Assignment 3, Part (a): Use Cases and UML
ENGR110 2014

Due: 17th September 5pm

Goals
The goal of this assignment is to learn about methods for analysis and design of computer based systems using UML. Part 1 is focused on Use Cases and UML, whilst Part (b) of the assignment will extend this by looking at Class Diagram design using UML.

To submit
- Your Use Case diagram for UniFilms
- The sequence diagram for the main success scenario of the use case *Purchase ex-rental film*.
- The activity diagram for the use case *Purchase ex-rental film*.
- Your UML Use Case Bodies for the following use cases:
    - A customer with standard membership renting a film;
    - A customer with standard membership upgrading to premium membership;
    - A staff member processing the sale of an ex-rental film;
- Your factorised use case, and a description of the necessary pre- and post-conditions.

***Note**: You can either complete the use case, sequence, and activity diagrams by hand, or use a program such as argouml on the lab computer, or web based http://www.draw.io. If you draw by hand, then a scanned or photographed copy is just fine to submit so long as it is readable! We need an electronic copy, but you may put the paper copy in the box in CO145 - it may help the markers give you more marks!)*
You should use Word (or similar, e.g. open office) to create the use case bodies, inserting tables where needed.

Exercise 1: UML Use Case Diagrams
Task: Design, on paper, a UML Use Case diagram for a library computer system. Your diagram should show the relationships among actors and the subject (system), and use cases.

The library system needs to be able to conduct simple bookkeeping tasks, such as keeping track of the books and journals that users have borrowed. Staff members are allowed to borrow both books and journals, but students can only borrow books. The system should also allow users to browse books, reserve books (if there is a copy available) and enable librarians to keep the catalogue up to date.

You will need to identify the actors in the system, the functional requirements of the system (i.e. the use cases), and the relationships between actors and functional requirements.

Exercise 2: UML Use Case Descriptions

The following UML Use Case Body describes the main success scenario and alternative and error scenarios for the use case.

Task: Create a sequence diagram for the main success scenario and an activity diagram for the use case.

Use Case Body
Title: Reserve books
Summary: This use case describes the interactions between a *Book borrower* and the library system when reserving books.
Actors: Book borrower (primary), Library Information System (secondary)
Creation date: 03/09/14 Modification date: 03/09/14
Version: 1.0 Person in charge: Rebecca Ford

Main Success Scenario:

Actions of actors:	Actions of system:
1. Book borrower navigates to library website	
2. Book borrower enters search parameters for book they wish to reserve	3. System returns search results
4. Book borrower selects book to reserve	5. System checks availability
	6. System requests log in
7. Book borrower enters details	8. System requests authentication from Library IS
9. Library IS confirms details	10. System confirms log in, places book on reserve, and updates catalogue information
	11. System notifies user

Alternative Scenarios:
A1: Username and/or password incorrect
The A1 scenario starts at point 8 of the main success scenario

Actions of actors:	Actions of system:
9. Library IS rejects username and/or password	10. Go to M.6

Error Scenarios:
E1: Book not available to reserve
The E1 scenario starts at point 5 of the main success scenario

Actions of actors:	Actions of system:
	6. System notifies user that book is not available to reserve at this time. The Use Case fails.

Assignment: UniFilms

You have been contracted to design a web-based system for a new DVD rental chain, UniFilms. UniFilms is a new company that offers competitive film and video game rental to students studying at any one of New Zealand's universities. Students should be able to join UniFilms online, and having become a member, they should be able to browse, reserve, and order available films and video games online. To keep overheads down, the company has no stores; instead they mail members their film/game orders.

Customers can keep their films/games for as long as they like, but there is a limit on the number of items they may have out on hire at any one time. The limit depends on what type of membership has been selected. Regular membership costs $20/month and enables students to access 1 premium and 2 regular items at any one time. Premium membership is $40/month and enables them to access 3 premium and 5 regular items at any one time. They system should allow members to upgrade or downgrade their status at any point in time after joining, and they should also be able to update their contact details, for example, if they move home.

In addition to hiring films and video games, Premium members are able to buy new and ex-rental copies of these items. The processing of membership fees and purchases means that the system must also be able to handle financial transactions securely.

