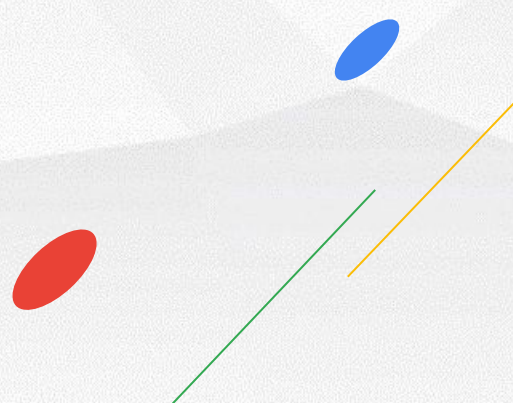


Spring2020-A5.1



Group 16

Jiaqiu Shen,
Yuhao Yang,
Huadong Hu,
Yufei Li





CONTENTS

1

Use Case Diagram

2

Use Case Specification

3

Bi-direction traceability Matrix

4

Sequence Diagram

5

Class Diagram Model

6

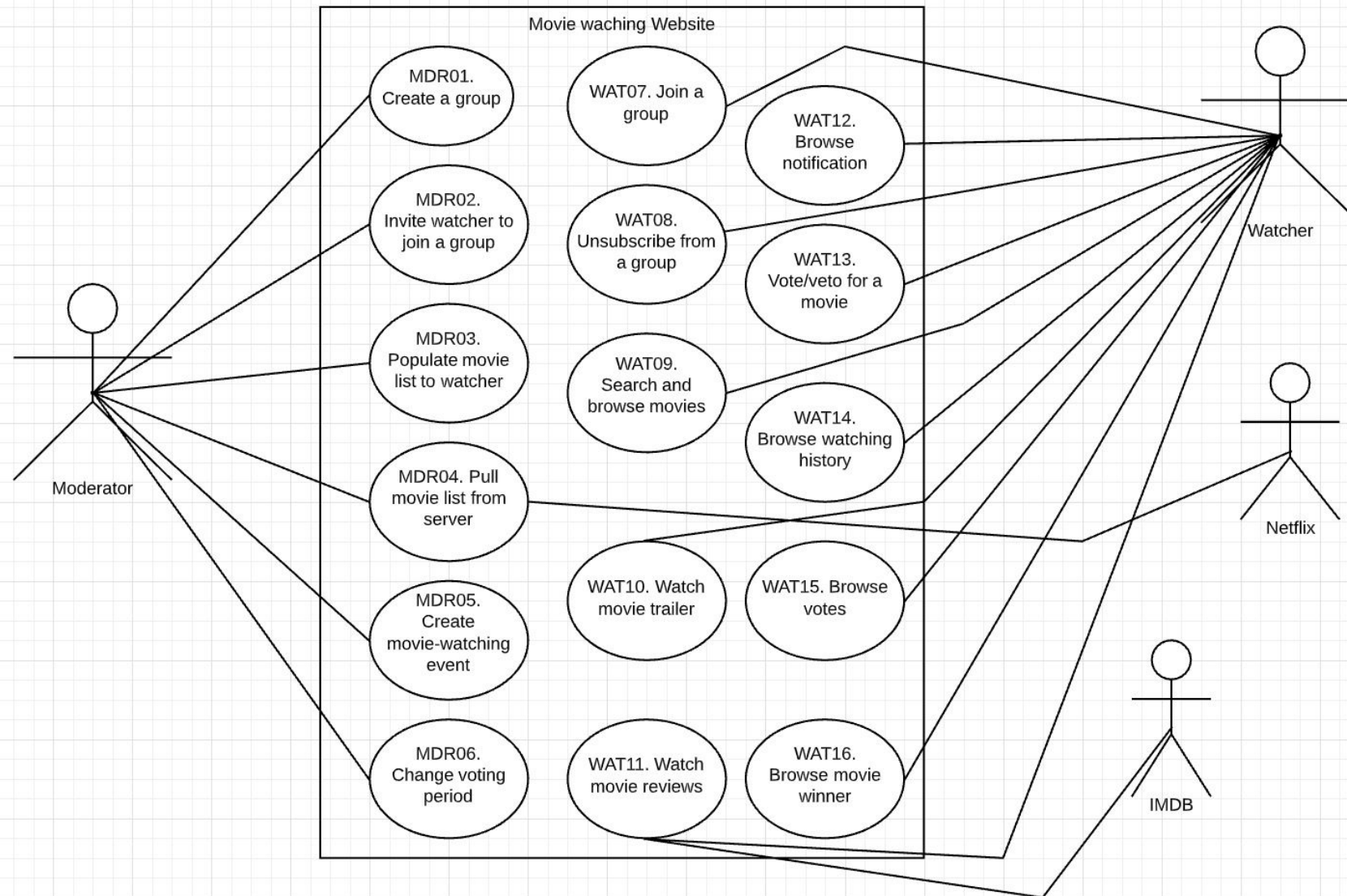
State Diagram



Use Case Diagram



Use Case Diagram





Use Case Specification



Use Case Specification

Use Case ID: MDR01

Title: Create a movie watcher group

Description: Moderator create a movie watcher group

Pre-Conditions: have an account; successfully logged in

Post-Conditions:

Steps: Moderator add group information in the create page -> create

Use Case ID: MDR02

Title: Invite watcher to join the group

Description: Moderator inviting watcher to join the group

Pre-Conditions: A group with the moderator exists

Post-Conditions: Watcher is in the group

Steps: 1. Moderator chooses the watchers to be invited 2. Moderator clicks the “invite” button 3. System sends the notification to the watcher 4. Watcher receives the invitation 5. Watcher either clicks accept or refuse 6. If accepts, watcher is in the group list; if refuse, the system notifies the moderator

Use Case ID: MDR03

Title: Populate movie list to watcher

Description: Moderator populate movie watcher’s groups with a list of potential movies that could be watched

Pre-Conditions: the moderator has a group

Post-Conditions: Watcher has a list of movies to watch

Steps: 1. Moderator selects a list of movies 2. Moderator makes the list of movies available to the watchers in the group 3. System adds the list of movies to the watcher’s homepage 4. Watchers can see a list of movies in their homepage

Use Case Specification

Use Case ID: MDR04

Title: Pull a movie list from a movie list server/API

Description: Moderator pull a movie list from a movie list server/API

Pre-Conditions: the moderator has a movie list, at least one server is available

Post-Conditions: A list of movies exists in the movie watching app

Steps: 1. Moderator connects to the movie server 2. Moderator sends a request to pull a movie list from the server 3. The movie server receives the request 4. The movie server sends the movie list to the system 4. System stores the movie list from the server

Use Case ID: MDR05

Title: Create movie-watching event

Description: Moderator create a movie-watching event

Pre-Conditions: The moderator has at least one group

Post-Conditions: The moderator and group members have the new event.

Steps: 1. Moderator clicks the “create movie event” button 2. System receives the request 3. System displays the creation page 4. Moderator types in the information and clicks submit 5. System receives the information 6. System checks duplicate, if no then next step, otherwise displays error and back to step 3 7. System displays new movie watching event homepage

Use Case ID: MDR06

Title: Change voting period

Description: Moderator open and close voting period for a specific movie event

Pre-Conditions: Moderator has at least one movie event

Post-Conditions: The voting period of the event is updated

Steps: 1. Moderator browse the list of the events he/she created 2. Moderator selects one event 3. Moderator clicks the “change the voting period” button 4. System receives the request 5. System displays change page 6. Moderator types in the new voting period and clicks “submit” button 7. System receives the information 8. System displays the new voting period

Use Case Specification

Use Case ID: WAT07

Title: Join group

Description: Watcher join a group through an invitation

Pre-Conditions: Watcher has an invitation

Post-Conditions: Watcher has a new group

Steps: 1. Watcher browse the list of the invitations he/she received 2. Watcher selects the group he wants to join 3. Watcher click the “accept” button 4. System receives the request 5. System adds the watcher to the group 6. Watcher is in the group’s member list

Use Case ID: WAT08

Title: Unsubscribe from a group

Description: Watcher unsubscribe a specific group

Pre-Conditions: Watcher is in a group

Post-Conditions: Modify the user’s group list (delete the specific group)

Steps: 1. Watcher browse the list of the groups he/she subscribed 2. Watcher chooses the group he wants to unsubscribe from 3. Watcher clicks the “unsubscribe” button 4. System receives the request 5. Systems remove the watcher from the group

Use Case ID: WAT09

Title: Search and browse movies

Description: Users browse movies provided by the moderator and click to see details.

Pre-Condition: Movie list should not be empty

Post-Condition: N/A

Steps: 1. Watcher types in the name of the movie in the search bar 2. Watcher clicks the “search” button 3. System receives the request 4. System displays a list of movies related to the search 5. Watcher sees a list of movies on his page

Use Case Specification

Use Case ID: WAT10

Title: Watch movie trailer

Description: Users watch movie trailer

Pre-Condition: Movie trailer is not empty

Post-Condition: N/A

Steps: 1. Watcher navigates to a specific movie 2. Watcher clicks the “trailer” bottom 3. System receives the requests 4. System plays the movie trailer 5. Watcher sees the movie trailer on his page

Use Case ID: WAT11

Title: Watch movie reviews

Description: User browses other people’s reviews regarding this movie.

Pre-Condition: Movie reviews exist

Post-Condition: N/A

Steps: 1. Watcher navigates to a specific movie 2. Watcher clicks the “review” button 3. System receives the request 4. System displays the movie reviews 5. Watcher sees the movie reviews on his page

Use Case ID: WAT12

Title: Browse notification

Description: Notify users of movie-watching event.

Pre-Condition: A movie should not be voted by same user. Movies should come from the list populated by the moderator.

Post-Condition: A movie can only be voted once by same user

Steps: 1. Watcher clicks the “notification” button 2. System receives the request 3. System displays the notifications the watcher has received 4. Watcher sees a list of notifications on his page

Use Case Specification

Use Case ID: WAT13

Title: Vote and veto for movies

Description: watchers cast a vote to a movie

Pre: Watcher in movie-watching event, moderator opened movie watching period

Post: movies have their own votes

Steps: 1. Watcher navigates to the specific movie her wants to vote 2. Watcher clicks the thumb up or thump down button to vote 3. System receives the request 4. System updates the current voting of the movie

Use Case ID: WAT14

Title: Browse movie watching history

Description: watchers can browse watching history

Pre: Watchers watch movies

Post: N/A

Steps: 1. Watcher clicks the “watching history” button 2. System receives the request 3. System displays the watcher’s watching history 4. Watcher see his own movie watching history on his page

Use Case ID: WAT15

Title: Browse votes

Description: watchers can browse votes

Pre: watchers can cast vote

Post: N/A

Steps: 1. Watcher navigates to a specific movie 2. Watcher can see the votes next to the movie title

Use Case Specification

Use Case ID: WAT16

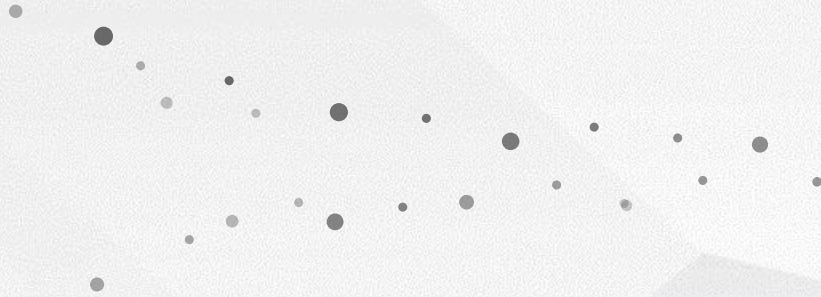
Title: Browse winners

Description: watchers browse winners in a movie watching event

Pre: winners elected

Post: N/A

Step: 1. Watcher chooses a movie watching event 2. The winners are displayed in the movie watching event





Bi-direction Traceability Matrix



Bi-direction Traceability Matrix

Requirement Traceability Matrix (RTM) is a document that maps and traces user requirement with test cases.

