

Sequence Diagram: Review Submission

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 submit a review of a chosen place.

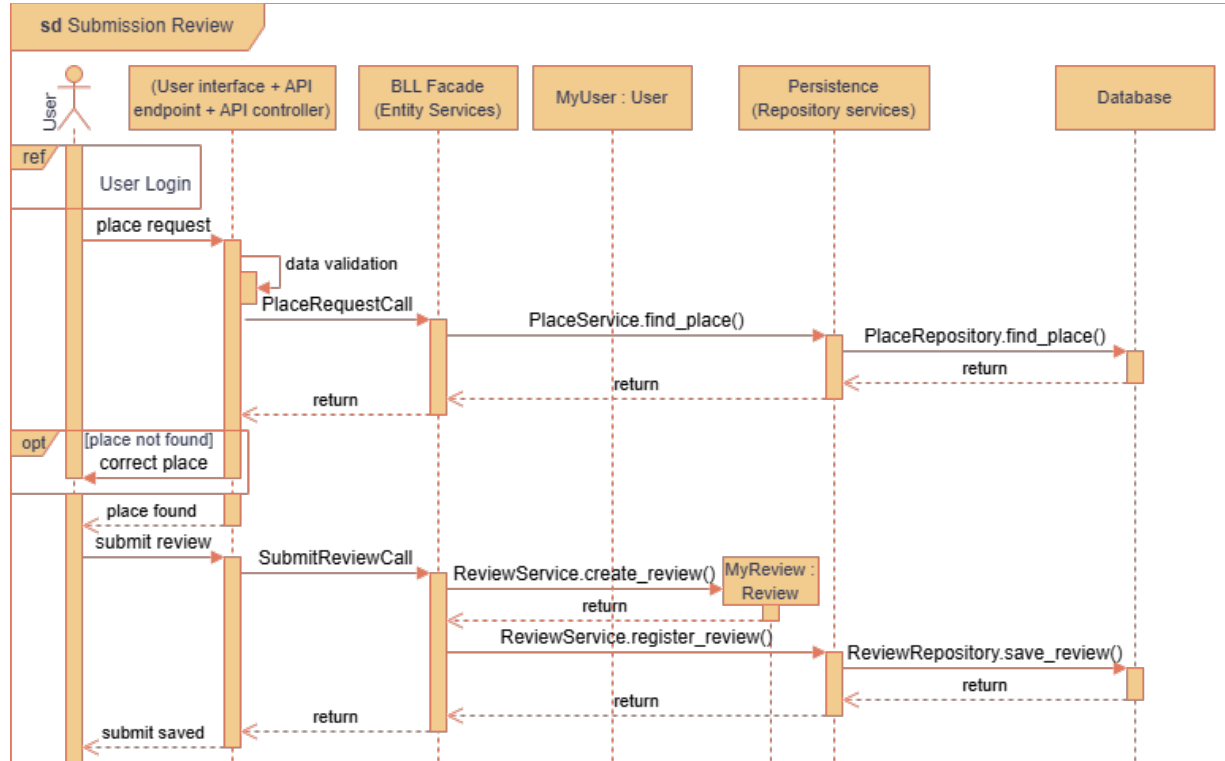


Fig. 1: review submission sequence diagram.

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.

Review Submission Process

Place Request:

- **Sender:** User
- **Receiver:** User Interface + API Endpoint + API Controller
- **Message:** *place request*
- **Description:** The user makes a request for a place's id
- **Data:** PlaceData

Place Request Call:

- **Sender:** User Interface + API Endpoint + API controller
- **Receiver:** BLL Facade (Entity Services)
- **Message:** *PlaceRequestCall*
- **Description:** After simple validation of the PlaceData introduced by the user, the interface asks the BLL Facade to find the reference id of the place to which the user provided information
- **Data:** PlaceData

Find Place Service Call:

- **Sender:** BLL Facade (Entity Services)
- **Receiver:** Persistence Layer (Repository Services)
- **Message:** *PlaceService.find_place()*
- **Description:** The Facade asks the Persistence Layer to find the chosen place
- **Data:** PlaceData

Find Place in Database:

- **Sender:** Persistence Layer (Repository Services)
- **Receiver:** Database
- **Message:** *PlaceRepository.find_place()*
- **Description:** The Persistence Layer requests the Database to retrieve the id of the selected place
- **Data:** Place Data

Place 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 the id of the place if it found it or an empty value otherwise*
- **Data:** *place_id/empty*

Option Fragment

- **Condition:** *[place not found]*
- **Description:** *If the requested place is not found in the database, an error message is returned to the user, prompting them to enter a correct place. The main flow (submitting the review for a valid place) would then not proceed*
- **Implied Flow:** *If the place is found, the flow continues normally with the *submit_review* message (this part is outside the *opt* fragment)*

Submit Review:

- **Sender:** *User*
- **Receiver:** *User Interface + API Endpoint + API Controller*
- **Message:** *submit review*
- **Description:** *The user makes a request to submit a review*
- **Data:** *Review Data, place_id*

Submit Review Call:

- **Sender:** *User Interface + API Endpoint + API controller*
- **Receiver:** *BLL Facade (Entity Services)*
- **Message:** *SubmitReviewCall*
- **Description:** *The interface transmits the review data to the BLL Facade for review creation*
- **Data:** *ReviewData, place_id*

Review Creation Service Call:

- **Sender:** *BLL Facade (Entity Services)*
- **Receiver:** *Review class*
- **Message:** *ReviewService.create_review()*
- **Description:** *The BLL Facade creates an instance of Review containing the comment made by the user regarding the place*
- **Data:** *ReviewData, place_id, MyUser.id*

Review Creation Return:

- **Sender:** *MyReview*
- **Receiver:** *BLL (Entity Services)*
- **Message:** *return*

- **Description:** *The Review class confirms the creation of a new Review instance to the BLL and returns its reference*
- **Data:** *MyReview reference*

New Review Registration:

- **Sender:** *BLL Facade (Entity Services)*
- **Receiver:** *Persistence Layer (Repository Services)*
- **Message:** *ReviewService.register_review()*
- **Description:** *The BLL Facade asks the Persistence Layer to register the new review*
- **Data:** *MyReview reference*

Data Storage:

- **Sender:** *Persistence Layer (Repository Services)*
- **Receiver:** *Database*
- **Message:** *ReviewRepository.save_review()*
- **Description:** *The Persistence Layer asks the Database to save the new review. Returns a validated creation message to the user*
- **Data:** *MyReview reference*

Register Review 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 confirms to the interface that the review has been created and stored*
- **Data:** *Signal*

Review Submission Return:

- **Sender:** *User Interface + API Endpoint + API Controller*
- **Receiver:** *User*
- **Message:** *submit saved*
- **Description:** *The user receives confirmation that its review has been successfully submitted*
- **Data:** *Message*