CS3500 Software Engineering

Assignment 1: Modelling in UML

Adrian O'Riordan, November 2020

This practical is part of the continuous assessment for CS3500. You will be marked on the answers you submit. Total = 30 marks.

Submission details: You should submit a Word document via Canvas with **three** diagrams (generated from Modelio or a similar tool) and use case descriptions. Note that you do not submit the Modelio project.

The due date is *Monday 23st Nov.*

In your models, you will need to make (reasonable) assumptions but there is no need to state these assumptions but add notes to the document or diagrams if you need to clarify something that may not be clear. Model the system as described below and do not add extra information.

Following is a description for a information system for managing multi-screen cinemas. Note that this is a simplification of what would be present in a fully application, for example we do not model the user interface or the data storage in this assignment:

> *A company that operates cinemas wants a Web-based cinema software system to manage multiple cinemas allowing customer to book and buy tickets in advance. The system will be used by cinema staff to update information and by customers for ticket bookings.*
>
> *This is an information system for cinemas (described by name, address) that keeps all information relevant to the showing of films (name, length, distributor, cert, screen number), screenings (date, time), booking of tickets by patrons (name, address, dob) and payments. It provides the following basic services to customers: view listings, booking, payment and cancelling tickets. A booking can only be made if a screening is not full. Booking information needs to be recorded for auditing purposes. Payments can be made PayPal or credit card.*

Use Modelio or a similar tool to produce the following outputs:

i. A **Use case diagram** showing all the major use cases and actors. Write a description in the report for three **use cases**. You can follow the template of Fowler as in the lecture slides. *(12 marks)*

ii. A **Class diagram** that should contain classes, attributes, type information for attributes, operations, visibility specifiers for attributes and operations, associations, and multiplicity (if appropriate). Use UML standard types for type information (e.g. string, integer, date). There is no requirement to specify the return type or parameters for operations. Also, exclude constructors, destructors, and any implementation-oriented operations (e.g. toString) from the model. All associations should have a name, it may be useful also in some cases be add association roles. *(12 marks)*

iii. A **Sequence diagram** for the payment function. *(6 marks)*

end.