The main purpose of Requirement Traceability Matrix is to validate that all requirements are checked via test cases such that no functionality is unchecked during Software testing.

There are 3 types of RTM:

1. Forward: Requirements->Test Cases
2. Backward: Test Cases->Requirements
3. Bi-directional: Forward+Backward

/// Bi-direction Traceability Matrix

Test Case ID	Test Case Description
TC01	Verify if moderator is able to create a new movie watcher group
TC02	Verify if moderator is able to invite watcher to join the group
TC03	Verify if moderator is able to pull a movie list from IMDB/Douban Verify if moderator is able to populate a movie list to watcher
TC04	Verify if moderator is able to create a movie-watching event
TC05	Verify if moderator is able to change voting period
TC06	Verify if watcher is able to join a group
TC07	Verify if watcher is able to unsubscribe from a group
TC08	Verify if watcher is able to search and browse movies/trailers/reviews
TC09	Verify if watcher is able to browse notification
TC10	Verify if watcher is able to vote and veto for movies
TC11	Verify if watcher is able to browse watching history Verify if watcher is able to browse votes history Verify if watcher is able to browse winners history

/// Bi-direction Traceability

	Group Movie APP		
Requirement ID	Required Features	Requirement Use Case ID	Test Case ID
1	Moderator creates a movie watcher group	MDR01	TC01
2	Moderator invites watchers to join a group	MDR02	TC02
3	Moderator populates a movie list from IMBD/Douban to group	MDR03, MDR04	TC03
4	Moderator creates a movie-watching event	MDR05	TC04
5	Moderator changes voting period	MDR06	TC05
6	Watchers join a group	WAT07	TC06
7	Watcher unsubscribes form a group	WAT08	TC07
8	Watcher searches and browses movies/ trailers/reviews	WAT09, WAT10, WAT11	TC08
9	Watcher browses notification	WAT12	TC09
10	Watcher votes for movies	WAT13	TC10
11	Watcher browse watching/ votes/ winners history	WAT14, WAT15, WAT16	TC11



