GetUP
Analysis Model

Submitted to:

Asst. Prof. Ma. Rowena C. Solamo
Faculty Member
Department of Computer Science
College of Engineering
University of the Philippines, Diliman

Submitted by:
Jian Chan
Gab Datiles
Hans Santos

In partial fulfillment of Academic Requirements
for the course
CS 191 Software Engineering I
of the
1st Semester, AY 2019-2020

Unique Reference:

The documents are stored in the https://github.com/gab51299/GetUP referenced with 'GetUp-7-Analysis Model.pdf' .

Purpose:

The document presents the Object Model and Behavioral Models of GetUP. Included are the related boundary, control and entity classes, the Analysis Model of the system, and the sequence diagrams of the Use-Case Model scenarios. These are provided to help visualize the elements and processes in the GetUP system.

Audience:

The document is prepared for the developers of GetUP as reference for their CS 191 & CS 192 project, as well as for Professor Ma. Solamo for the requirements of CS 191.

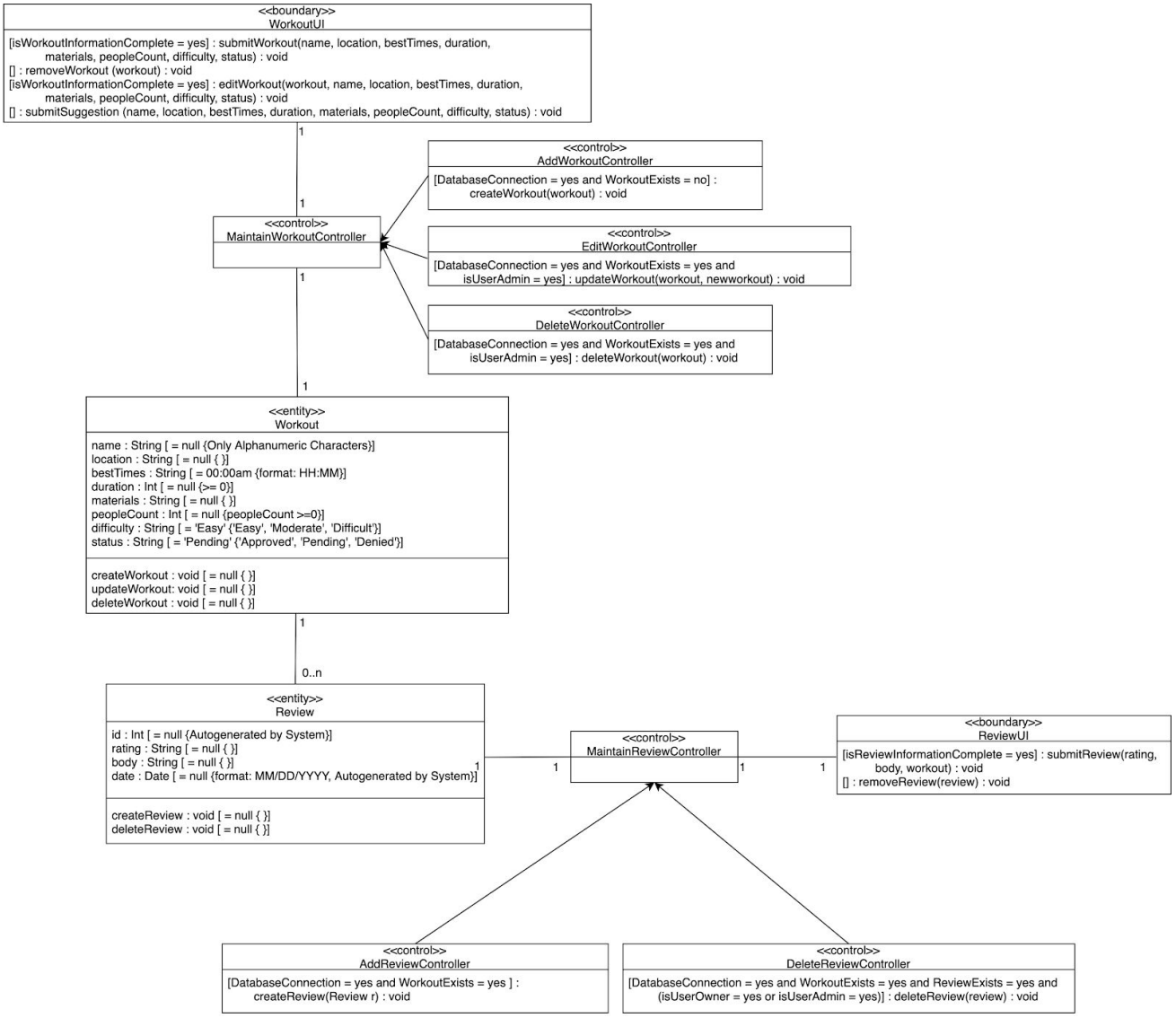
Revision Control:

Revision Date	*Person Responsible*	*Version Number*	*Modification*
09/28/19	Gab Datiles	1.0	Initial Document and Information; Add Boundary, Control, and Entity for Workouts
10/01/19	Jian Chan	2.0	Add Boundary, Control, and Entity for Reviews
10/02/19	Hans Santos	3.0	Create and Add Object Model
10/02/19	Jian Chan	4.0	Create and Add Behavioral Model for all Scenarios of all Use Cases

***System Name*:** GetUP Application System

Description: The GetUP Application is created to provide users with workouts around the UP Diliman campus, with reviews suggestions for each workout. The system also allows users to create their own workouts, or review other workouts available. They may also edit and delete their workouts if necessary.

Analysis Model:



Boundary Classes:

Class Name	Description
AthleteRecordUI	This is the interface of the club staff to the system whenever he or she needs to maintain athlete record. *Responsibilities*: public void enterAthleteData(String lastname, String firstname, String mi, String address, String postalCode, Date bday, Char gender, String status) public void enterGuardianData(String lastname String firstname, String mi, String address, String postalCode, String telephone)
WorkoutUI	This is the interface of workouts available to users and administrators whenever one wishes to Add, Delete, or Edit Workouts *Responsibilities*: public void submitWorkout (String name, String location, String bestTimes, Int duration, String materials, Int peopleCount, String difficulty, String status) public void removeWorkout (Workout w) public void editWorkout (Workout w, String name, String location, String bestTimes, Int duration, String materials, Int peopleCount, String difficulty, String status) public void submitSuggestion (String name, String location, String bestTimes, Int duration, String materials, Int peopleCount, String difficulty, String status)
ReviewUI	This is the interface of reviews available to users and administrators whenever one wishes to Add or Delete Reviews *Responsibilities*: public void submitReview(Int id, String rating, String body, Date dateTimePosted,Workout w) public void removeReview(Review r)

Control Classes:

