

# Binge Buddy

## Use Case Specification

Submitted to:

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

Submitted by:

Jose, Giovan Samuel R.  
Pineda, Lance Dominic M.  
Sison, Abelardo III P.

In partial fulfillment of academic requirements  
for the course  
CS 191 Software Engineering I  
of the  
1<sup>st</sup> 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/).

***Unique Reference:***

The documents are stored in the <https://github.com/Riler4899/Cs-191-Show-Tracker> referenced with BingeBuddy-1.0-Maintain Watchlist.pdf.

***Document Purpose:***

The purpose of this document is to define the Use Cases and to identify the scenarios that occur within the specified use case.

***Target Audience:***

The target audience are users who wish to know each functions featured in the application being developed

***Revision Control:***

<b><i>Revision Date</i></b>	<b><i>Person Responsible</i></b>	<b><i>Version Number</i></b>	<b><i>Modification</i></b>
09/21/2019	Jose, Giovan Samuel R.	1.0	Prepared Initial Document with all the required details except for the ER diagram
09/23/2019	Jose, Giovan Samuel R.	1.1	Added ER diagram. Updated description.

**Use-Case Name:** 1.0 Maintain Watchlist

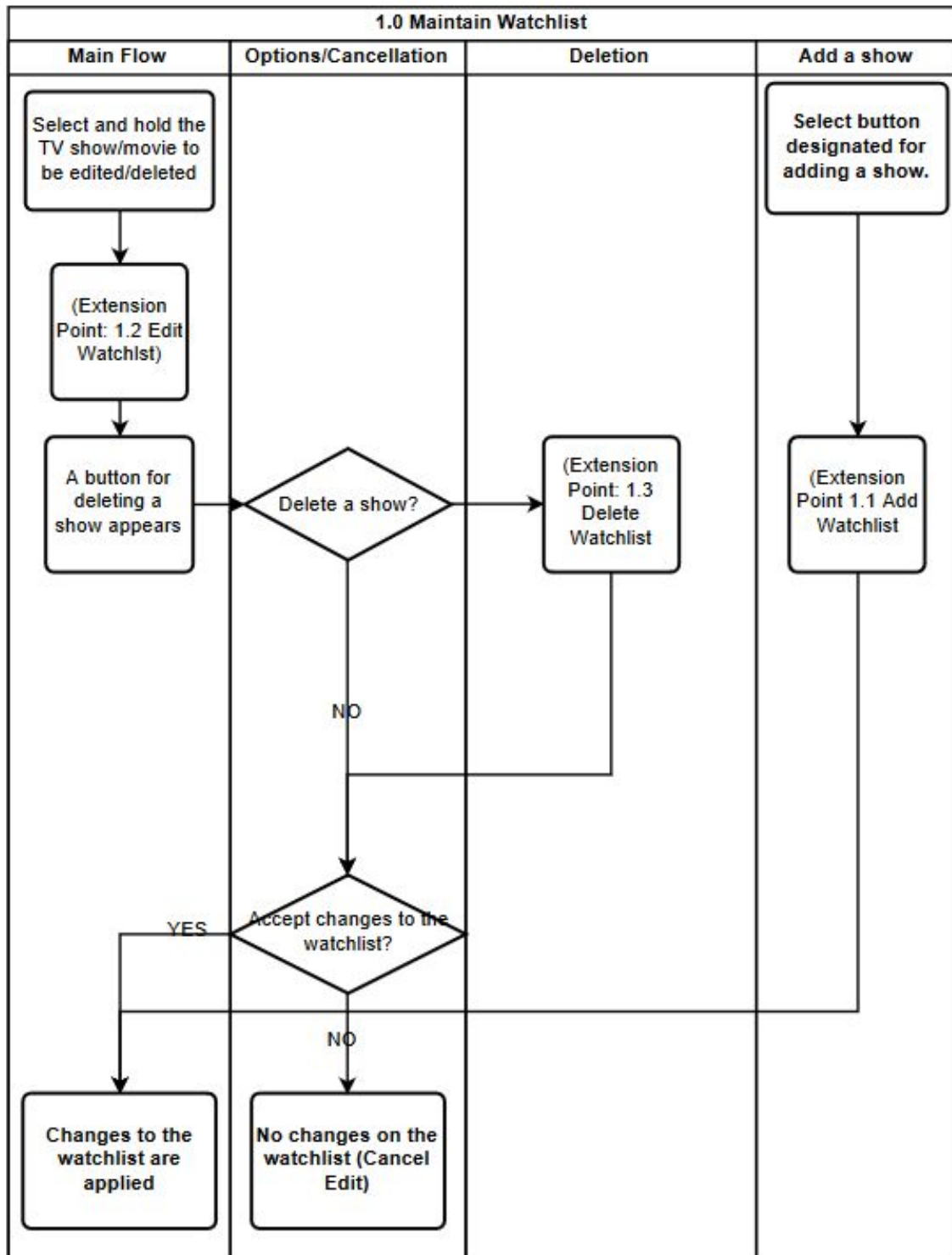
**Description:** This Use Case allows the user to access functions which can apply various modifications or actions to the watchlist. There are 3 scenarios that will be defined below. Each makes use of the extended use-cases which has different functions.