04

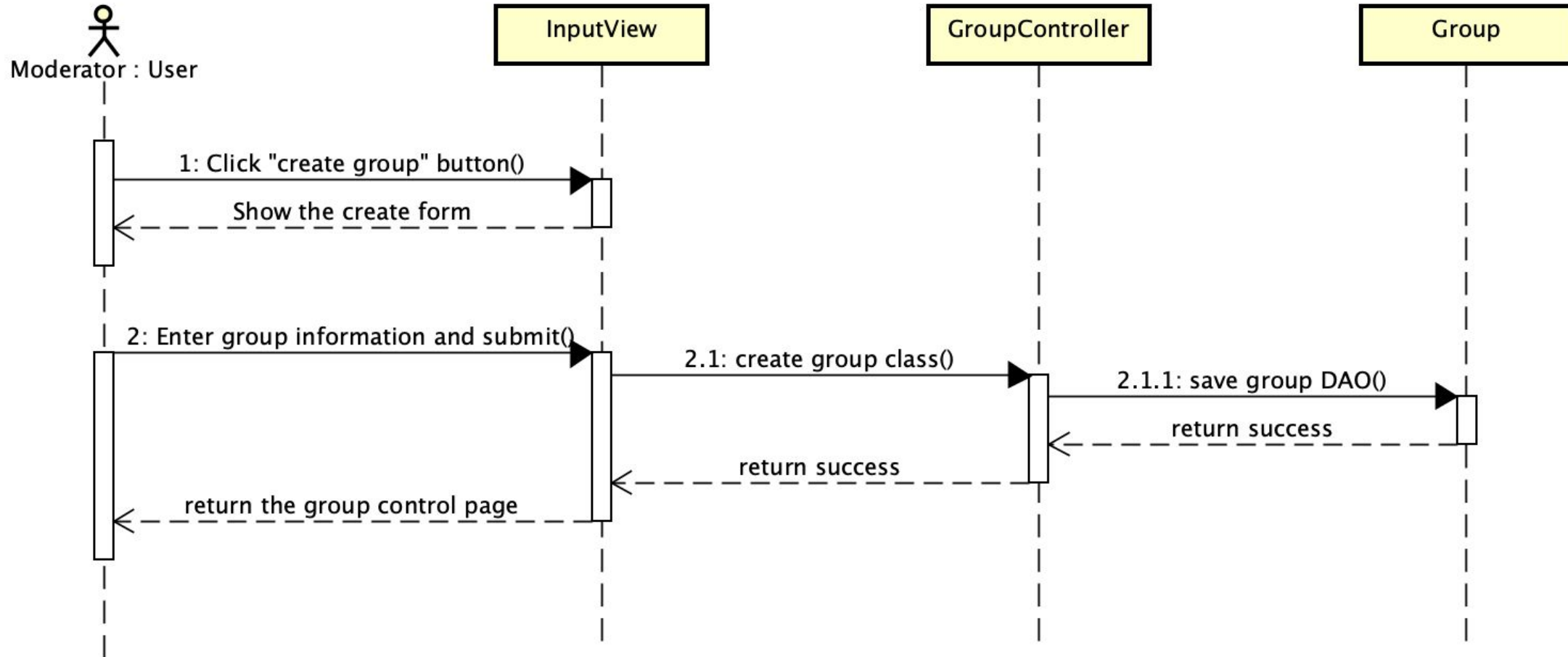
Part Four

Sequence Diagram

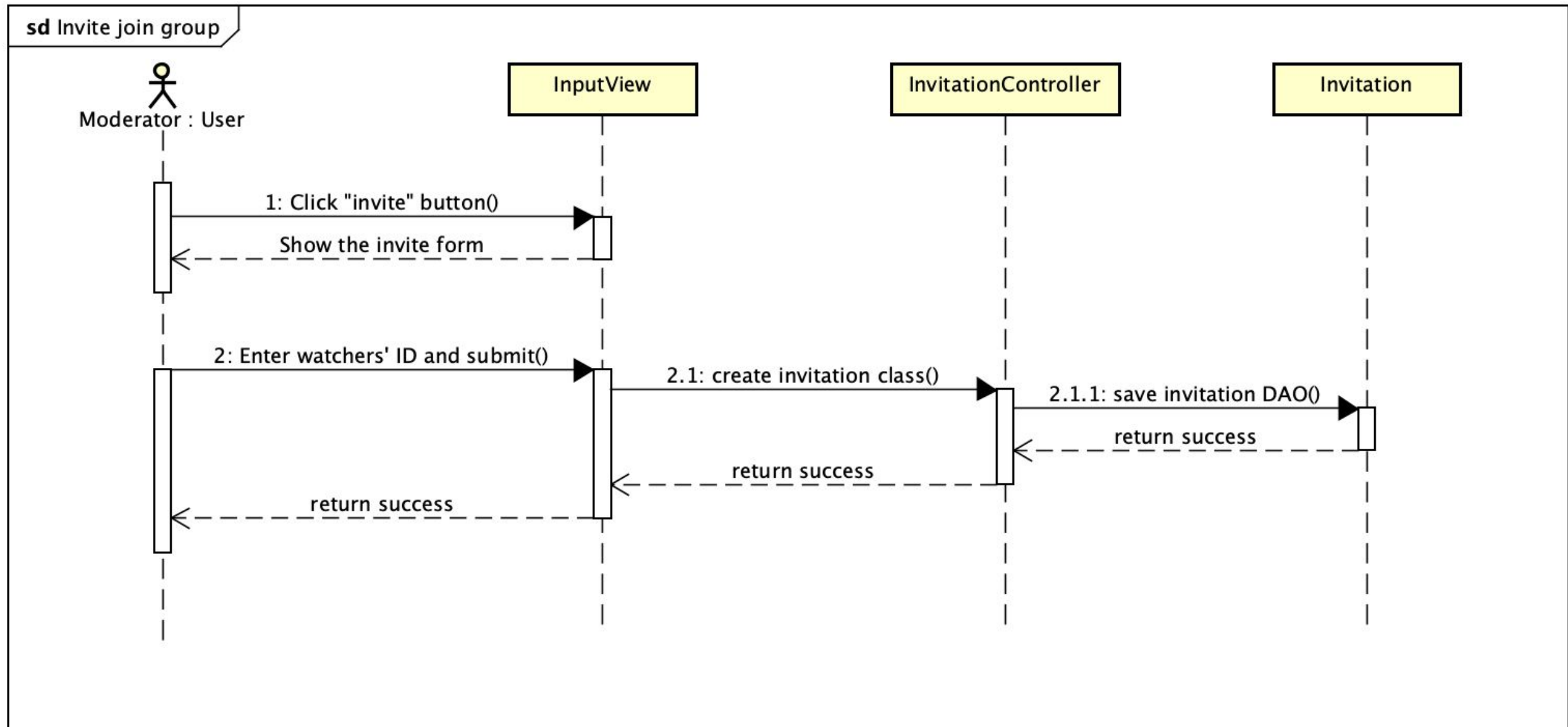


Sequence Diagram

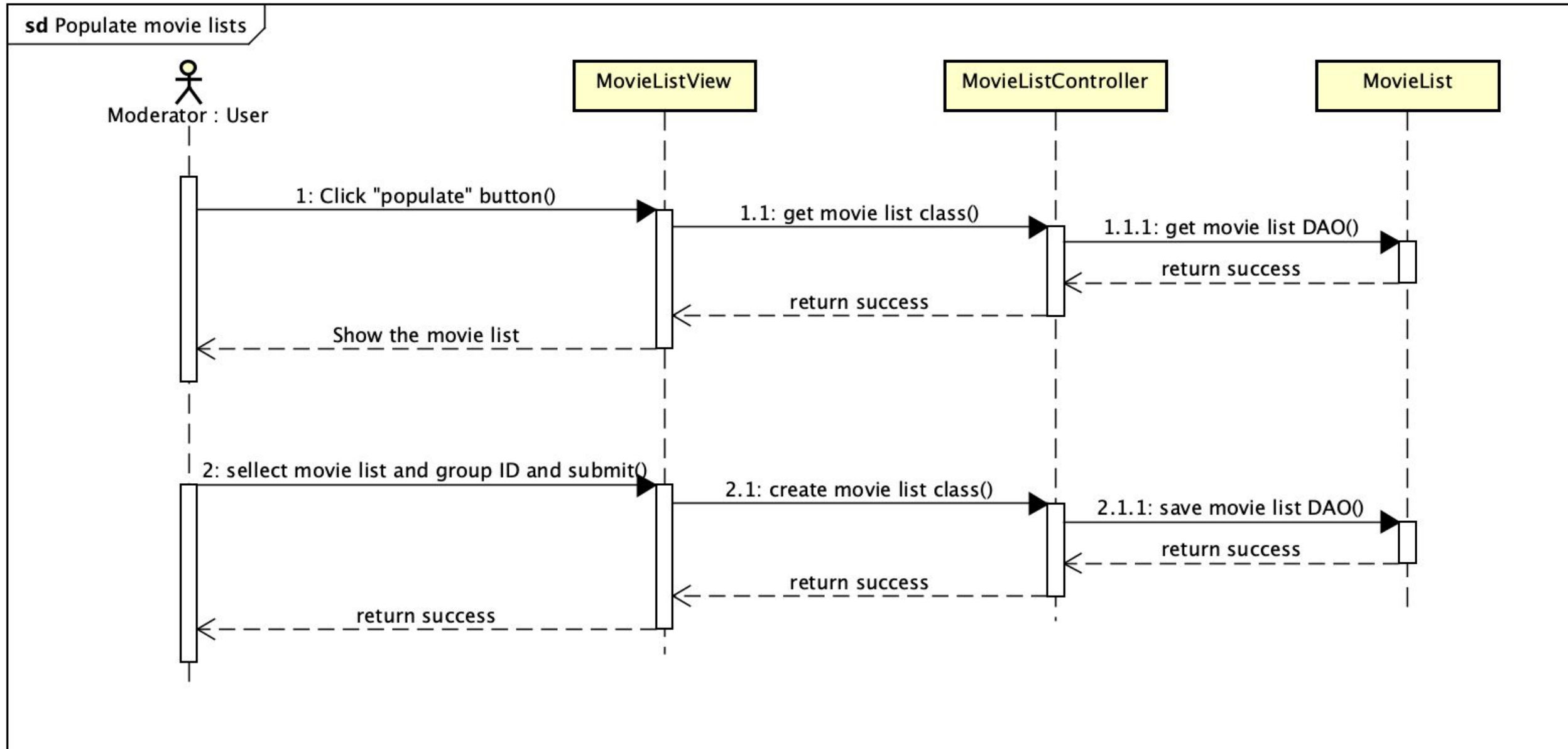
sd Create Group



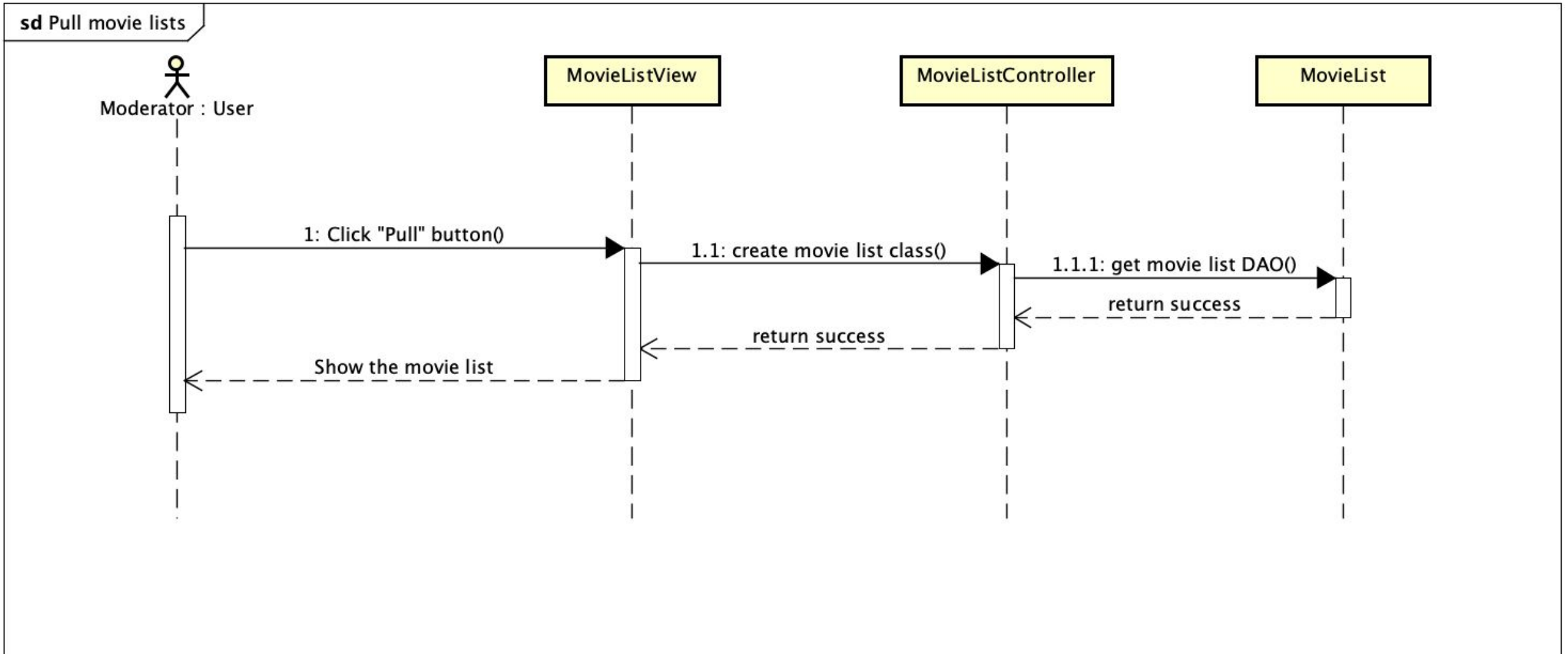
Sequence Diagram



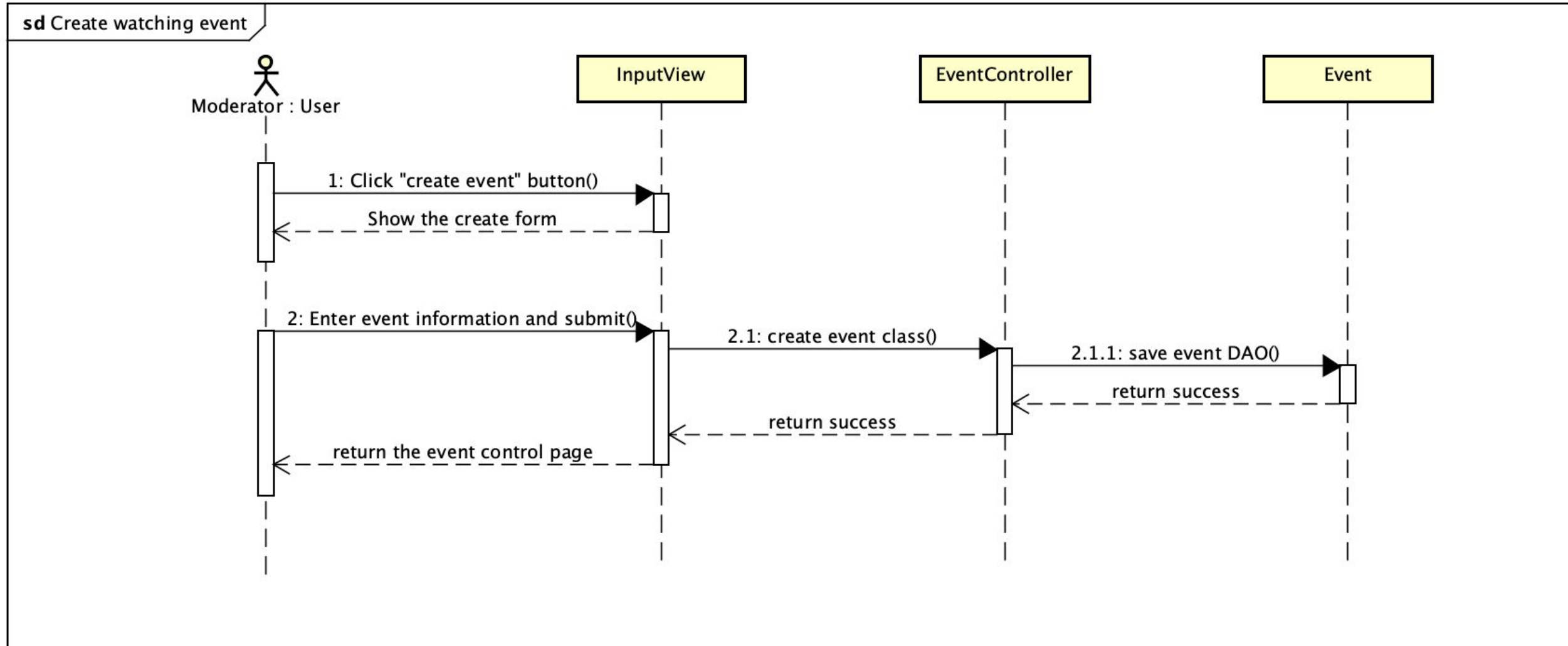