Class Name	Description
MaintainAthleteController	This is the control that maintain athlete record. It is considered an abstract class.
AddAthleteController	This is the control that adds an athlete to the system. It extends MaintainAthleteController. *Responsibilities*: public void AddAthlete(Athelete a)
MaintainWorkoutController	This is the control that maintain workouts. It is considered an abstract class.
AddWorkoutController	This is the control that adds a workout to the system. It extends MaintainWorkoutController. *Responsibilities*: public void createWorkout(Workout w)
EditWorkoutController	This is the control that edits a workout from the system. It extends MaintainWorkoutController. *Responsibilities*: public void editWorkout(Workout w, String name, String location, String bestTimes, Int duration, String materials, Int peopleCount, String difficulty, String status)
DeleteWorkoutController	This is the control that adds a review to the system. It extends MaintainWorkoutController. *Responsibilities*: public void deleteWorkout(Workout w)
MaintainReviewController	This is the control that maintain reviews. It is considered an abstract class.
AddReviewController	This is the control that adds a review to the system. It extends MaintainReviewController. *Responsibilities*: public void createReview(Review r)
DeleteReviewController	This is the control that deletes a review from the system. It extends MaintainReviewController. *Responsibilities*: public void deleteReview(Review r)

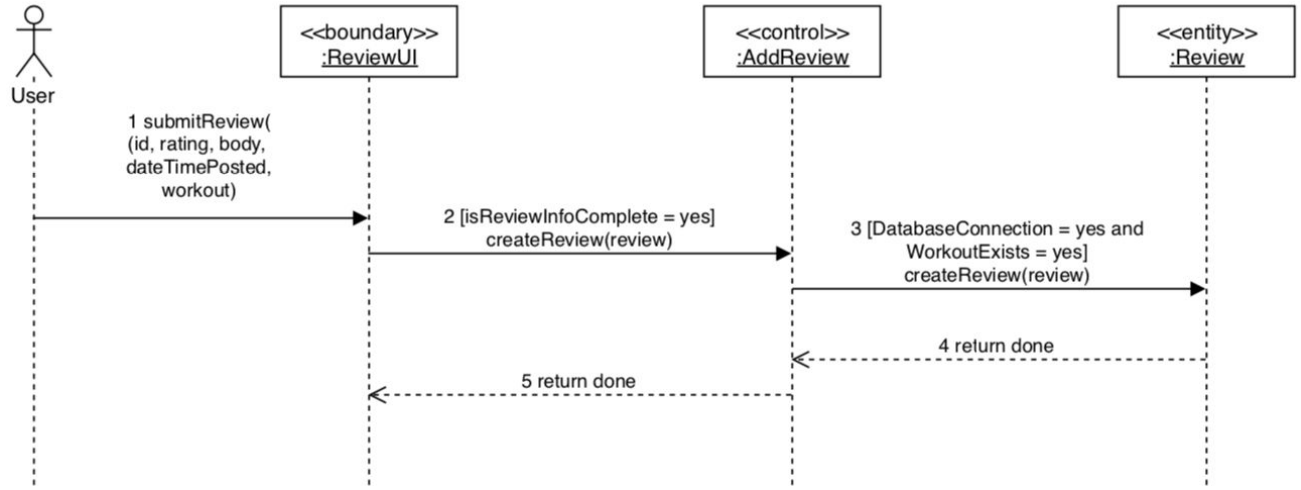
Entity Classes:

Class Name	Description
Athlete	This is the entity class athlete, which contains the data about the athlete. *Attributes*: private int athleteID private String lastname private String firstname private char gender = [M or F] … private String status; // provides the status code of the athlete eg. ACTIVE, SUSPENDED
Workout	This is the entity class athlete, which contains the data about the workout. *Attributes*: private String name private String location private String bestTimes private Int duration private String materials private Int peopleCount private String difficulty private String status // provides the status of the workout eg. APPROVED, PENDING *Methods*: public void createWorkout(Workout w) public void editWorkout(Workout oldw, Workout neww) public void deleteWorkout(Workout w)
Review	This is the entity class athlete, which contains the data about the review. *Attributes*: private Int id private String rating private String body private Date dateTimePosted *Methods*: public void createReview(Review r) public void deleteReview(Review r)

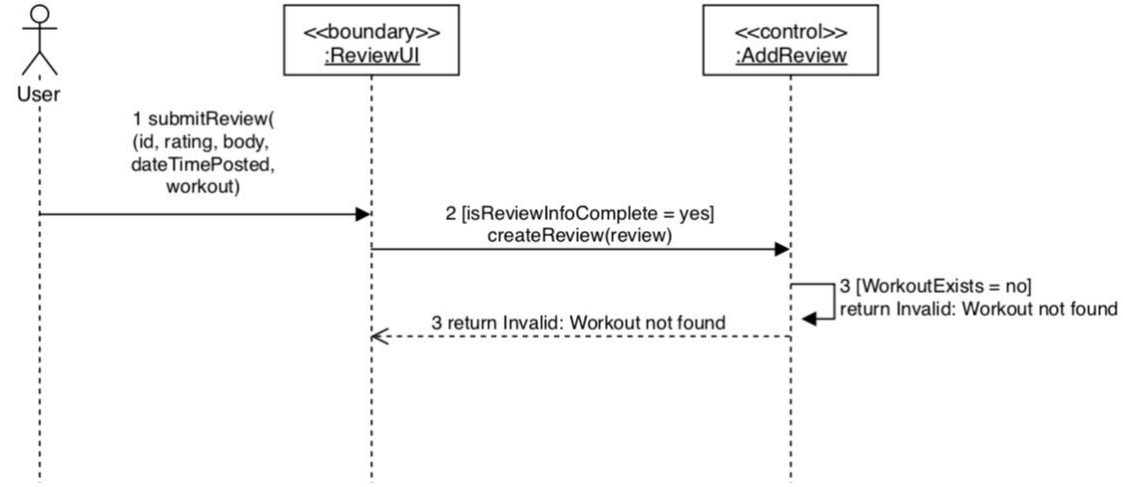
Behavioral Model:

***Use-Case Name*:** 1.1 Add Review
Description: The users and administrators can add reviews on particular workouts.

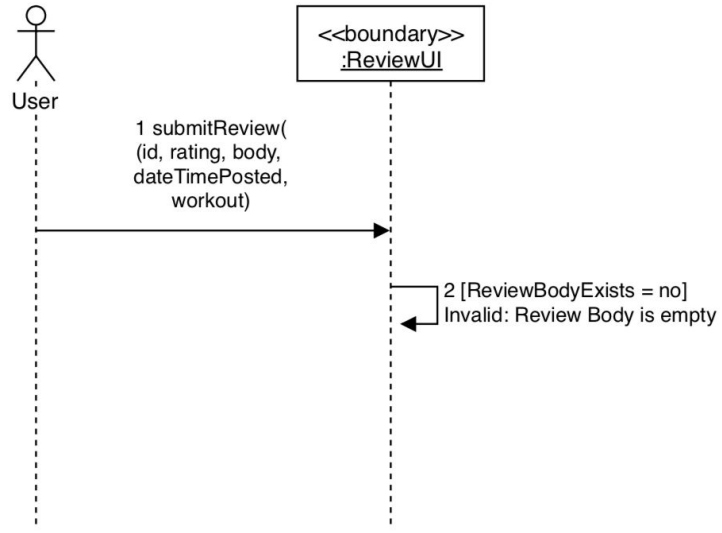
Scenario 1: Successful Add Review. (Basic Flow)