Because the system is available to students around New Zealand, head office has decided to have storage and mailing facilities in Auckland, Hamilton, Wellington, Christchurch, and Dunedin. Each of these facilities has their own range of films and games to hire, but they should also be able to transfer films and games between facilities according to regional differences in customer demand. These transfers and changes in stock availability must be tracked to keep records up to date within each facility.

Stage 1: Use Case Diagrams (30%)

Create a UML Use Case diagram, showing the relationships among actors and the subject (system), and use cases. You will need to identify the actors in the system, the functional requirements of the system (i.e. the use cases), and the relationships between actors and functional requirements.

Hints:

- Use a big sheet of paper, or a white board to start with.
- Remember to think about both the primary and the secondary actors.
- For each actor think about each functional requirement that the system must offer.
- Start simple, getting just the basic set of use cases. For example, you might ignore secondary actors to begin with. But make sure to include them in your final diagram.
- Keep copies of the diagrams of your simpler designs, in case you mess it up when you try to make them more complex.
- Remember to include a system boundary.

Stage 2: From Use Case Bodies to Dynamic Depictions (30%)

The following UML Use Case Body describes the main success scenario and alternative and error scenarios for the use case. From this you need to create:

1. The Sequence Diagram for the main success scenario [10%]
2. The Activity Diagram for the use case [20%]

Use Case Body

Title: Purchase ex-rental film (i.e. the use case)

Summary: This use case allows the UniFilms customer who holds a premium membership to purchase an ex-rental film.

Actors: UniFilms customer (primary), UniFilms Authentication System (secondary), Card Authorisation System (secondary)

Creation date: 01/09/14
Modification date: 03/09/14
Version: 1.1
Person in charge: Rebecca Ford

Main Success Scenario:

Actions of actors:	Actions of system:
1. Premium customer navigates to webpage	
2. Premium customer browses or searches site for the film they wish to purchase	
3. Premium customer selects film to purchase	4. System requests log in (user name and password details)
5. Premium customer enters details	6. System requests authentication from UniFilms AS
7. UniFilms AS confirms username an password and indicates membership level	8. System confirms log on to customer.
	9. System confirms that customer has premium membership, and checks to see if film is available to purchase ex-rental
	10. System displays price and requests payment details
11. Premium customer enters payment details.	12. System requests authorisation from Card AS
13. Card AS confirms details and processes payment	14. System informs customer that payment has been successful.
	15. System creates record of purchase and submits for processing.

Alternative Scenarios:

A1: User name and password incorrect
The A1 scenario starts at point 6 of the main success scenario

Actions of actors:	Actions of system:
7. UniFilms AS rejects username and/or password	8. Go to M.4

A2: Payment details incorrect
The A2 scenario starts at point 12 of the main success scenario

Actions of actors:	Actions of system:
13. Card AS rejects payment details	14. Go to M.10

Error Scenarios:

E1. Customer not a premium customer
The E1 scenario starts at point 8 of the main success scenario

Actions of actors:	Actions of system:
	9. System shows that customer does not have premium membership 10. System informs customer that they are unable to purchase film with their current membership status; the Use Case fails

E2: Film not available to purchase
The E2 scenario starts at point 8 of the main success scenario

Actions of actors:	Actions of system:
	9. System shows that the film is not available for purchase 10. System informs customer that they are unable to purchase film; the Use Case fails.

Hints:

- For the scenario diagram think about who (actors, system) initiates each action
- Remember that the activity diagram is essentially a flowchart, showing flow of control from activity to activity. The transitions are triggered at the end of activities or actions; steps can be carried out in parallel or in sequence.
- Activities are complex and can be broken down or interrupted by events, actions are simple and cannot be broken down or interrupted.

Stage 3: More Use Case Bodies (30%)

Create UML Use Case Bodies for the following use cases:

1. A customer with standard membership renting a film;
2. A customer with standard membership upgrading to premium membership;
3. A staff member processing the sale of an ex-rental film;

Remember to include the main success scenario and all the alternative and error scenarios in the use case body.

Stage 4: Organising and Structuring Use Cases (10%)

Often we see repeating steps across a number of different use cases. It is possible to avoid this and organise use cases by adding <<include>> relationships between use cases, and making use of pre- and post- conditions.

Identify a part that the different use cases in your system have in common and factorise it in a new case to be included in the former.

Describe your new use case using a text description (Use Case Body), and determine the appropriate pre- and post- conditions for the new use case and former use cases in your system.