Sequence Diagram



Sequence Diagram

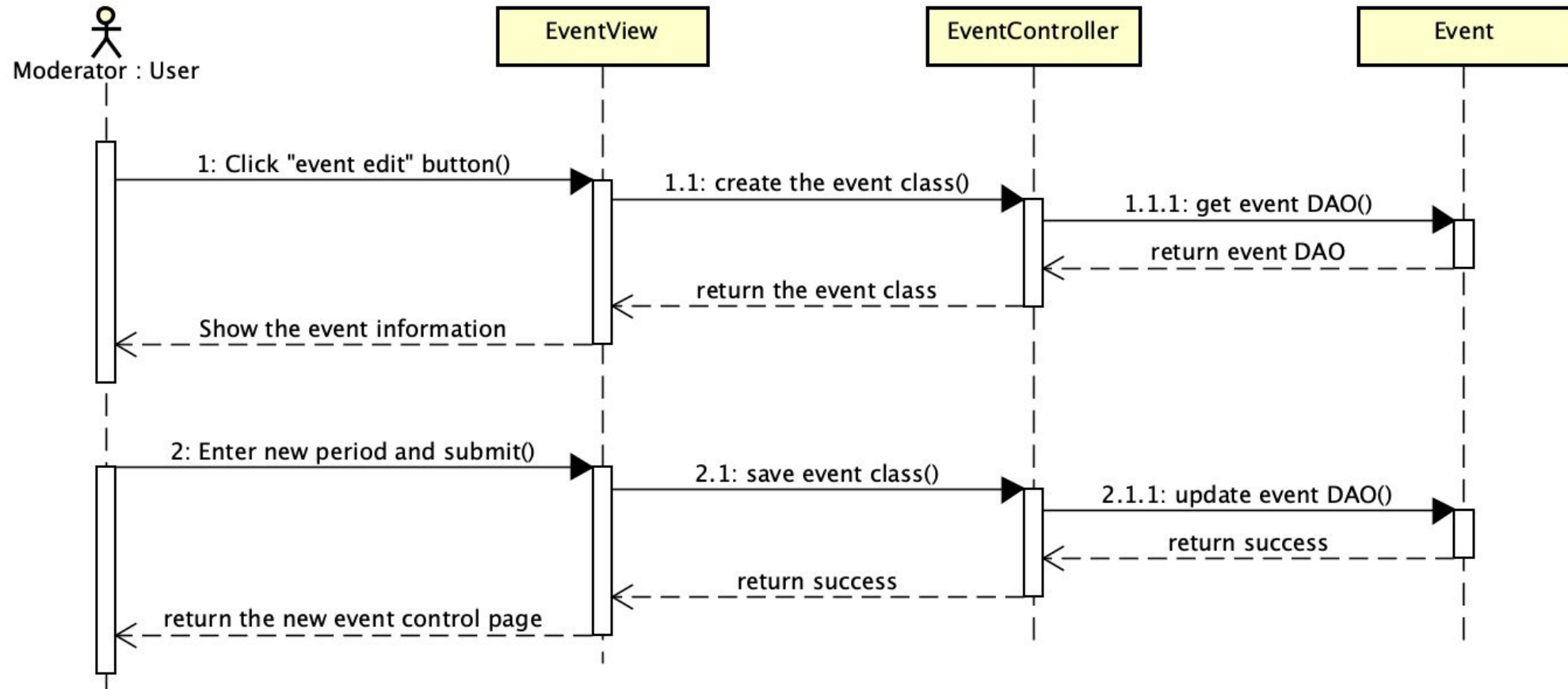


Sequence Diagram

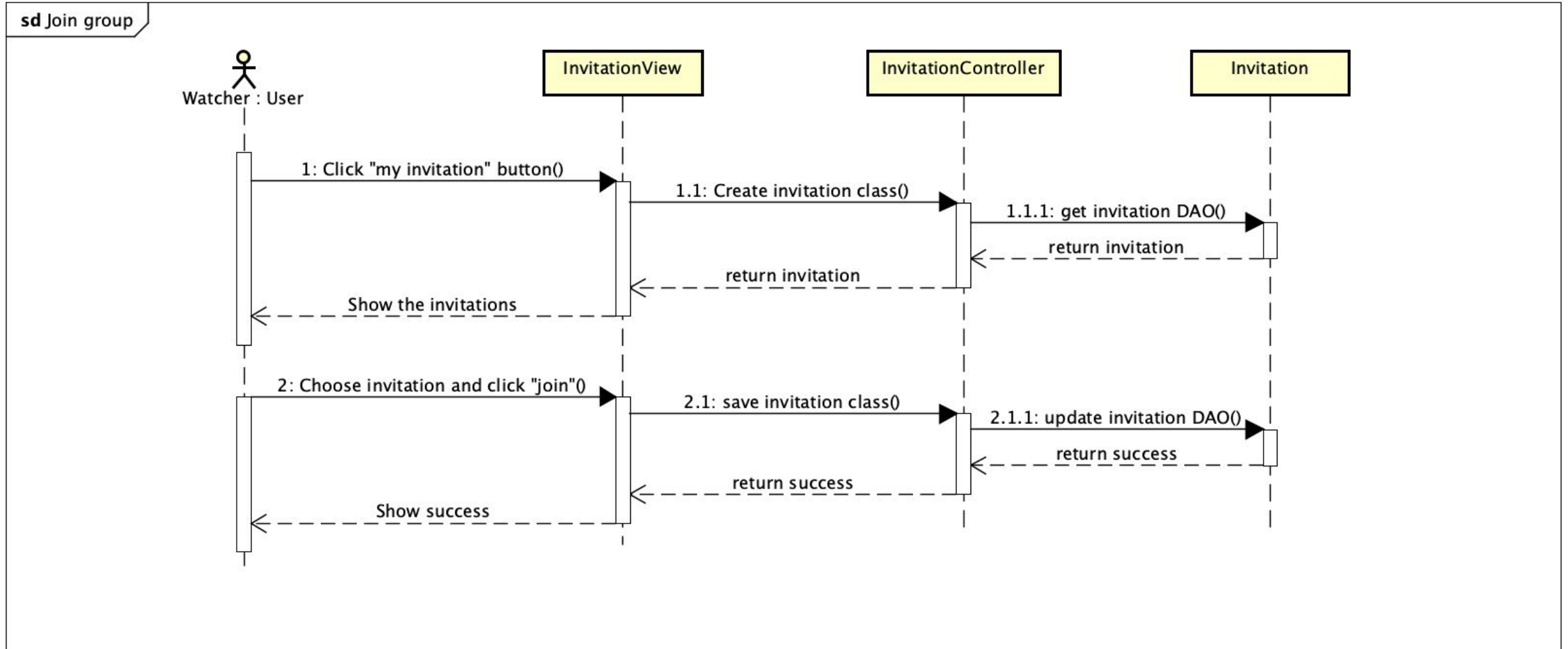


Sequence Diagram

sd Change voting period

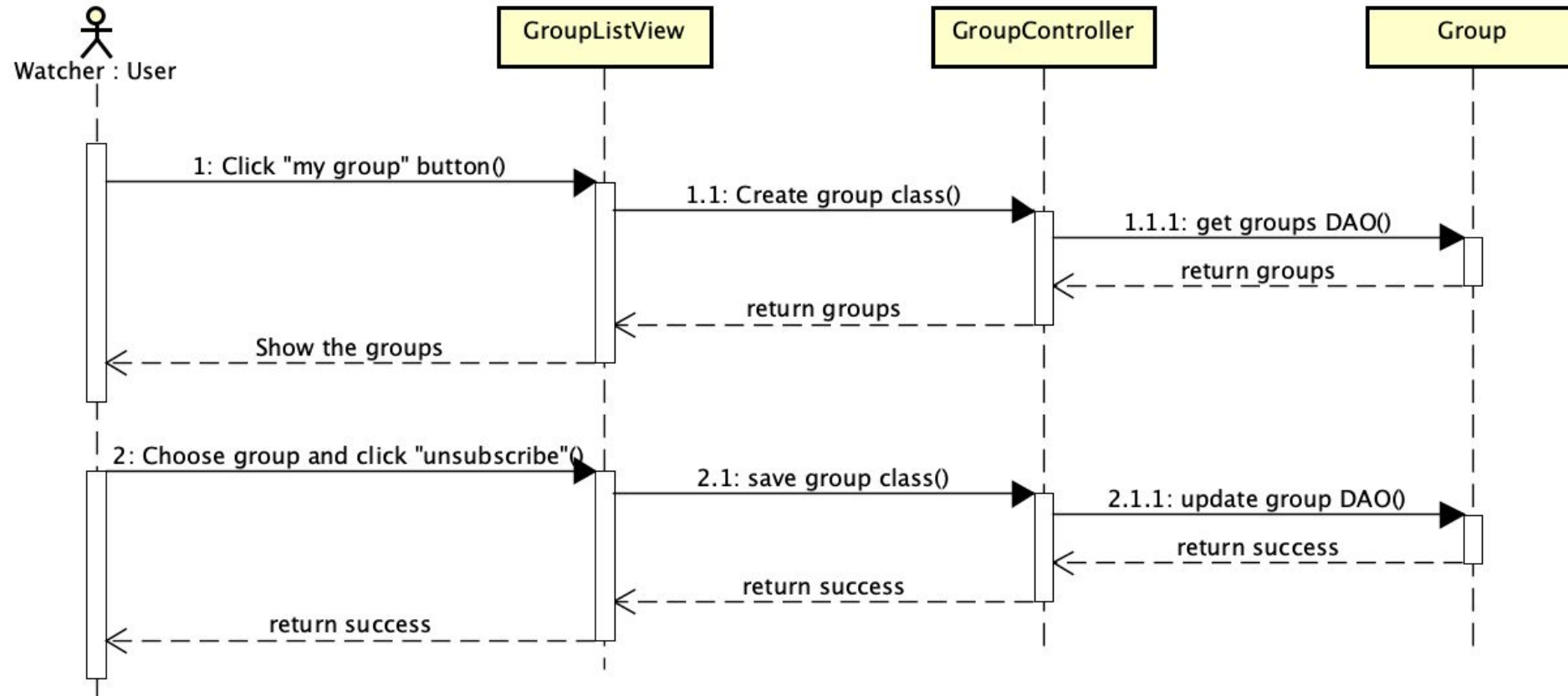


Sequence Diagram

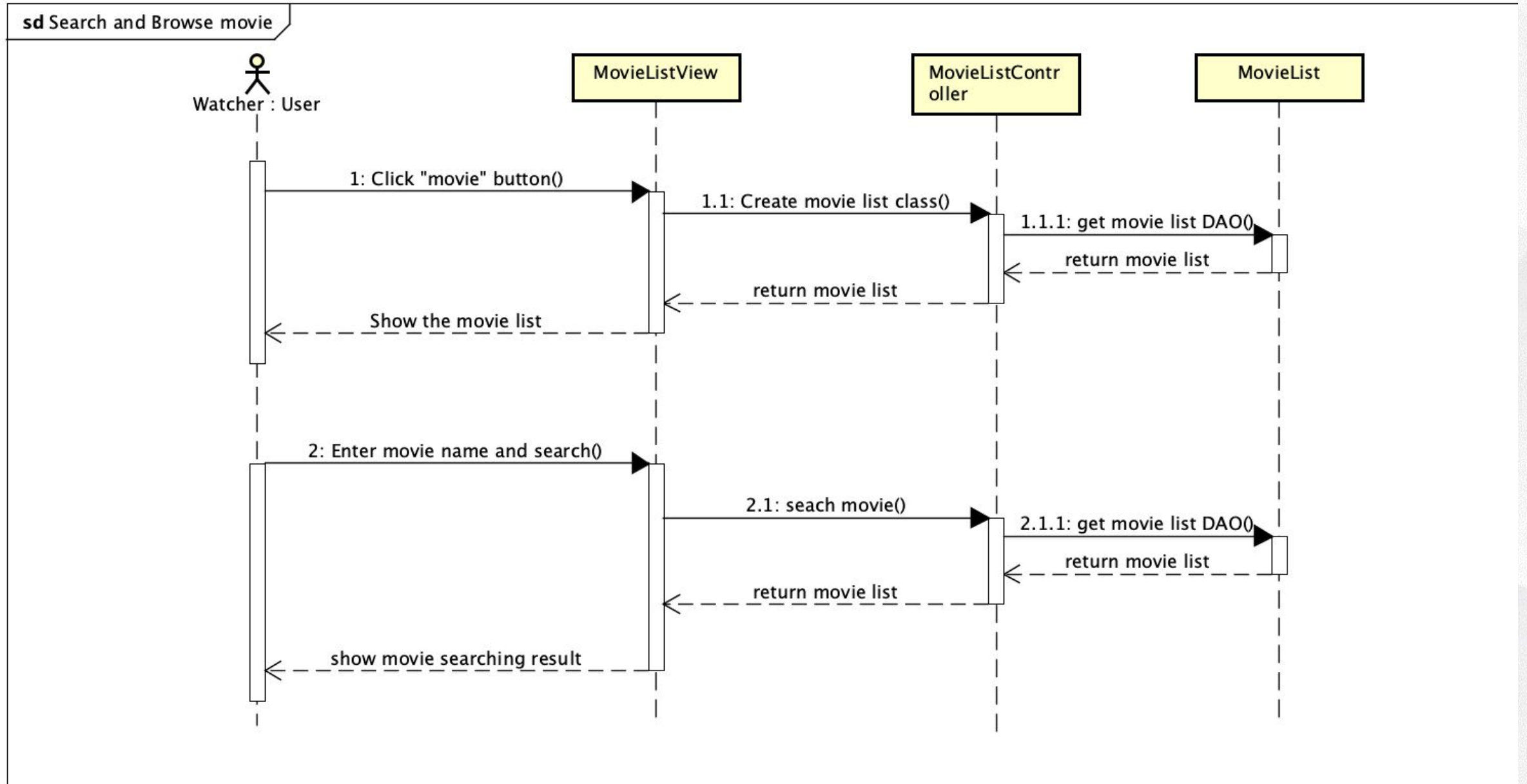


Sequence Diagram