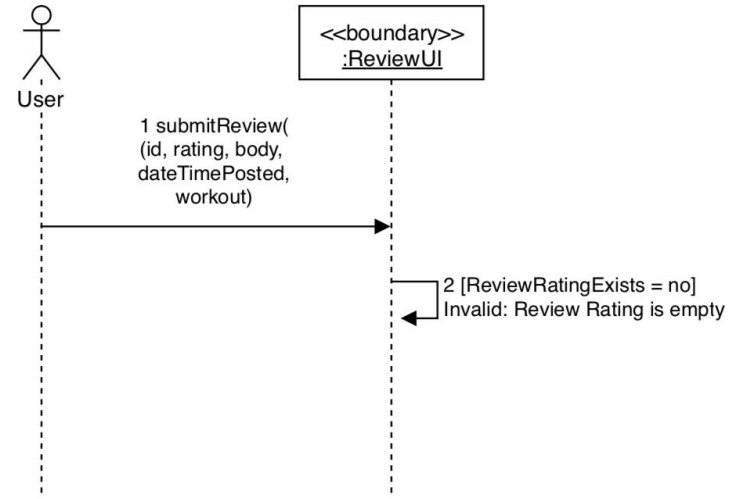
Scenario 2: Invalid: Workout does not exist.



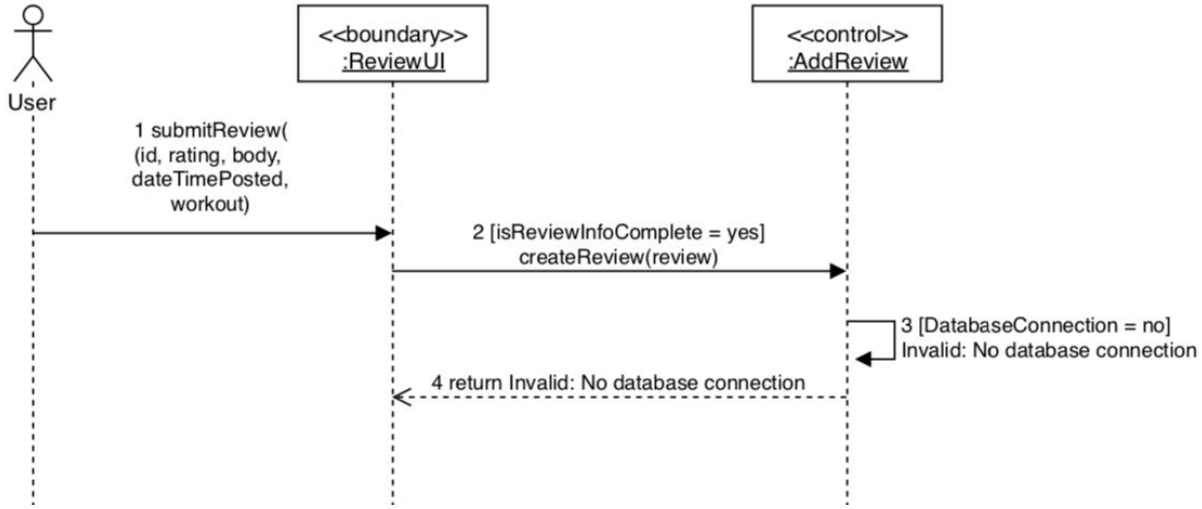
Scenario 3: Invalid: Review body is empty.



Scenario 4: Invalid: Review rating is empty.



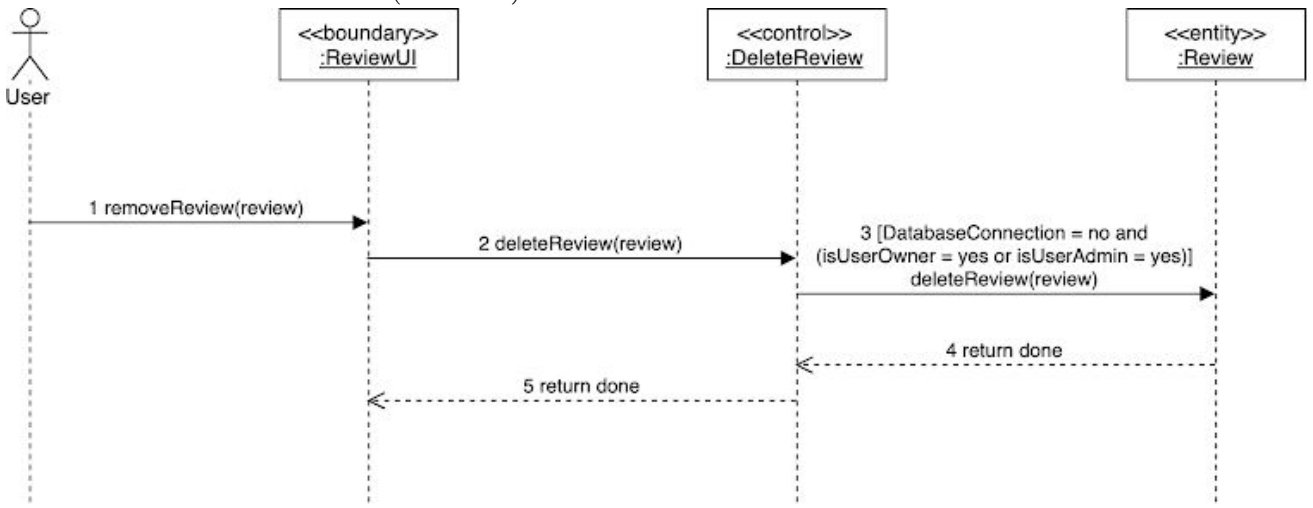
Scenario 5: No database connection.



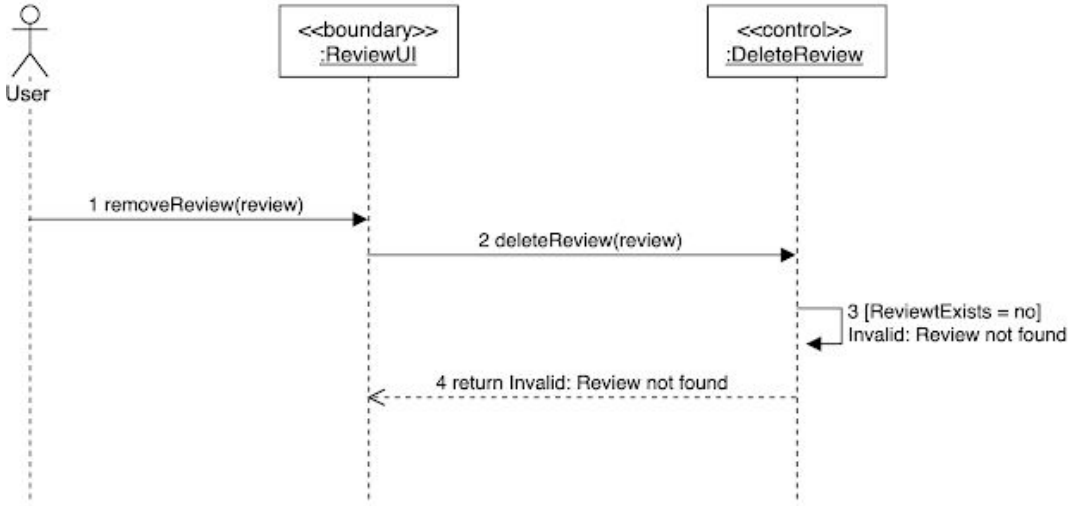
***Use-Case Name*:** 1.2 Delete Review

Description: Reviews can also be deleted, if necessary. Users may only delete their own review, while Administrators may delete any review.

