nim znajduje się kod klasy `Member`, która reprezentuje jednego użytkownika biblioteki. Użytkownik posiada atrybuty takie jak `login`, który pozwala na zalogowanie się do biblioteki na stronie startowej. W klasie znajdują się atrybuty które opisują aktualne rezerwacje, wypożyczenia oraz historię wypożyczeń użytkownika. Jeżeli użytkownik posiada Status 'Librarian' czyli bibliotekarz to jego atrybuty odnośnie wypożyczeń są nieaktywne. W klasie są również zaimplementowane metody, które służą do np. Zwrotu książek, anulowania rezerwacji czy wypożyczenia książki
- Moduł `library.py` - w nim znajduje się kod klasy `Library`, która odpowiada za aktualizowanie plików JSON, podczas korzystania z aplikacji. Jest to taka 'jednostka sterująca' całą aplikacją, to z niej wychodzą wszystkie połączenia odnośnie czegoś wykonanego przez użytkownika. Najpierw wołana jest metoda instancji klasy `Library`, a później metody dla wypożyczeń, użytkowników lub książek.

- Moduł `libapp_exceptions.py` - w tym module znajdują się wszystkie wyjątki, które stworzyłem z myślą o lepszym opisie projektu i tego co w nim się dzieje.

Katalog `misc_functions`, w nim znajdują się moduły, które odpowiadają np.:

- Moduł `plot_functions.py` - odpowiada za rysowanie wykresów, które pokazują statystyki jednego czytelnika lub statystyki wszystkich wypożyczeń biblioteki w ciągu ostatnich 12 miesięcy.
- Moduł `time_functions.py` - znajdują się w nim funkcje odpowiadające za obliczanie domyślnych dat wypożyczeń oraz formatowanie ich. Znajduje się w nim również funkcja, która przygotowuje daty do wyświetlenia na wykresie.
- Moduł `lists_from_files.py` - znajdują się w nim funkcje odpowiadające za załadowanie danych z plików JSON i zwrócenie list odpowiednio z instancjami klasy `Book`, `Member` oraz `Renting`.

Katalog `ui`, w nim znajdują się kod do GUI napisanego dzięki PyQt oraz designerowi.

Katalog `tests`, w nim znajdują się testy do klas oraz funkcji, które mogłem stestować. Niestety nie wiedziałem jak przetestować kod do GUI, więc nie podjąłem się tego.

Katalog `jsonfiles`, w nim znajdują się pliki JSON, w których zapisywane są dane z biblioteki.

Instrukcja obsługi:

Po uruchomieniu aplikacji `Libapp`, wyświetli się strona, która poprosi użytkownika o wpisanie jego loginu. Jeżeli loginu nie ma w bazie danych to użytkownik nie może się

zalogować. Dla Pana Doktora Masarczyka przygotowane są konta wmasarczykclient oraz wmasarczyklib odpowiednio dla trybu czytelnika oraz bibliotekarza.

Czytelnik

Po zalogowaniu się na konto czytelnika, wyświetli się nam strona główna, z której można przechodzić na inne. Każda strona odpowiedzialna jest za coś innego... Zacznę od przeglądania książek. Książki można przeglądać poprzez filtr gatunku.

Czytelnik wybiera z jakiego gatunku chciałby wyświetlić książki. Po naciśnięciu na drugiej liście wyświetli się lista książek z wybranego gatunku. Klikając na książkę może wyświetlić więcej informacji na jej temat takie jak ile osób ją zarezerwowało, czy jest dostępna. Jeżeli o rezerwację chodzi to trochę jest to płytkie, nie miałem pomysłu jak to zaimplementować...

Czytelnik może wypożyczyć książkę poprzez kliknięcie przycisku Borrow book. Wszystko automatycznie zapisuje się w plikach oraz setupuje wraz z używaniem aplikacji. Użytkownik może również zarezerwować książkę poprzez kliknięcie przycisku Reserve a Book. Po prawie każdej akcji użytkownikowi wyświetla się informacja zwrotna o sukcesie akcji lub jej niepowodzeniu i przyczynie niepowodzenia.

Czytelnik może również przeglądać swoje własne wypożyczenia poprzez kliknięcie w głównym menu przycisku Display current rentings. Jeżeli użytkownik nic nie wypożyczył wyświetli mu się komunikat, że nie ma nic wypożyczonego. Może zwrócić książkę do biblioteki lub przedłużyć wypożyczenie o kolejny miesiąc. Czytelnik może również przeglądać swoje rezerwacje, gdzie może anulować zarezerwowanie książki lub wypożyczyć ją (jeśli oczywiście jest dostępna).

Aby przeglądać historię swoich wypożyczeń należy kliknąć w przycisk Display renting history. Po naciśnięciu go ku oczom użytkownika ukaże się kalendarz, na którym na zielono są podświetlone dni, w których użytkownik dokonał wypożyczenia książki. Jeśli w dniu nie było żadnego wypożyczenia przez użytkownika to taki dzień nie jest podświetlony żadnym kolorem a po kliknięciu w niego użytkownikowi ukaże się komunikat, że nic nie pożyzył danego dnia. Po naciśnięciu w "zielony dzień" wyświetlana jest lista wypożyczeń danego dnia, można wyświetlić więcej informacji szczegółowych o wypożyczeniu poprzez kliknięcie na wypożyczenie na liście. Do wylogowania się z konta służy przycisk Logout.

Na każdej stronie znajduje się przycisk Main Menu, który poprzez kliknięcie zabiera użytkownika do głównego menu czytelnika.

Bibliotekarz

Po zalogowaniu się wyświetla się strona główna, skąd bibliotekarz może wybrać akcje, które chce wykonać. Pierwszą możliwością jest dodanie książki do biblioteki. Bibliotekarz może przejść na stronę dodawania książki poprzez kliknięcie przycisku Add book to library.