**Preconditions:** NONE

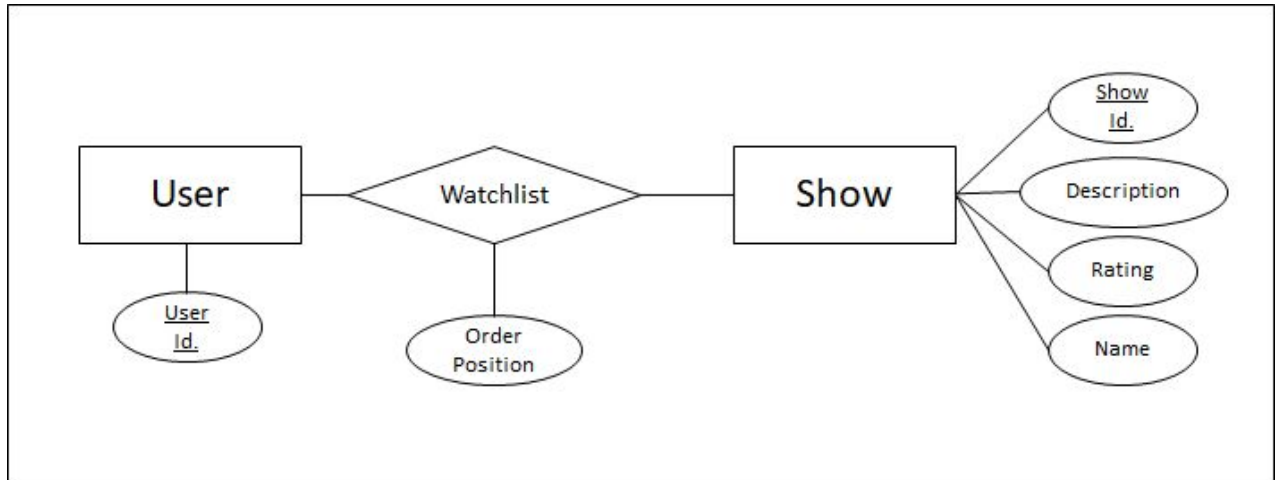
**Flow of Events:**

<i>Scenario Name</i>	<i>Description</i>
Scenario 1 (Basic Flow) User edits the watchlist.	1. User selects and hold the desired TV show/movie to be edited 2. (Extension Point: 1.2 Edit Watchlist) 3. A button for deleting a show appears. 4. User is prompted if we wants to accept the changes to the watchlist. 5. If User accepts, then apply changes to the watchlist, else do nothing to the watchlist.
Scenario 2 User adds a show to the watchlist.	1. User selects a button designated for adding a show. 2. (Extension Point 1.1 Add Watchlist) 3. Changes to the watchlist are applied.
Scenario 2 User deletes a show in the watchlist.	1. User selects and hold the desired TV show/movie to be deleted 2. (Extension Point: 1.2 Edit Watchlist) 3. A button for deleting a show appears. 4. User selects the button for deleting a show. 4. (Extension Point: 1.3 Delete Watchlist) 5. User is prompted if we wants to accept the changes to the watchlist. 6. If User accepts, then apply changes to the watchlist, else do nothing to the watchlist.

*Activity Diagram of the Flow of Events:*



**Other Diagram: (Entity-Relationship Diagram)**



**Postcondition:** NONE

**Relationships:** NONE

**Special Requirements:**  
NONE