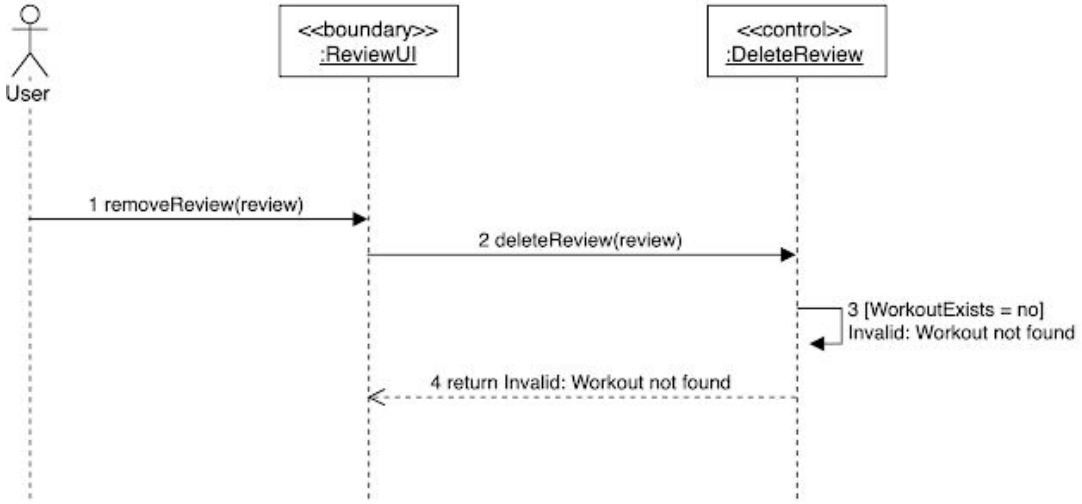
Scenario 1: Successful Delete Review (Basic Flow)



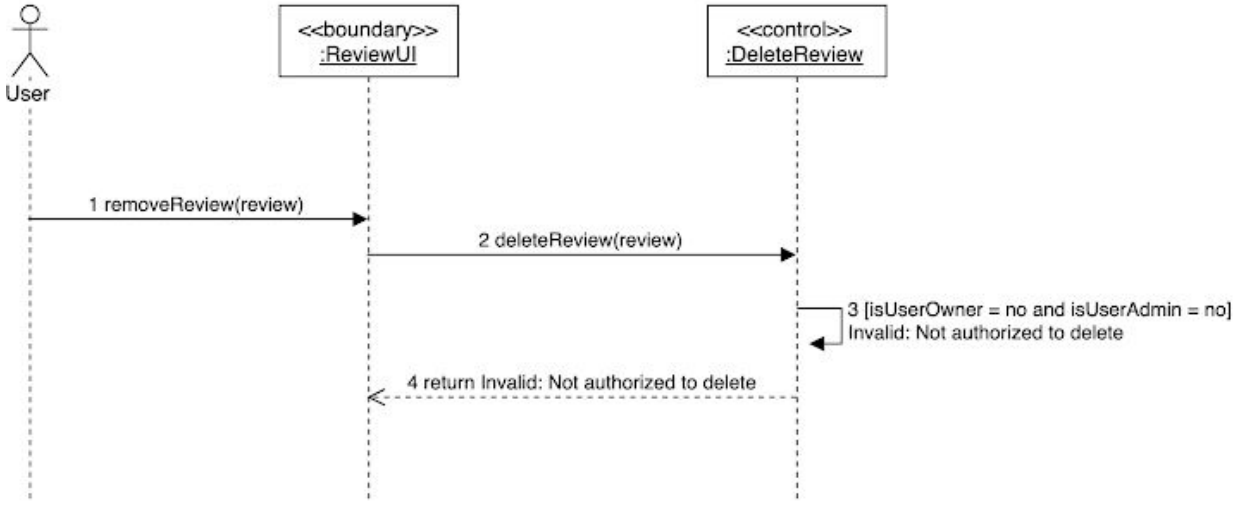
Scenario 2: Invalid: Review does not exist.



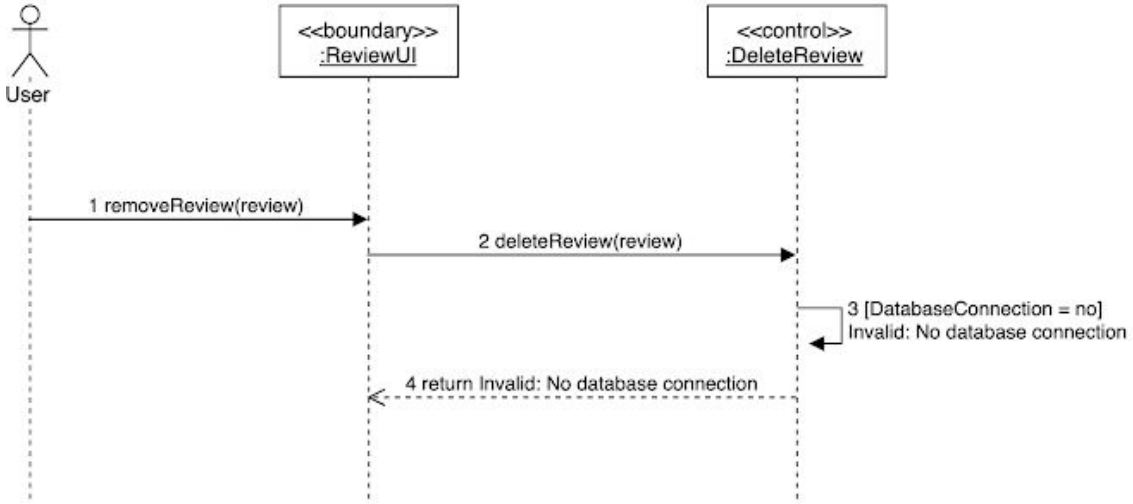
Scenario 3: Invalid: Workout does not exist.



Scenario 4: Invalid: Review is not User's own review.

