Name: Casper Kristiansson **Student#:** 20938643 **Lec. sec.:** L15

Date: 22-11-09

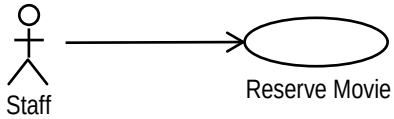
COMP 3111: Software Engineering

Lecture 15 Exercise: Movie Shop—Class Design

Reserve Movie—*Staff* scenario

This use case allows a staff to reserve up to 5 movies concurrently for a member.

Use-case Diagram



Preconditions

1. The member has less than 5 movies reserved.

Basic Flow

1. The use case begins when the staff actor chooses to reserve a movie.
2. The system prompts the staff to input the member number and movie ID.
3. The staff inputs the member number and movie ID.
4. The system reserves a copy of the movie for the member.
5. The system notifies the staff that the reservation has been made.
6. The use case ends.

Given the above use case and the Movie Shop domain model, answer the following questions.

a) How would you determine whether a copy of a movie can be reserved (i.e., what would you need to check)?

There are two different things we need to check if a copy of a movie can be reserved. The first thing is that the movie should already not be reserved. The second thing is that the movie should not be rented. If both statements are correct the movie can be reserved.

b) Given your answer to (a), how can the process of determining whether a copy of a movie can be reserved be made more efficient?

B) Something that can improve the efficiency is that the RentalCopy class should have an attribute we should keep track if the current copy is already rented, reserved or available. Doing this will minimize the number of different places we need to lookup if the movie is currently being rented. Another improvement could also be to store the movieID with the RentalCopy class. Doing this means that we do not need to follow the link between RentalCopy and Movie when we are checking if different copies are available.