Na stronie należy wpisać informację o książce jaką użytkownik chce dodać do biblioteki, czyli tytuł, autorów, gatunek oraz id (id nie trzeba wpisywać ustawi się automatycznie niewpisane) Jeżeli użytkownik zostawi jakieś pole tekstowe puste, to wyskoczy mu błąd.

Podobnie do wyżej wymienionej strony działa strona, która wyświetla się po naciśnięciu w przycisk Add member to library, tyle, że użytkownik nie dodaje wtedy książki, lecz nowego członka do biblioteki. Należy wybrać czy członek ten będzie bibliotekarzem czy czytelnikiem oraz wybranie mu loginu. Jeżeli login jest zajęty, wyświetli się błąd. Chciałem zrobić okienka CheckBox dla wyboru czy ma być to Client czy Librarian, ale nie udawało mi się po w sumie kilku godzinach walki i stwierdziłem, że zaimplementuję co innego.

Strona Remove book from library pozwala bibliotekarzowi na usuwanie książek z biblioteki. Poprzez klikanie na listę wyświetla się więcej informacji o książce. Jeżeli nie ma żadnych książek w bibliotece, zostanie wyświetlony komunikat o tym. Usunięcie książki powoduje automatyczne zakończenie wypożyczenia (jeśli ktoś ma ją wypożyczoną) oraz anulowanie rezerwacji (jeśli jest przez kogoś zarezerwowana).

Jeśli bibliotekarz postanowi usunąć czytelnika z biblioteki to może tego dokonać poprzez naciśnięcie przycisku Display members. Ku jego oczom ukaże się lista aktywnych czytelników w bibliotece. Z tej strony bibliotekarz może również wyświetlić statystyki z ostatnich dwunastu miesięcy w bibliotece (ile wypożyczeń dokonał czytelnik w danym miesiącu). Jeżeli bibliotekarz usunie czytelnika z biblioteki. Jego wypożyczenia zostaną automatycznie zwrócone do biblioteki a jego rezerwacje anulowane (status książek, które były przez czytelnika wypożyczone zostanie zaktualizowany).

Bibliotekarz może również otrzymać podgląd statystyk wypożyczeń całej biblioteki (przez wszystkich czytelników) w ciągu ostatnich 12 miesięcy poprzez kliknięcie w przycisk Display info.

Do wylogowania się służy oczywiście przycisk Logout, a przyciski Main menu podobnie jak w przypadku pracy w trybie czytelnika zabierają bibliotekarza na jego stronę główną.

Używane biblioteki, w ramach projektu

- Biblioteka dataclasses, z niej używam dataclass oraz field, pomogły mi one w zaprojektowaniu szkieletów klas Book, Member oraz Renting
- Biblioteka datetime oraz relativedelta, pomogły mi z zaimplementowaniem formatowania dat oraz ich obliczania
- Biblioteka typing, pobieram z niej klasę List oraz Union, które pomogły mi przy inicjalizowaniu typów zmiennych przy pisaniu klas.
- Biblioteka json, odpowiedzialna za formatowanie plików json tak aby dało się z nich korzystać w języku python
- Qt, biblioteka dzięki której zaprojektowałem Interfejs graficzny
- Matplotlib, biblioteka, która posłużyła mi w rysowaniu wykresów dla użytkowników oraz biblioteki.

Zapisywane pliki

Pliki, z których korzysta biblioteka są trzy:

Members.json - przykładowy format pliku, najpierw jest klucz (jeden z trzech "Books", "Members" oraz "Rentings"). Klucz wskazuje na wartość, którą jest lista słowników. Dane każdej instancji z trzech wyżej wymienionych są przechowywane w słowniku, gdzie mogą być zagnieżdżone listy słowników, np. W przypadku klasy member. Member posiada klucz "Current rentings", który wskazuje na listę słowników (słownikami są wypożyczenia, które z kolei posiadają na kluczu Book kolejny zagnieżdżony słownik apropos książki). Wszystko obsługuje nam klasa Biblioteki oraz metody, zaimplementowane poprzez dekorator "@classmethod". Przykład pliku members.json:

```
{
  "Members": [
    {
      "Name": "Wojciech",
      "Surname": "Masarczyk",
      "Login": "wmasarczyklibrarian",
      "Status": "Librarian",
      "Current rentings": [],
      "Renting history": [],
      "Current reservations": []
    },
    {
      "Name": "Mikolaj",
      "Surname": "Rozek",
      "Login": "mikej9lib",
      "Status": "Librarian",
      "Current rentings": [],
      "Renting history": [],
      "Current reservations": []
    },
    {
      "Name": "Wojciech",
      "Surname": "Masarczyk",
      "Login": "wmasarczykclient",
      "Status": "Client",
      "Current rentings": [],
      "Renting history": [],
      "Current reservations": []
    },
    {
      "Name": "Mikolaj",
      "Surname": "Rozek",
      "Login": "mikej9client",
      "Status": "Client",
      "Current rentings": [],
      "Renting history": [],
      "Current reservations": []
    }
  ]
}
```

Moje przemyślenia po zakończeniu projektu

Uważam, że projekt był super zabawą, a progres jakiego dokonałem poprzez siedzenie tyle czasu nad GUI, importem oraz exportem plików, klasami jest przeogromny. Uważam, że wiele rzeczy mogło być zrobionych lepiej (na przykład te przeklęte rezerwacje), ale nie pozwolił mi na to czas. Niemniej jednak jestem nawet zadowolony z tego co powstało i uważam, że to super uczucie stworzyć coś totalnie od ZERA. Dowiedziałem się bardzo dużo nowych rzeczy oraz wiele nauczyłem.