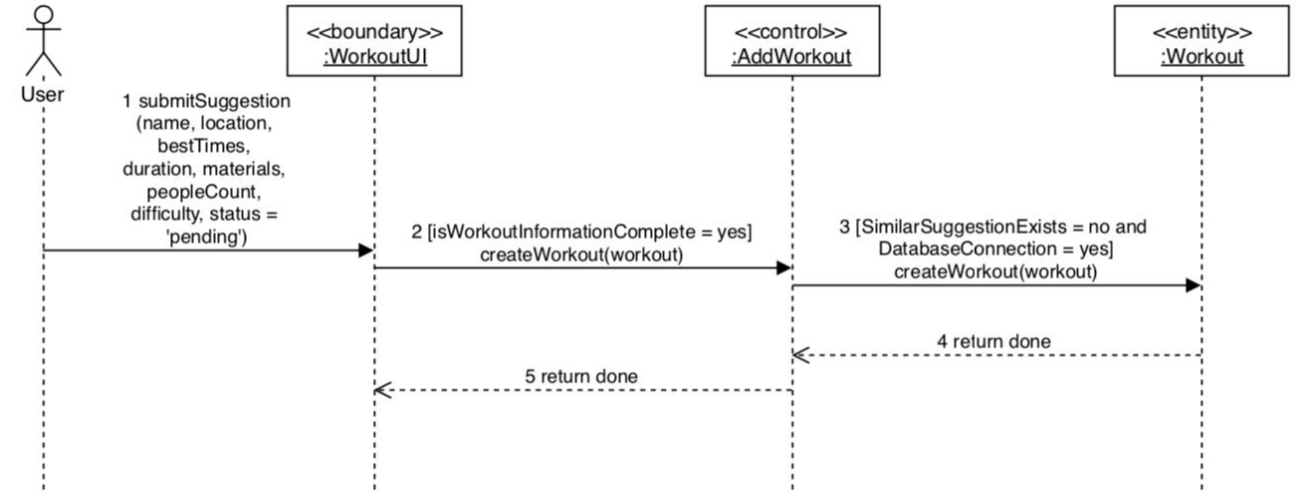


Scenario 5: No database connection.

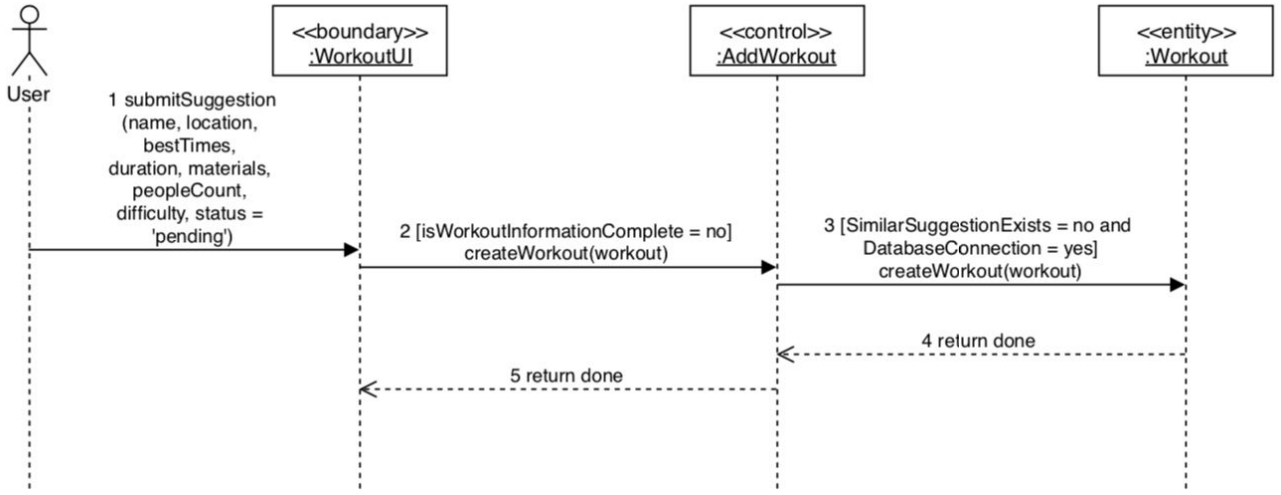


***Use-Case Name*:** 2.1 Add Workout Suggestion
Description: The users and administrators can add workouts, that have default status of pending.

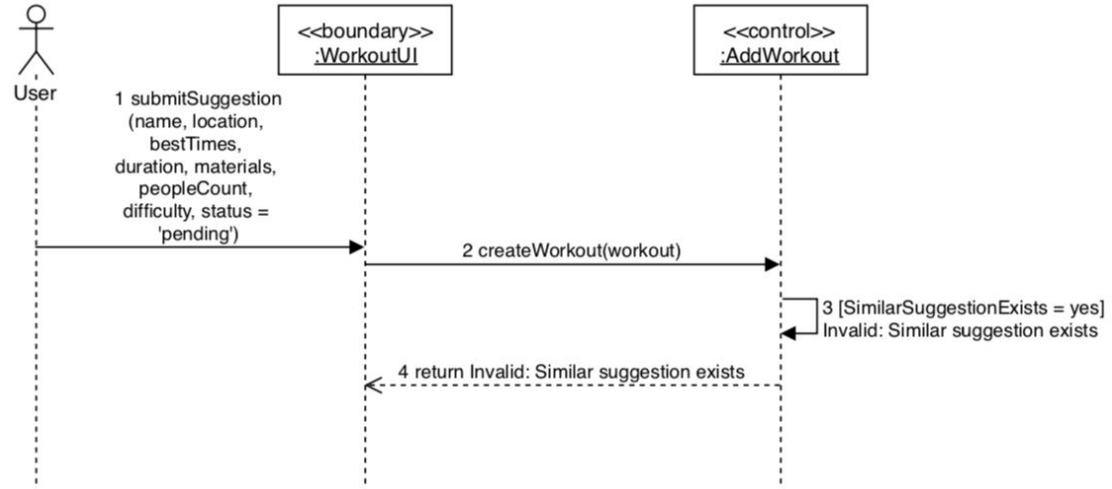
Scenario 1: Successful Add Workout Suggestion with complete information. (Basic Flow)



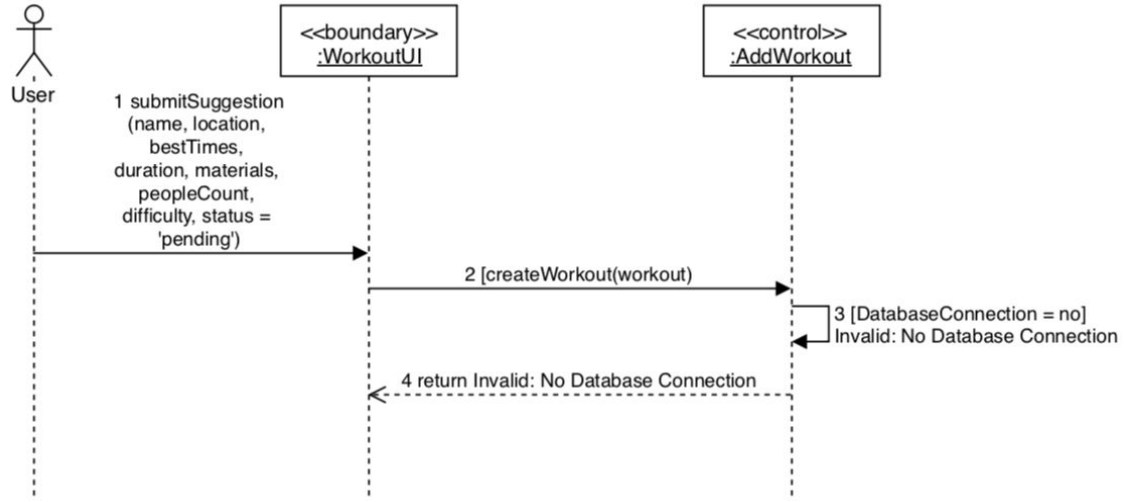
Scenario 2: Successful Add Workout Suggestion with incomplete information.



Scenario 3: Invalid: Similar suggestion exists (same information).



