Sequence Diagram: List of Places

The sequence diagram presented in Fig. 1 illustrates the dynamic interactions between the application layers (Presentation, Business Logic, and Persistence) when a registered and logged user attempts to display the list of places obeying certain criteria.

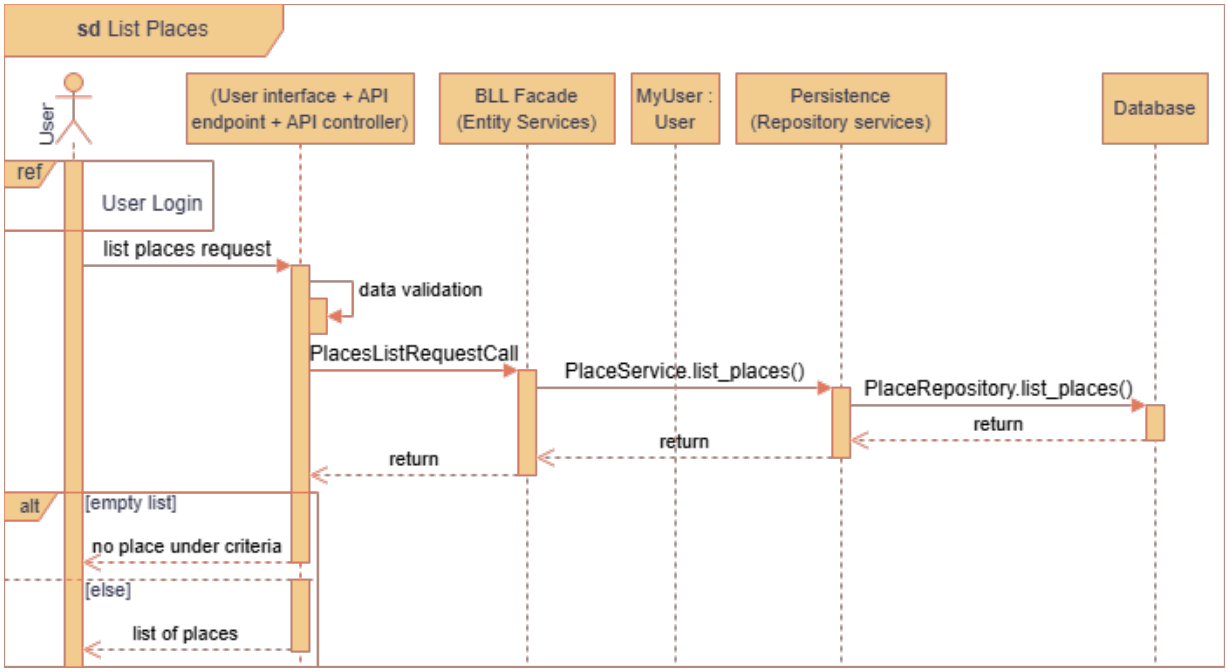

Fig. 1: list of places sequence diagram.

Actors, System Elements and Lifelines

- **User**: *The external actor who interacts with the system*
- **User Interface + API Endpoint + API Controller (Presentation Layer)**: *Responsible for receiving user requests, performing simple input validation, data deserialization/serialization, and calling the Business Logic Layer (BLL)*
- **BLL Facade (Entity Services)**: *Entry point for business logic, orchestrator of complex business operations, interactions with the presentation layer and delegation of persistence operations to repository services*
- **MyUser : User**: *Represents an instance of the User class, an object MyUser of class User*
- **Persistence Layer (Repository Services)**: *Represents the data access component responsible for direct interactions with the database, handling the storage and retrieval of specific entity data*
- **Database**: *The data storage system*

Ref User Login

Refers to the User Login sequence diagram and its flow. It calls to the fact that the user needs to be logged-in to perform the Place Creation operation, which is also why the object NewUser exists and is available at the business layer.

Process of Fetching a List of Places

List of Places Request:

- **Sender***: User*
- **Receiver***: User Interface + API Endpoint + API Controller*
- **Message***: list places request*
- **Description***: The user makes a request for a list of places*
- **Data***: Criteria Data*

Places List Request Call:

- **Sender***: User Interface + API Endpoint + API controller*
- **Receiver***: BLL Facade (Entity Services)*
- **Message***: PlaceListRequestCall*
- **Description***: After simple validation of the Criteria Data introduced by the user, the interface asks the BLL Facade to retrieve a list of places that satisfy such criteria*
- **Data***: Criteria Data*

List Places Service Call:

- **Sender***: BLL Facade (Entity Services)*
- **Receiver***: Persistence Layer (Repository Services)*
- **Message***: PlaceService.list_places()*
- **Description***: The Facade asks the Persistence Layer to find the places fulfilling the criteria in order to list them*
- **Data***: Criteria Data*

Database Verification:

- **Sender***: Persistence Layer (Repository Services)*
- **Receiver***: Database*
- **Message***: PlaceRepository.list_places()*
- **Description***: The Persistence Layer requests the Database to find the places in the database obeying the conditions in Criteria Data*
- **Data***: Criteria Data*

Places List Request Return:

- **Sender1***: Database*
- **Receiver1***: Persistence Layer (Repository Services)*
- **Sender2***: Persistence Layer (Repository Services)*
- **Receiver2***: BLL Facade (Entity Services)*

- **Sender3***: BLL Facade (Entity Services)*
- **Receiver3***: User Interface + API Endpoint + API Controller*
- **Message***: return*
- **Description***: The Persistence Layer provides a list with the id's of all places satisfying the demanded criteria, it may return an empty list if no places were found to satisfy the criteria*
- **Data***: list[place_id]/list[empty]*

Alternative Fragment:

- **Conditions***:*
    - **[empty list]***: No places corresponding to the provided criteria have been found in the database, a message is displayed saying as much.*
    - **[else]***: At least one place was found to satisfy the provided criteria and the list of all such places is presented to the user via the interface.*