Dokumentacja systemu rejestrującego czas pracy

Wojciech Kozłowski

16 kwietnia 2020

1 Założenia symulacji

Program jest napisany w języku python w wersji 3.6. Program symuluje działanie czytnika kart RFID. Projekt jest podzielony na podprogramy:

- jeden program główny (będący serwerem)
- dowolna ilość programów symulujących termial z czytnikiem RFID.

2 Implementacja

2.1 Serwer

2.1.1 Pola

• `term_file`, `logs_file`, `card_file` - nazwy plików w których składuje się dane (w formacie csv).
• `terminals` - lista zawierająca informacje o ID terminala i jego położeniu (`"in"` albo `"out"`)
• `cards` - lista zawierająca informacje o numerze karty, imieniu i nazwisku jej posiadacza.

2.1.2 Metody

• `load_cards_from_file` - metoda pobierająca z pliku o nazwie przechowywanej w `card_file` identyfikatory zarejestrowanych kart wzraz z imionami i nazwiskami ich posiadaczy. Wszystkie informacje trafiają do pola `cards`.
Argumenty brak
Zwraca `None`

• `load_term_from_file` - to samo co wyżej, ale informacje dotyczą terminali.
Argumenty brak
Zwraca `None`

• `append_new_card_to_file` - metoda dodająca informacje o nowo dodanej karcie do pliku. Metoda wykonywana prywatnie, interface `Menu` nie pozwala się do niej dostać.
Argumenty brak
Zwraca `None`

• `append_new_term_to_file` - to samo dla terminali
Argumenty brak
Zwraca `None`

• `rewrite_cards_in_file` - Po usunięciu, lub modyfikacji danych związanych z kartami całość jest przepisywana na nowo, ponieważ i tak przy zmianie w pliku trzeba byłoby przejść po każdej linii. Metoda wykonywana prywatnie.
Argumenty brak
Zwraca `None`

• `rewrite_terms_in_file` - To samo dla terminali.
Argumenty brak
Zwraca `None`

• `add_card` - dodaje nową karte do listy `cards`, oraz do pliku przechowującego informacje o kartach. Gdy karta z takim samym numerem już istnieje, to nie zostaje dodana.
Argumenty `card_num` - (str) numer karty (wartość, która zostaje zwracana przez czytnik RFID), `username` - (str) imię właściciela (domyślnie puste), `usersurname` - (str) nazwisko właściciela (domyślnie puste).
Zwraca `None`

• `remove_card` - karta o danym numerze zostaje usunięta z danych serwera.
Argumenty `card_num` - (int) numer karty do usunięcia
Zwraca `None`

• `add_terminal` - dodaje nowy terminal o ile podane ID nie istnieje w liście aktywnych terminali.
Argumenty `ID` - (str) ID terminalu, `place` - (str) miejsce terminalu `"in"` lub `"out"`
Zwraca `None`

• `remove_terminal` - dane o terminalu z podanym ID zostaną usunięte z danych serwera.
Argumenty `terminalID` - (str) id terminala do usunięcia z danych.
Zwraca `None`

• `find_person` - zwraca imię i nazwisko osoby, która jest właścicielem karty o podanym numerze.
Argumenty `RFIDnum` - (int) numer karty RFID
Zwraca `None`

• `gate_update` - dodaje akcje do logów. Akcja to przejście właściciela karty o danym numerze przez bramke z danym terminalem.
Argumenty `ID` - (str) ID terminalu, który wysłał informację, `RFIDnum` - (str) numer karty, która została rozpoznana w terminalu.
Zwraca `None`

• `get_logs` - zwraca liste list danych o wszystkich przejsciach przez bramki.
Argumenty brak
Zwraca `list[list[str]]`

2.2 Listener

2.2.1 Pola

• `broker` - (str) adres IP (tutaj localhost)
• `client` - (mqtt.Client) obiekt, który odpowiada za odbieranie wiadomości.
• `terget` - (Server) obiekt, na którym wywoływane są odpowiednie metody w zależności od typu wiadomości.

2.2.2 Metody

• `process_message` - wywoływana, kiedy `client` odbierze wiadomość. W zależności od typu wiadomości na obiekcie `target` wywoływane są odpowiednie metody.
Argumenty `client` - (Client) obiekt nasłuchujący, `userdata` - (None) wartość nieużywana, `message` - (MQTTMessage) obiekt zawierający treść wiadomości
Zwraca `None`

• `connect_to_broker` - ustawia wszystkie potrzebne informacje w `client` i uruchamia pętlę nasłuchującą.
Argumenty brak
Zwraca `None`

• `disconnect` - kończy nieskończoną pętlę. Uruchamiana tuż przed zakończeniem programu `main.py`.
Argumenty brak
Zwraca `None`

2.3 sender (osobny program)

2.3.1 Zmienne globalne

• `terminal_id` - (str) ID terminalu który jest obłsugiwany przez ten program.
• `place` - (str `"in"` lub `"out"`) miejsce danego terminala (przy wyjściu czy wejściu).
• `broker` - (str const) IP na którym jest broker mqtt.

2.3.2 Funckcje

• `call_worker` - wysyła wiadomość do brokera mqtt jaki pracownik wszedł przez terminal.
Argumenty `RFID_num` - (str) numer zczytanej karty pracownika.
Zwraca `None`

• `connect_to_broker` - łączy się z brokerem mqtt.
Argumenty brak
Zwraca `None`

• `disconnect_from_broker` - odłącza się od brokera mqtt.
Argumenty brak
Zwraca `None`

• `run_sender` - pętla wysyłająca wiadomości przez broker mqtt do serwera na temat numerów RFID osób, które przeszły przez bramkę (numery podane z klawiatury).
Argumenty brak
Zwraca `None`

3 Przykłady użycia

W pierwszej kolejności włączamy serwer poprzez plik `main.py`. Kiedy mamy już włączony serwer możemy niezależnie włączać programy symulujące terminale poprzez plik `sender.py`. Na początku uruchomienia programu wysyłającego musimy wprowadzić z klawiatury ID terminala (zalecane jest wprowadzanie liczb dla większej czytelności tabel, ale nie jest to wymagane). Następnie podajemy miejsce terminala (`"in"` lub `"out"`). Gdy wprowadzimy inną wartość zostanie automatycznie zmieniona na `"in"`. Dalej program w pętli pyta się nas o numer karty RFID, która przeszła przez terminal. Możemy wprowadzać dowolne wartości, ale tylko te, które są zapisane w danych serwera będą rejestrowane (pozostałe są ignorowane).
Program serwera z kolei w tle będzie reagował na przysyłane wiadomości. Dodatkowo będzie widoczne interaktywne menu podobne do tego z wersji pierwszej. Teraz jednak dodawanie terminali i akcji przejścia nie są możliwe z poziomu tego menu, gdyż są zależne od przysyłanych wiadomości. Pozostałe opcje to:

1. add card
2. change card owner
3. print terminals
4. print cards
5. print logs
6. quit

3.1 add card

Należy wpisać numer RFID, imię i nazwisko właściciela nowej karty.

3.2 change card owner

Trzeba wybrać numer karty a potem zmienić imię i nazwisko właściciela wybranej karty RFID.

3.3 print terminals

Wypisuje wszystkie aktywne terminale (ID i miejsce).

3.4 print cards

Wypisuje wszystkie aktywne karty RFID wraz z ich właścicielami.

3.5 print logs

Wypisuje wszystkie przejścia przez terminale na ekran.

4 Screen działania programu

Wyświetlone logi po przejściu przez dwie bramki po 3 osoby. Programy zczytujące karty są po prawej stronie, a program main po lewej.