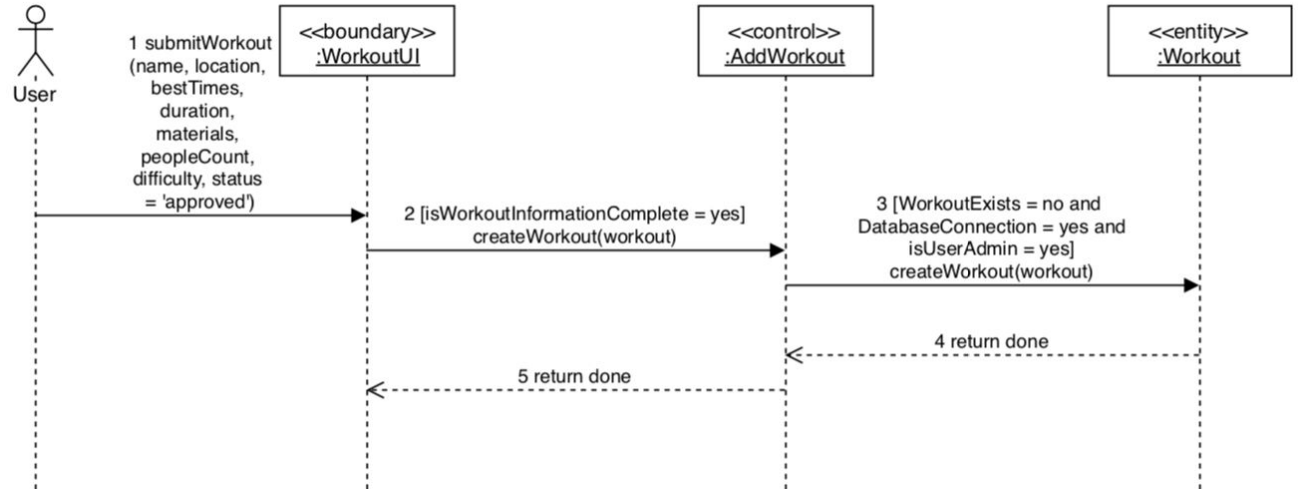
Scenario 4: No database connection.



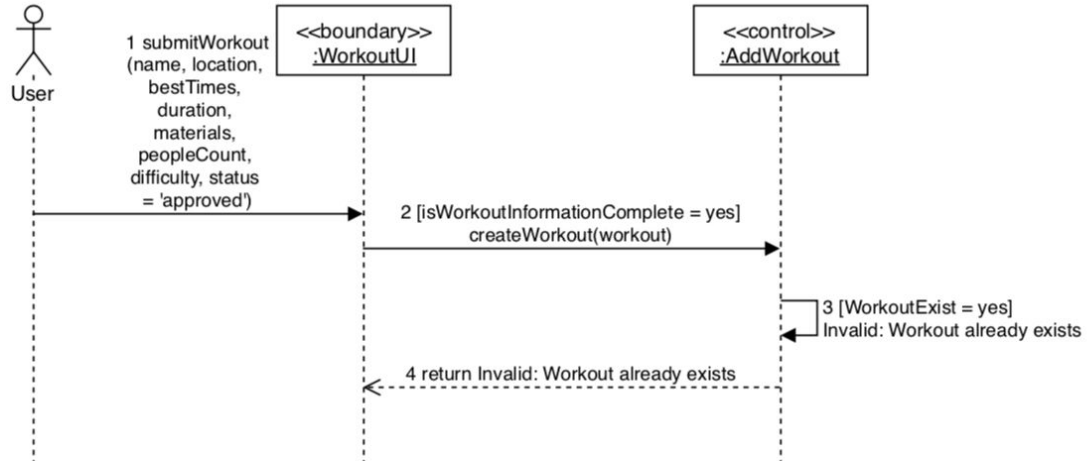
***Use-Case Name*:** 2.2 Add Workout

Description: Only Administrators can approve pending workouts by updating their status to approved. Administrators may also directly create a workout with status approved.

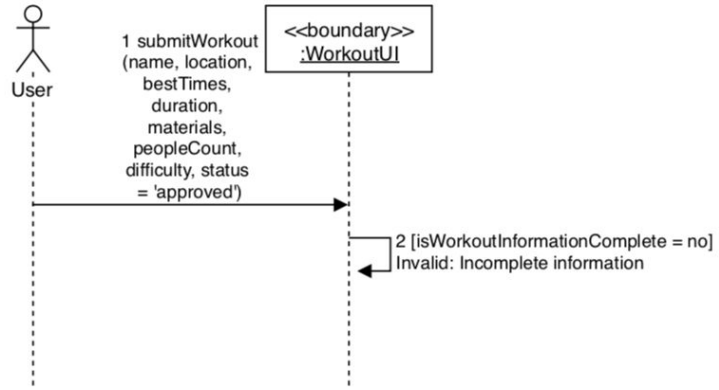
Scenario 1: Successful Add Workout. (Basic Flow)



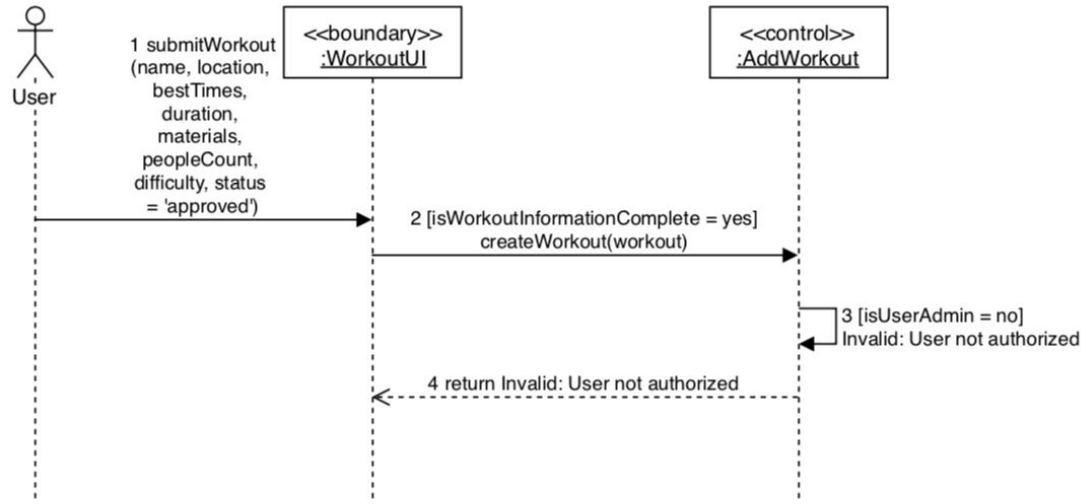
Scenario 2: Invalid: Workout already exists.



