

GetUP

Use Case Specifications

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



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/).

The documents are stored in the <https://github.com/gab51299/GetUP/tree/master/02-Requirements%20Engineering/Project%20Deliverables> referenced with GetUP-1-Maintain Reviews.pdf.

Document Purpose:

This document is intended to elaborate on the use case specifications for the GetUP project. Specifically, this document will be focusing on the “Maintain Reviews” use-case.

Target Audience:

UP Diliman community (students and faculty), particularly the class of CS 191 for the 1st semester of AY ‘19-’20.

Revision Control:

<i>Revision Date</i>	<i>Person Responsible</i>	<i>Version Number</i>	<i>Modification</i>
09/20/19	Jian Chan	1.0	Initial Document
09/20/19	Jian Chan	1.1	Document purpose, Target audience, Description, Use-case name, and Preconditions
09/23/19	Jian Chan	2.0	Add Flow of Events and Activity Diagram
09/23/19	Jian Chan	2.1	Add ER Diagram and list relationships

Use-Case Name: 1.0 Maintain Reviews

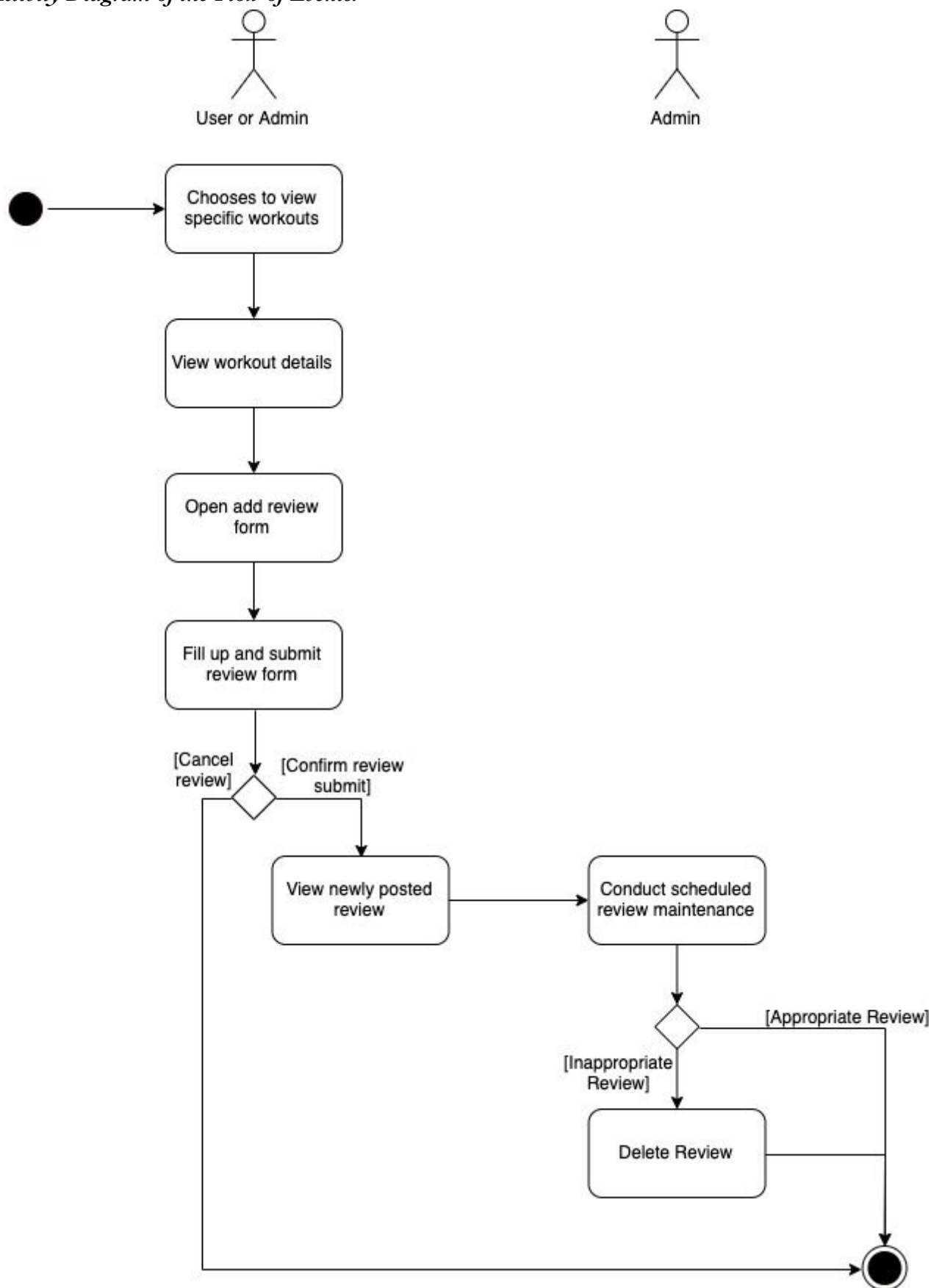
Description: The users and administrators can add reviews on particular workouts. At the same time, reviews can also be deleted, if necessary.