sd Unsubscribe group

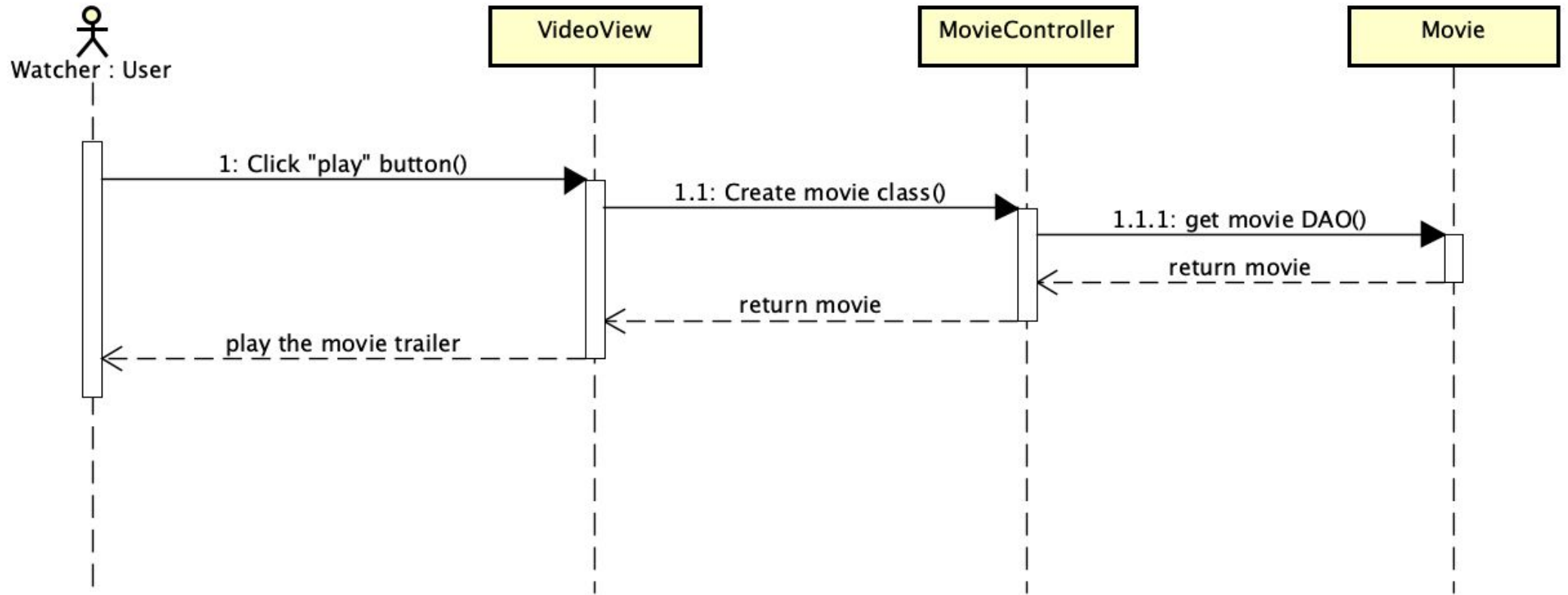


Sequence Diagram



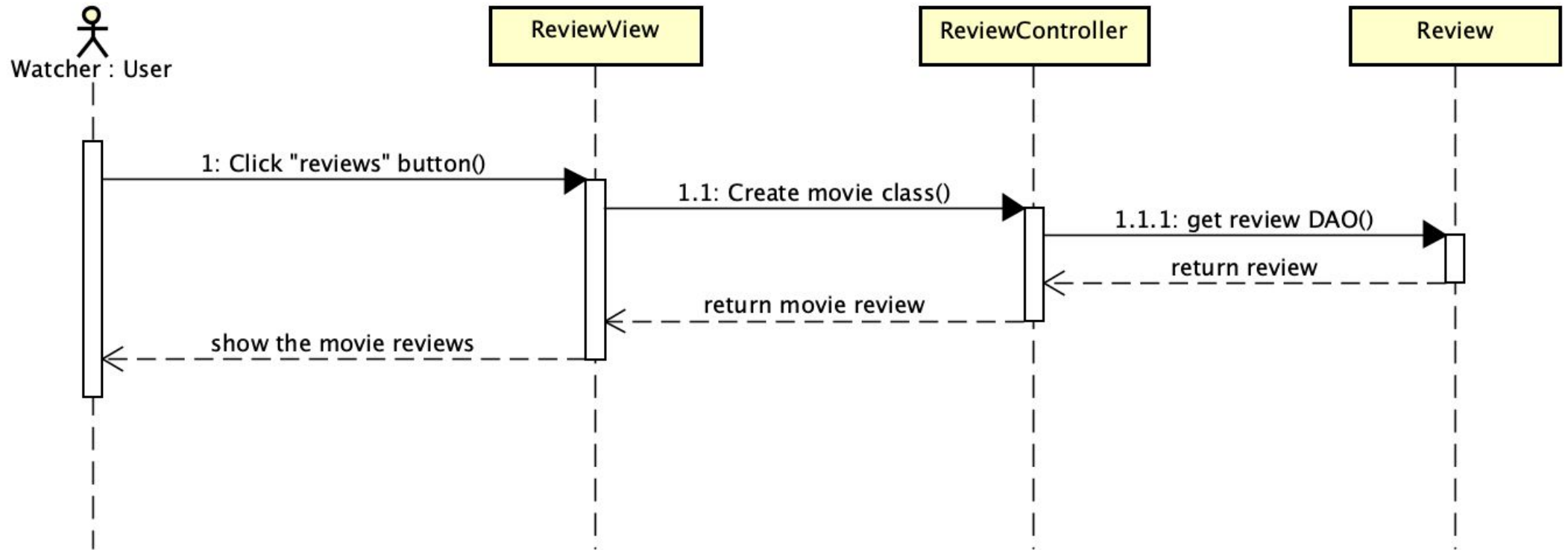
Sequence Diagram

sd Watch movie trailer



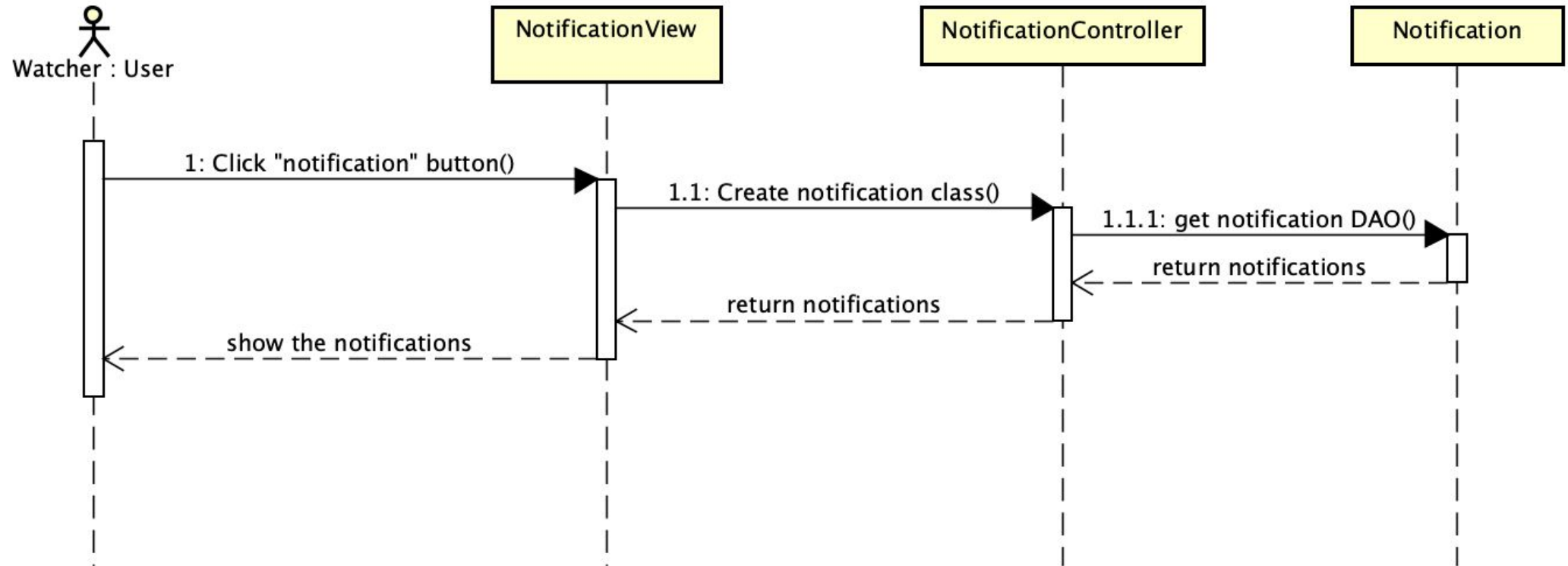
Sequence Diagram

sd Browse movie reviews



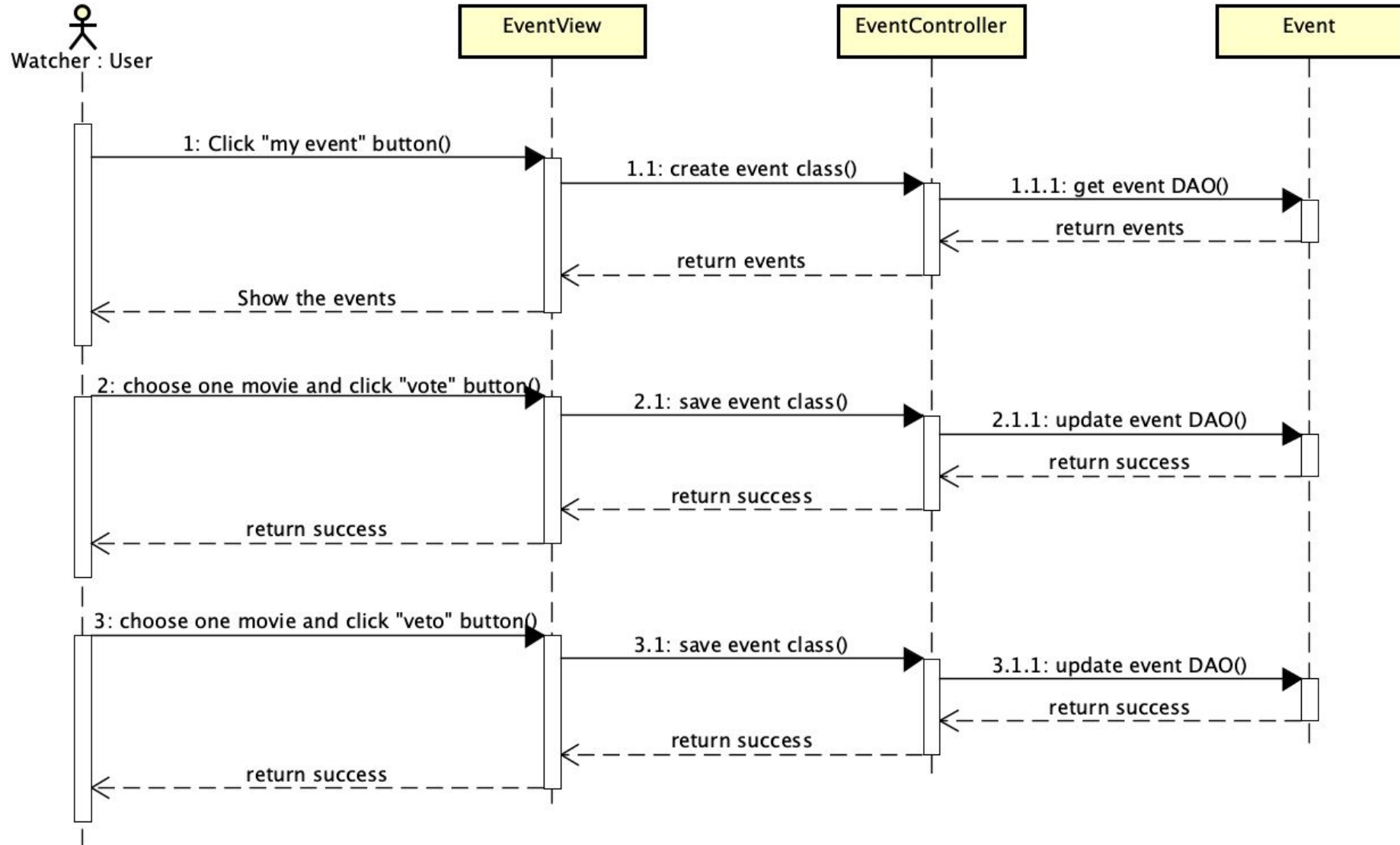
Sequence Diagram

