Title: European Ornithological Trust Survey System

Project Description

The European Ornithological Trust (EOT) is an international voluntary (not for profit) organization. Its mission is to promote the conservation of wild bird species through the provision of accurate technical and scientific data. This data covers bird populations in the wild and the threats to those populations, particularly the threats caused by human activity.

A European Ornithological Trust Survey System will be developed to:

- record observations of bird species on a digital map;
- allow the observation data to be analysed by bird species or by visit;
- allow the data to be transferred to the central system for collation;
- allow configuration data to be transferred from the central system;
- allow for future developments.

This group project aims at analysing the software requirements and designing object-oriented models using UML for the European Ornithological Trust Survey System based on the case study description. You need to complete the project in a group of 4-5 members and evaluation will be done in two phases.

Project Duration: Week 4 – Week 14

You are required to model the following UML diagrams:

- **Use case diagram**
  - Model a complete use case diagram for European Ornithological Trust Survey System.
  - Show explicitly the generalization, include/exclude relationships
  - You need to write a use case description for each use case based on the template provided (refer to Appendix 1). Besides referring to the SRS, you can also add more details to show the flow of events (paths) which represents the actor actions and system responses. You are required to set the priority of each use case too.

- **Class diagram**
  - Model an entity class diagram to show the structure of the European Ornithological Trust Survey System.
  - Show explicitly the class attributes and operations and also the relationships between the classes (e.g., generalization, aggregation, composition), multiplicity specifications, and other model elements that you find appropriate.
  - You need to apply **CRC (Class–Responsibility–Collaboration) cards** approach to model interaction between objects based on the use case scenarios. In the report, you need to include all the CRC cards for your class diagram.
  - Create a **data dictionary** to record definition of each class. Each class is written a short description defining its scope / detail of the class.

- **Activity diagrams**
  - Model activity diagrams to describe system functions that are represented by the use cases.
  - The pre-condition, post-condition and essential process or flow of events in the use case description can be used as the basis for drawing your activity diagrams.
  - Show explicitly swimlanes (partitions), decision and merge nodes, flows and guard conditions to the diagram.
  - You should model the detailed process of each use case using one activity diagram. If the process described in the SRS is too simple, you may include more steps OR merge two to three related use cases into one activity diagram.

- **Sequence diagrams**
  - Model the interactions that define the message passing between lifelines (e.g. objects) to capture the responses of all the objects that are included in a single use case.
  - You should model the Sequence diagrams for each use case.
  - The Sequence Diagrams should represent the detailed object interaction that occurs for one use case or for one operation. Show explicitly conditional interactions.

- **State Machine Diagrams**
  - Model a state machine diagram for each class that has more than one state (1 diagram for 1 class). Capture all the possible responses of a single object to all the use cases in which it is involved.
  - Indicate all possible states and transitions for the class being considered. For each transition, specify relevant events, triggers, conditions and actions.
  - For each diagram, you need to include a description that describe the object life cycle it is trying to show and specify how an object behavior causes it to change from one state to another state. For each transition, specify relevant events, conditions and actions.
  - Apply the Behavioural or the Lifecycle Approach or combination of the two to develop the state machine diagrams.

Evaluations:

- **Draft Submission – 9th week (3%)**
  Present your draft UML models to the course instructor. Evaluation criteria are:
  - Demonstration of knowledge, flow of information and effectiveness in the draft report.
  - Each group need to submit draft Use Case Diagram, Class Diagram and Activity Diagrams.
  - Peer evaluation. (A factor of 0, 0.5 or 1.0 to be multiplied with the marks obtained in this phase to produce the final marks for the phase.)
  - Zero mark if any group does not submit the draft report.

  Submission:
  - Submit soft copy of draft report and peer evaluation form via Spectrum
  - Submit hard copy of draft report to course instructor / put it into the mailbox (Block B)
  - **Due Date: 11 November 2016 (Friday, Week 9)**

- **Final Report – 14th week (20%)**
  Present your final report which include all the UML diagrams to the course instructor. Evaluation criteria are:
  - Demonstration of knowledge, flow of information and effectiveness
    - Use Case Diagram – 4%
    - Class Diagram – 4%
    - Activity Diagrams – 4%
    - Sequence Diagrams – 4%
    - State Machine Diagrams – 4%
  - Peer evaluation. (A factor of 0, 0.5 or 1.0 to be multiplied with the marks obtained in this phase to produce the final marks for the phase.)

  Submission:
  - Submit soft copy of report and peer evaluation form via Spectrum
  - Submit hard copy of report to course instructor / put it into the mailbox (Block B)
  - **Due date: 16 December 2016 (Friday, Week 14)**

- **Presentation skills. (2%)**
  - **Presentation Date:** 14 December 2016 (Wednesday, Week 14)

Peer Evaluation

Each student needs to fill in a peer evaluation form for each phase of evaluation. The peer evaluation will result in a contribution factor to be computed for each group member based on the formula stipulated below:

Peer Evaluation Score	Contribution Factor
0-4	0
5-8	0.2
9-12	0.4
13-16	0.6
17-20	0.8
21-24	1.0

Final contribution factor for each student will be the mode of all contribution factors given by his/her peers. Critical disputes in assigning peer evaluation scores will be investigated by the course instructor.