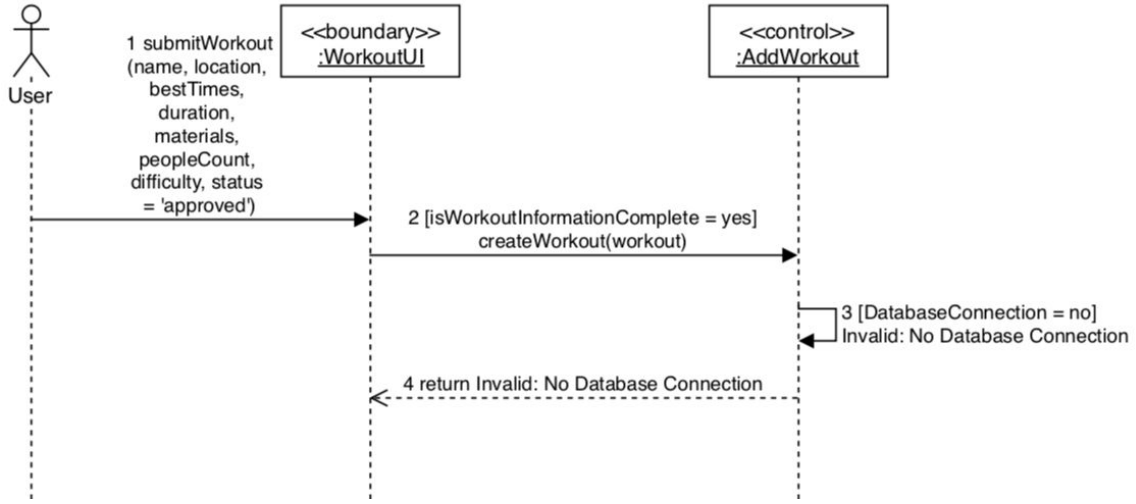
Scenario 3: Invalid: Information incomplete



Scenario 4: Invalid: User not authorized to add workout.

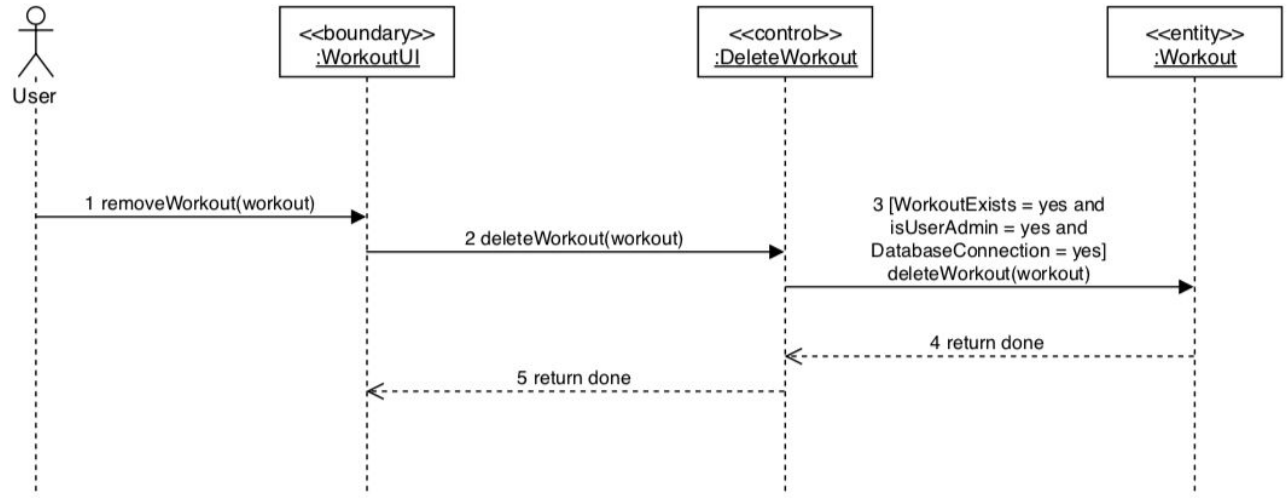


Scenario 5: No database connection.

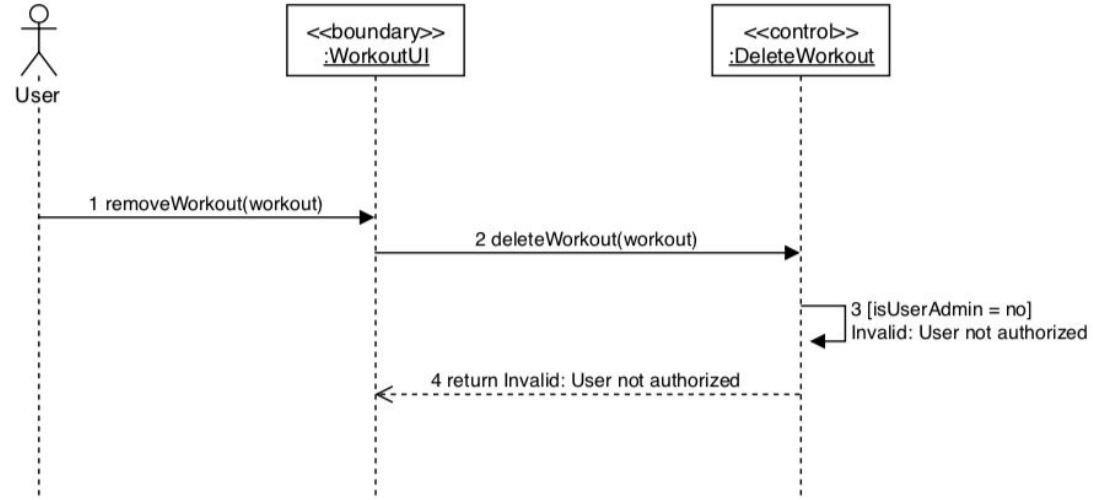


***Use-Case Name*:** 2.3 Delete Workout
Description: Administrators may delete existing workouts, when necessary.

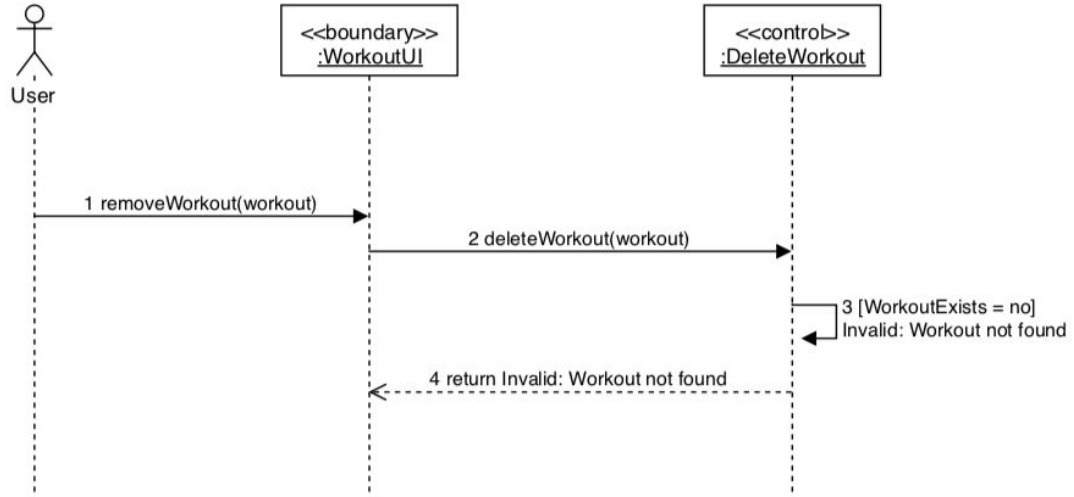
Scenario 1: Successful Delete Workout. (Basic Flow)