Preconditions: User or administrator is logged in

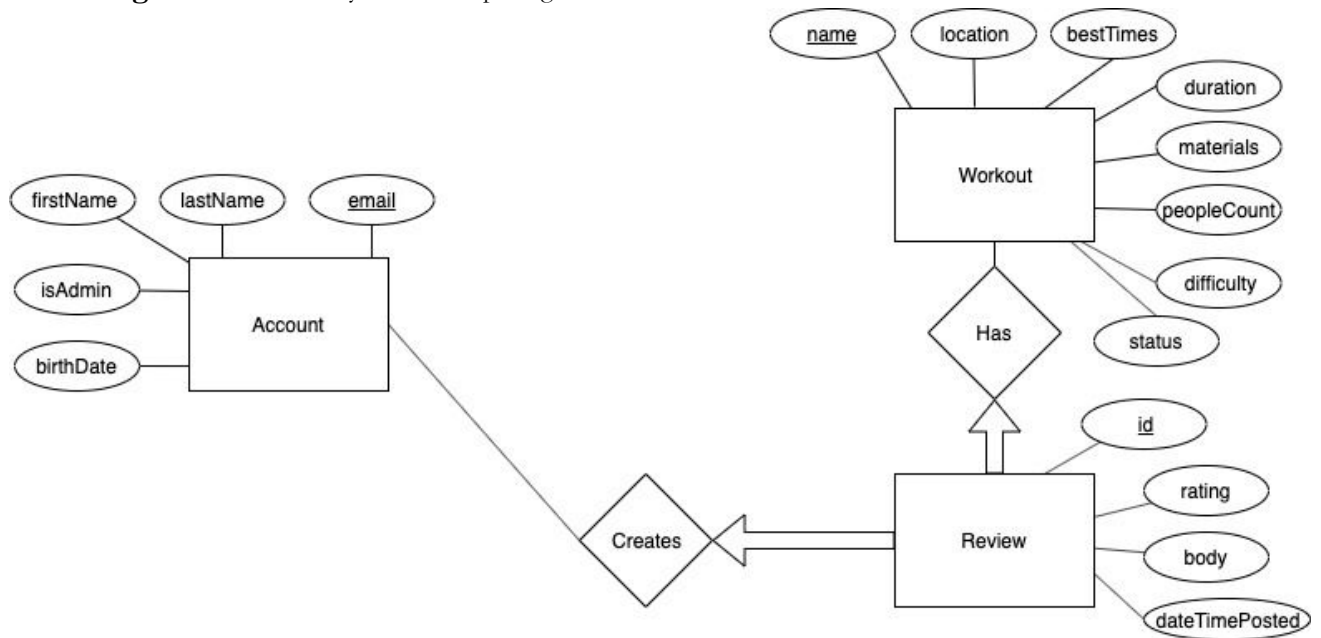
Flow of Events:

<i>Scenario Name</i>	<i>Description</i>
Scenario 1 (Basic Flow) User / Admin adds a review, which is later deemed appropriate by an Admin	<ol style="list-style-type: none">1. User/Admin opens the application and is viewing a specific workout.2. User/Admin views workout details on screen.3. User/Admin opens the add review form.4. User/Admin submits the completed form and confirms this submission.5. User/Admin can view newly posted review.6. Upon review management, if review is deemed appropriate, Admin leaves it as it is.
Scenario 2 User / Admin adds a review, which is later deemed inappropriate, and thus, deleted, by an Admin	<ol style="list-style-type: none">1. User/Admin opens the application and is viewing a specific workout.2. User/Admin views workout details on screen.3. User/Admin opens the add review form.4. User/Admin submits the completed form and confirms this submission.5. User/Admin can view newly posted review.6. Upon review management, if review is deemed inappropriate, Admin deletes this review..
Scenario 3 User / Admin opens the add review form, but does not push through with submission	<ol style="list-style-type: none">1. User/Admin opens the application and is viewing a specific workout.2. User/Admin views workout details on screen.3. User/Admin opens the add review form.4. User/Admin cancels submission of a review.

Activity Diagram of the Flow of Events:



Other Diagram: Entity-Relationship Diagram



Postcondition: NONE

Relationships: Has - Relationship between review and workout, each review must be under a single workout

Creates - Relationship between review and account, each review must be created by a single account

Special Requirements: NONE