sd Browse notification

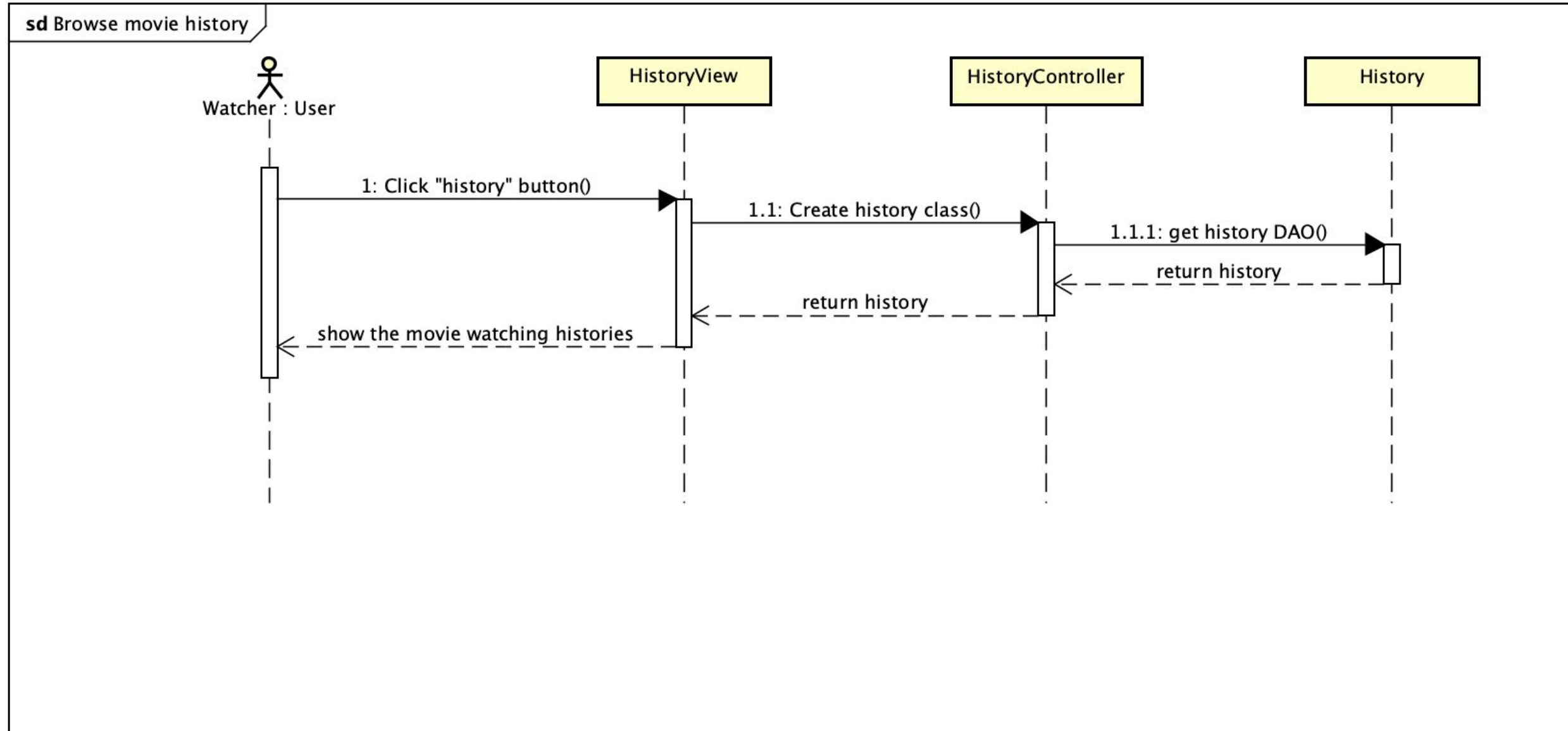


Sequence Diagram

sd Vote and veto

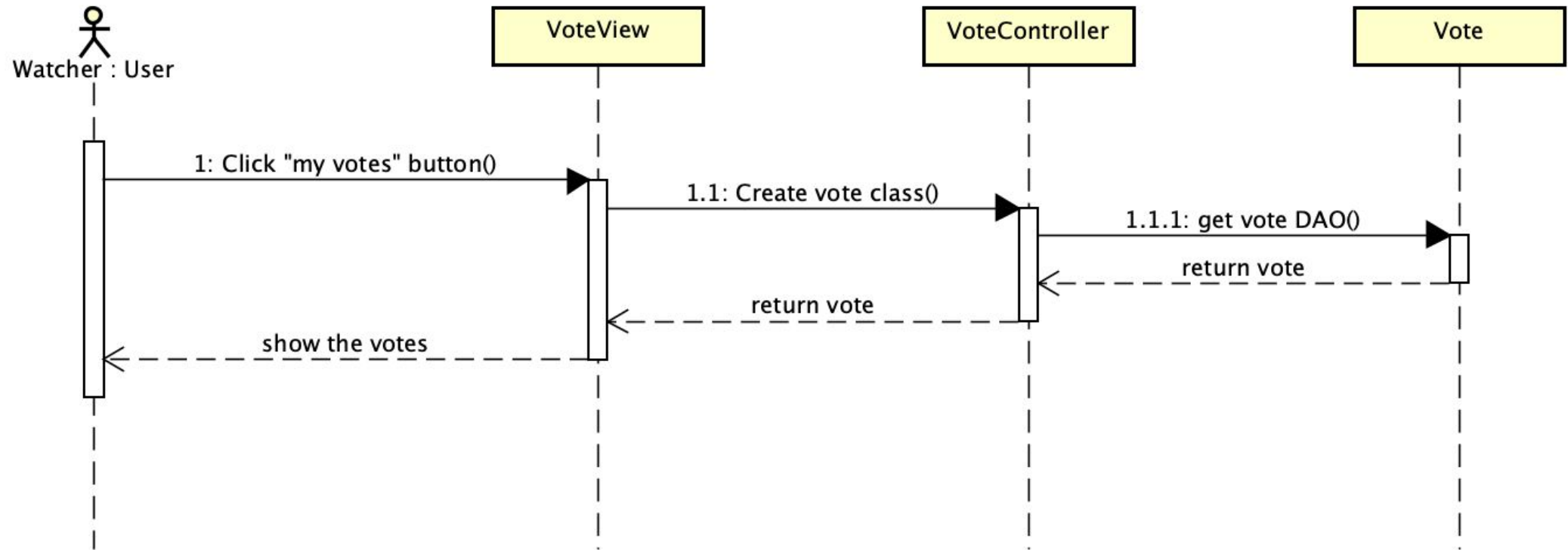


Sequence Diagram



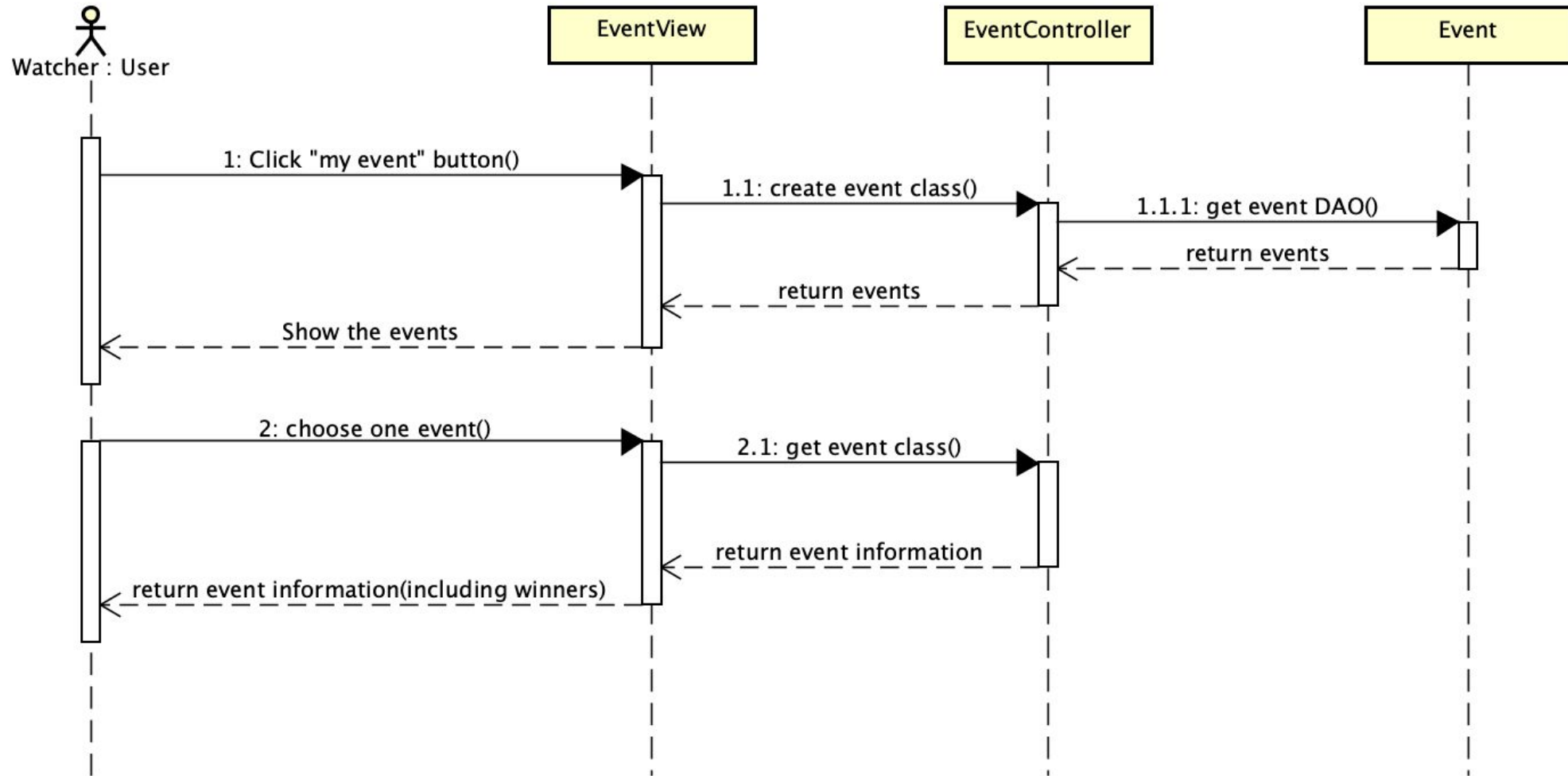
Sequence Diagram

sd Browse votes



Sequence Diagram

sd Browse winners