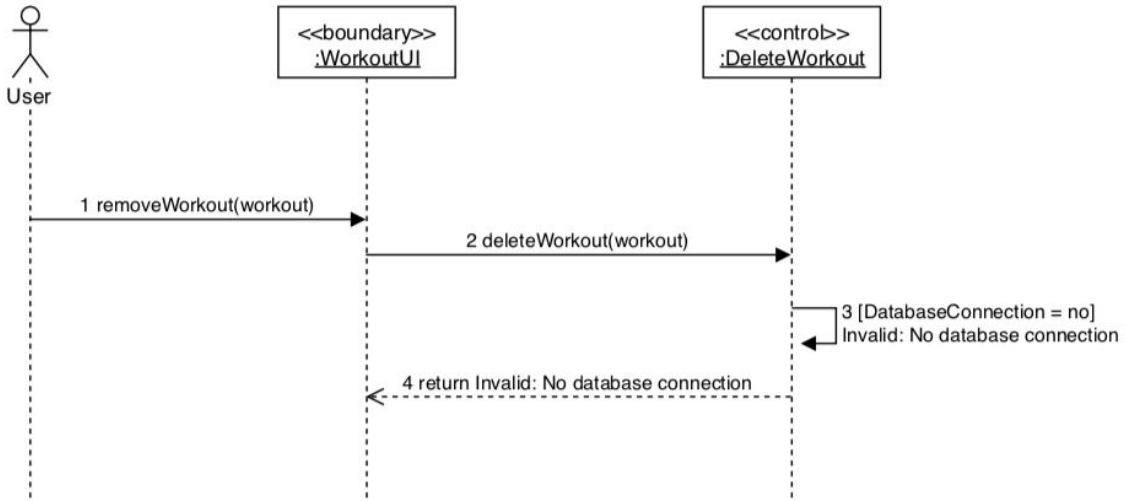
Scenario 2: Invalid: User not authorized to delete.



Scenario 3: Invalid: Workout does not exist.

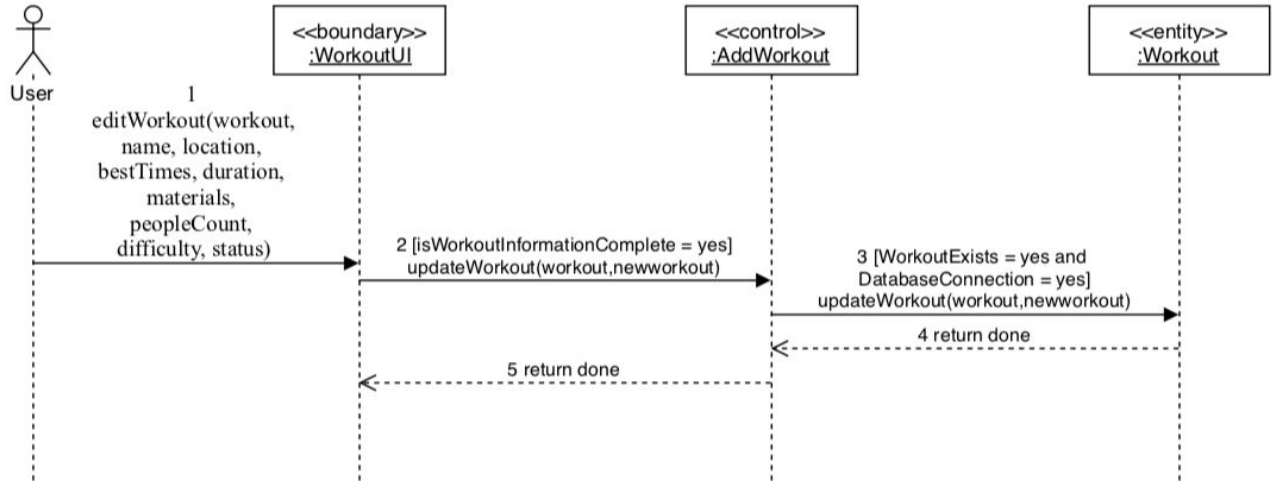


Scenario 4: No database connection.

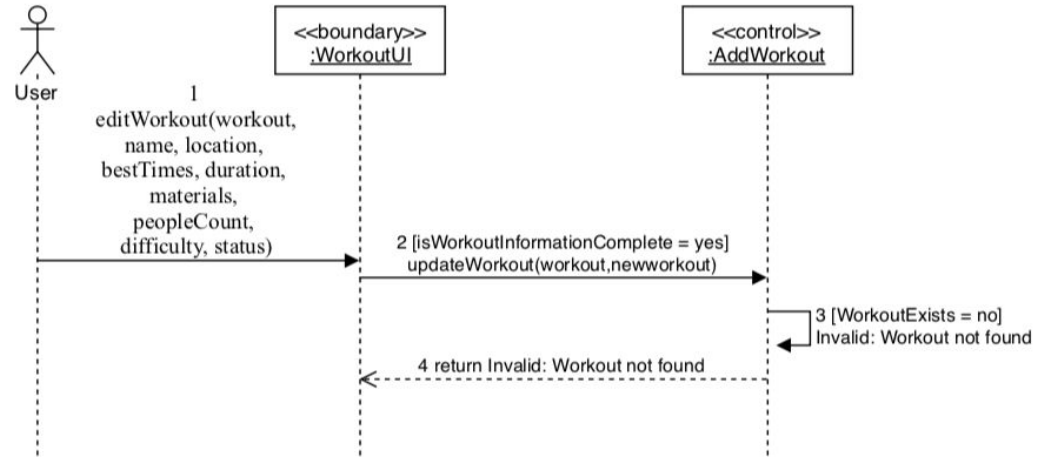


***Use-Case Name*:** 2.4 Edit Workout
Description: Administrators may update existing workouts, when necessary.

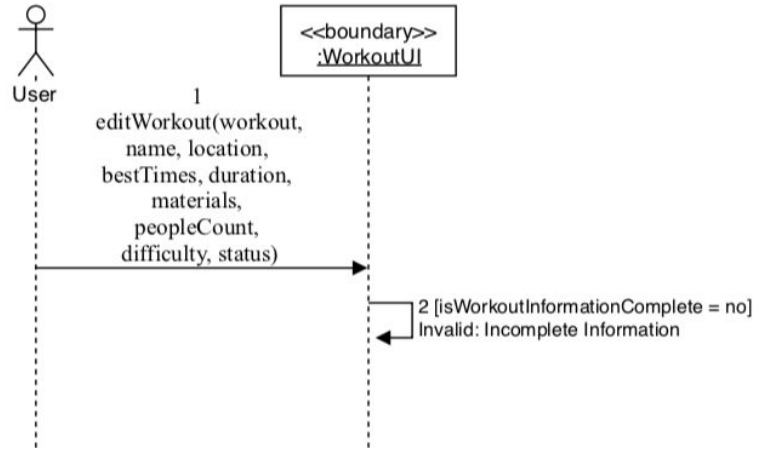
Scenario 1: Successful Edit Workout. (Basic Flow)



Scenario 2: Invalid: Workout does not exist.



Scenario 3: Invalid: Information is incomplete.



Scenario 4: No database connection.