Determination of Marks for Each Phase

The marks for each student in each phase will be computed using the following formula:

*Group's marks in that phase * final contribution score*

For example, if a group obtains 13 marks for phase 1 evaluation, and the final contribution factor for a student is 0.8, the student's marks for phase 1 evaluation will be:

*13 * 0.8 = 10.4*

Late Submission

If for some good reason it is impossible to get the assignment in by the deadline, email the lecturer to negotiate a late submission. This should be done prior to the deadline. Late assignments without negotiation will have marks deducted (5% of the mark for that assessment per week overdue to a maximum of 50%).

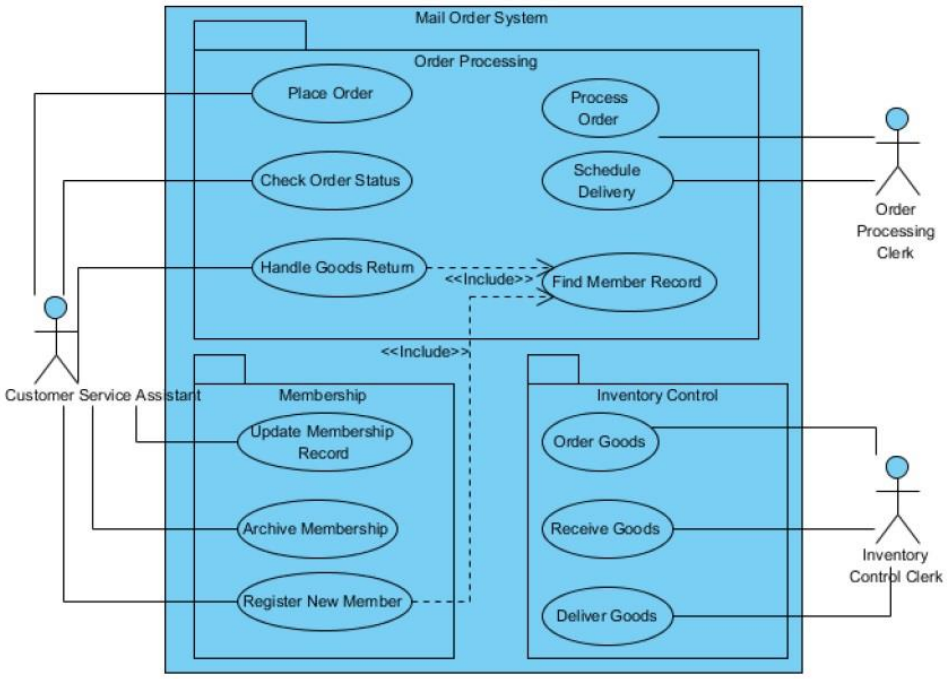
Penalties

If it is determined that plagiarism has occurred, one or more of the following penalties may be imposed:

- loss of all or part marks for the assessment item;
- downgrading the final grade in the course;
- imposing a grade of fail in the course.

Appendix 1:

A Sample Use Case Diagram for a Mail Order System



Template for Use Case Description

Element	**Description**	
Use Case Name	Place Order	
Use Case ID	UC-100	
Requirements ID	(Specify the Requirements ID in SRS that you cover in this Use Case)	
Priority	High	
Actor(s)	Customer Service Assistant	
Description	A Customer Service Assistant places an order then submits it for processing.	
Pre-condition(s)	The member must have registered with the system.	
Post-condition(s)	The Customer's order will be directed to the order processing department for processing.	
Flow of events	**Actor Action**	**System Responses**
	1. Select "Place Order" 3. Add Item to Order 4. Submit order	2. A blank order form with generated order ID 5. Notify order was saved and forward to Order Processing Clerk.
Alternative flows and exceptions	At any time the Customer Service Assistant can decide to suspend the ordering process and come back to it later, or decide to cancel the order.	

Definition for each element in the template:

- **Use Case Name:** Name of the use case
- **Use case ID:** ID of the use case
- **Requirements ID:** ID of the requirements specified in SRS
- **Priority:** The development priority of this use case
- **Actor(s):** the roles that people, other systems or devices take when communicating with a particular use case or use cases
- **Description:** the purpose and the role of this use case in enabling the actors to do their job
- **Pre-condition(s):** the conditions must be true/satisfied before the use case can take place
- **Post-condition(s):** the conditions that will be established as a result of invoking this use case
- **Flow of events** (Actor action and system response): a step-by-step description of the interactions between the actor(s) and the system, and the functions that must be performed in the specified sequence to achieve a user goal
- **Alternative flows and exception:** Major alternatives or exceptions that may occur in the flow of events