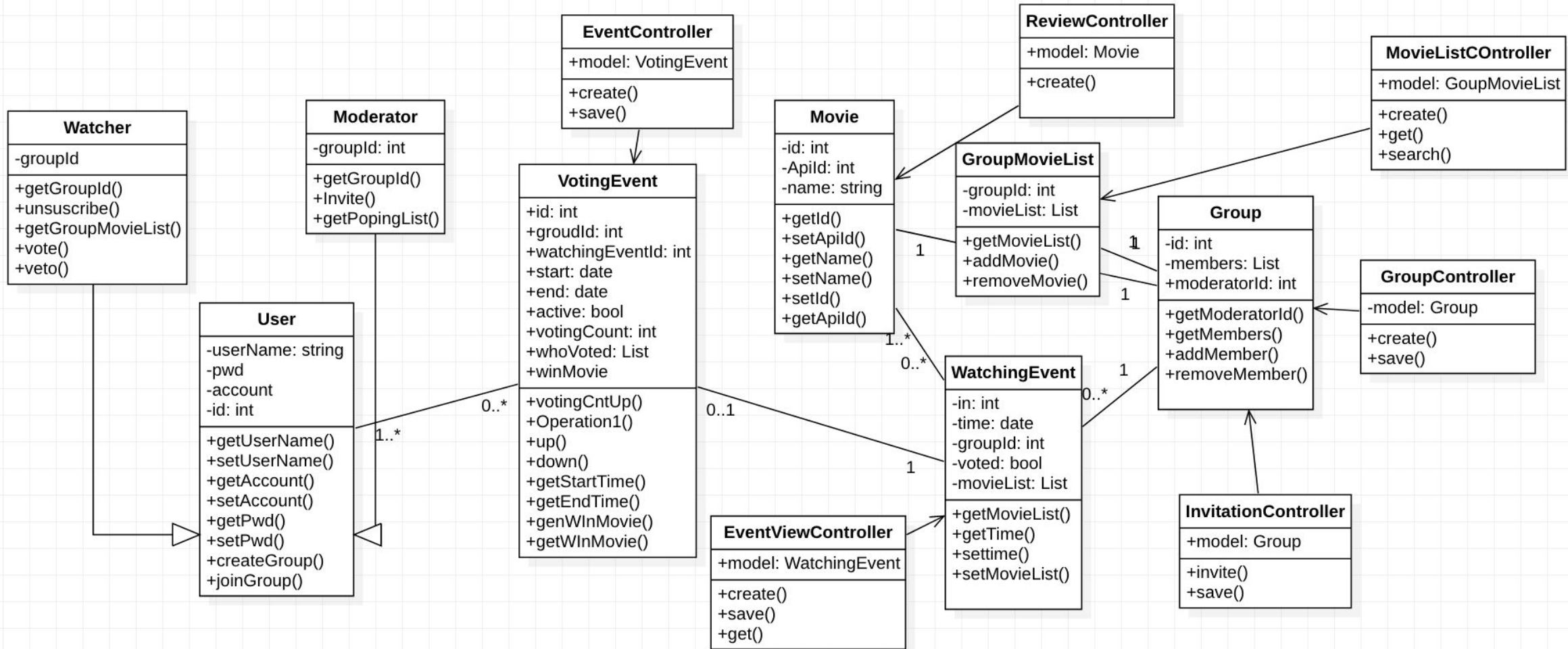
05

Part Five

Class Diagram Model



// Class Diagram Model

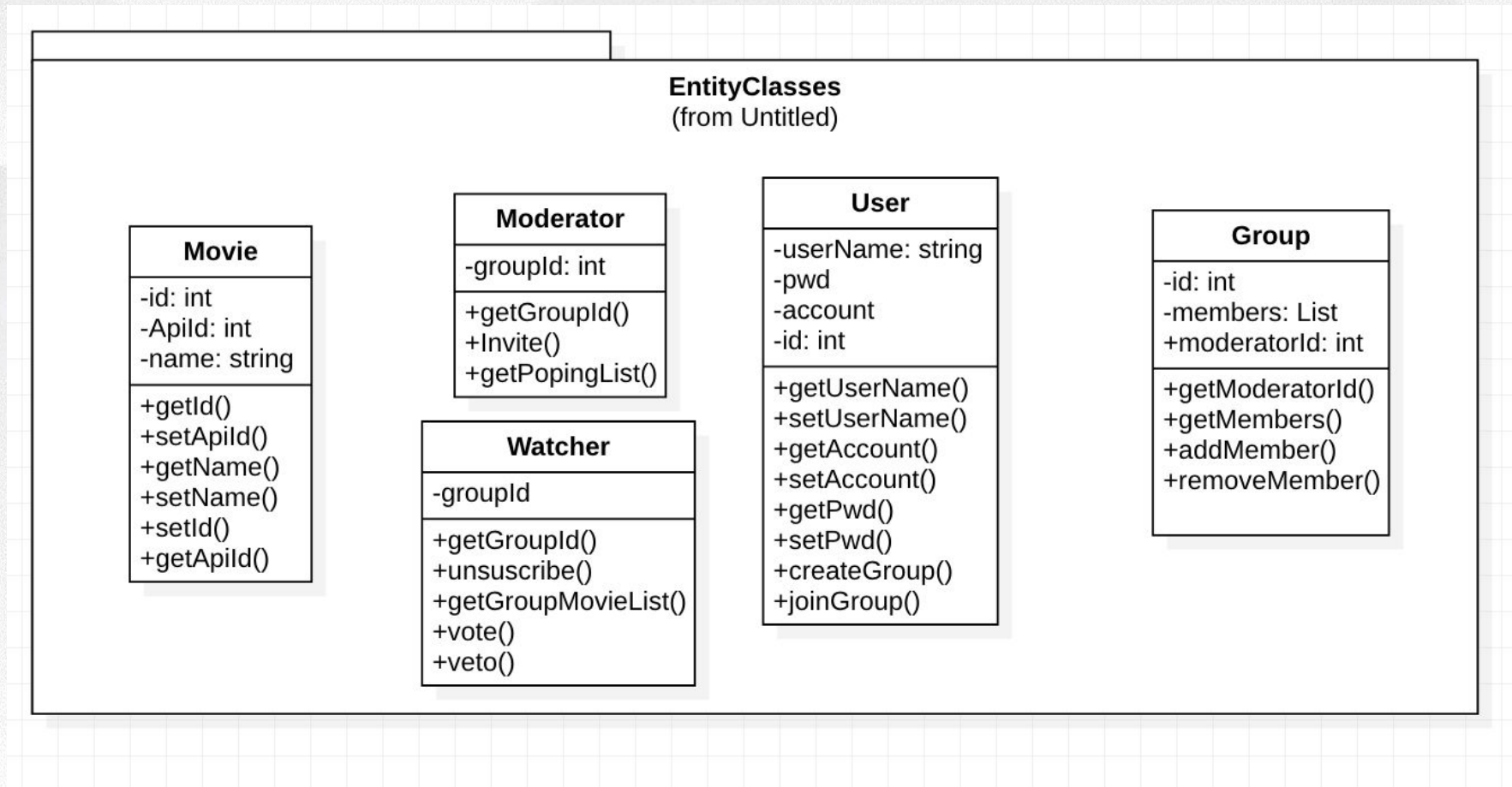




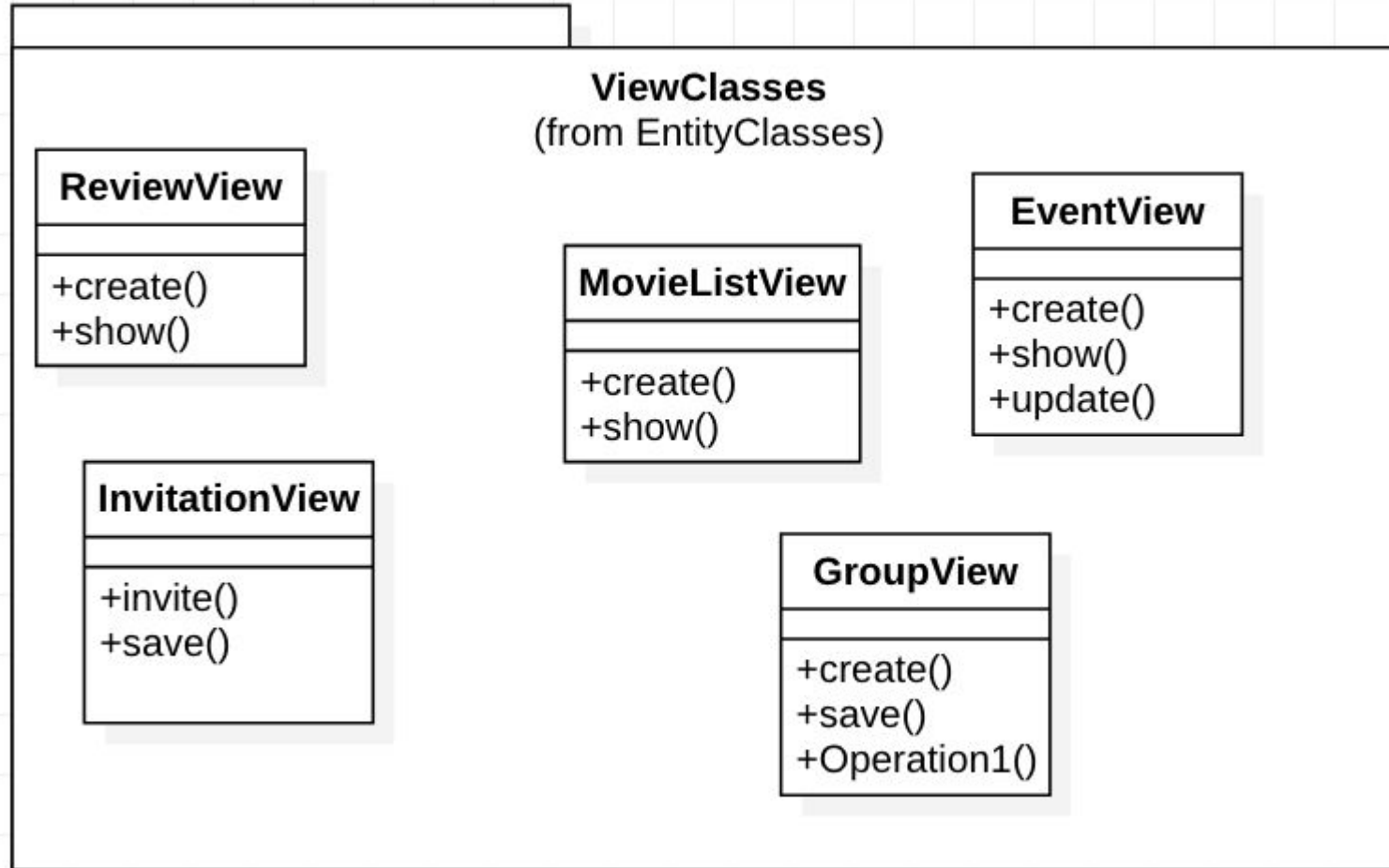
UML Packages



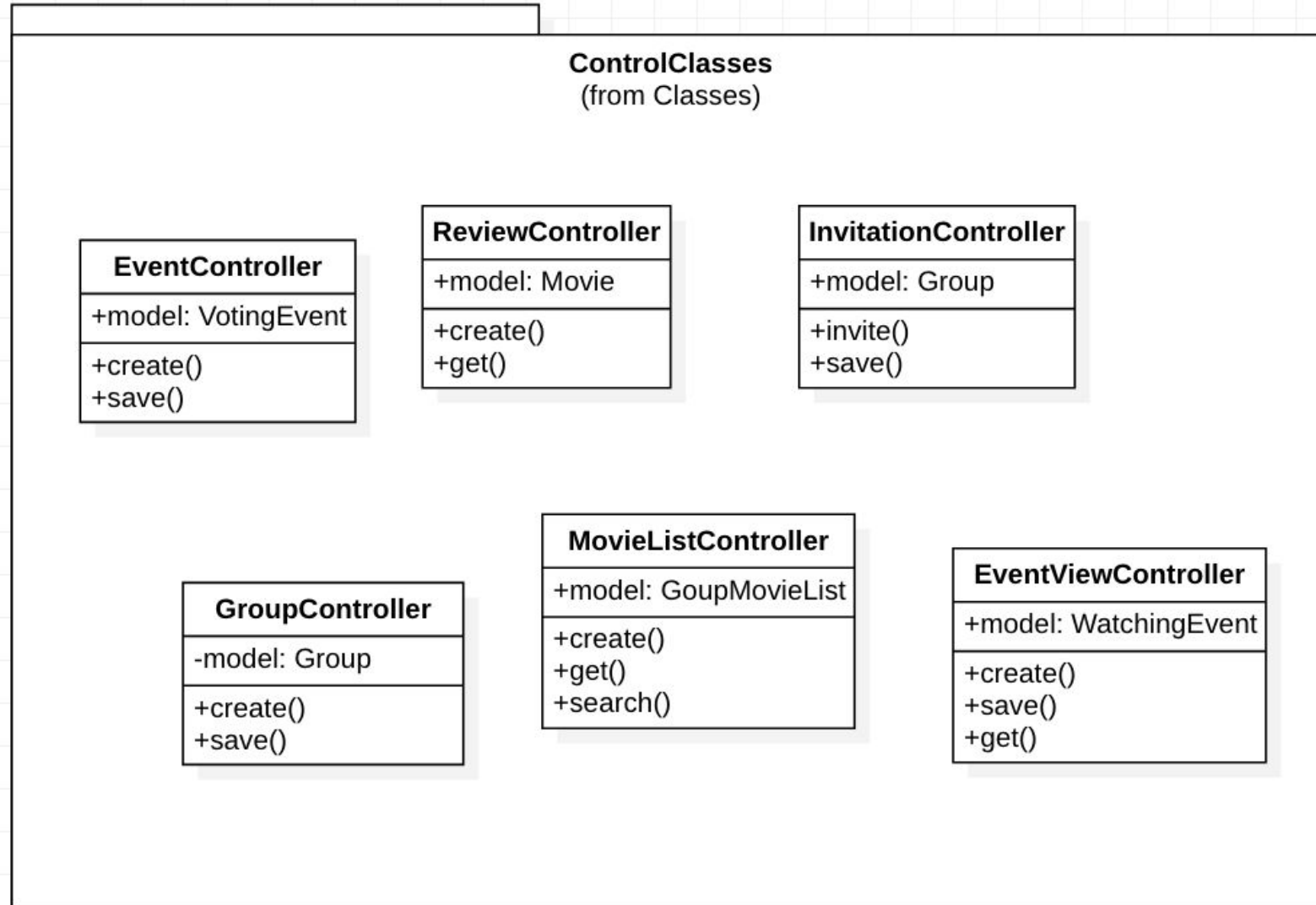
UML Packages



UML Packages



UML Packages

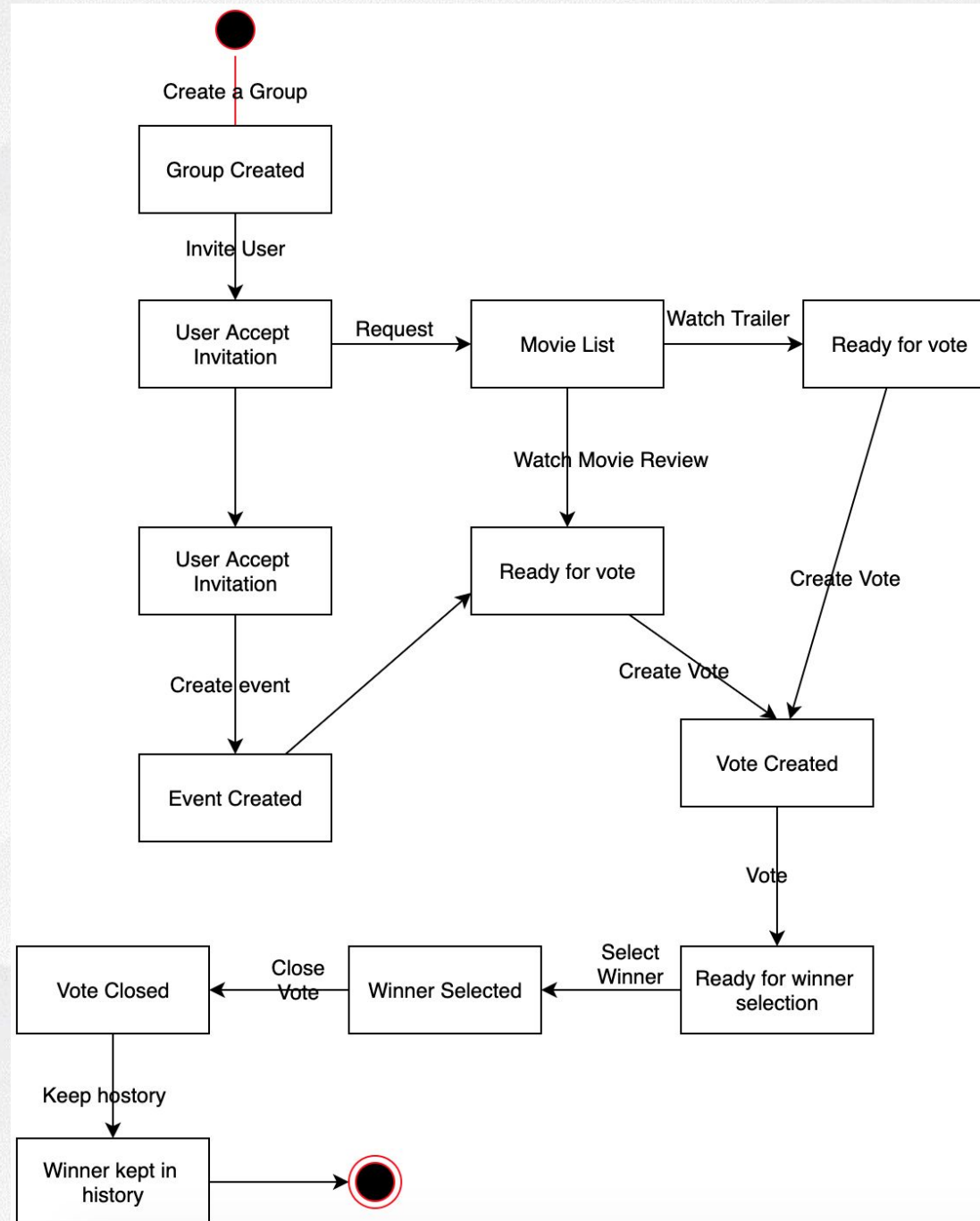




State Diagram



State Diagram





Thanks

The background features a complex pattern of overlapping light gray polygons. In the upper right, there is a yellow oval, a blue line, a red line, and a green oval. In the lower left, there is a red oval, a green line, a yellow line, and a blue oval.