Unit:
Object-Oriented System Analysis and Design

Assignment title:
Estate Agency System

Autumn 2024 – Summer 2025

Important notes

- Please refer to the *Assignment Presentation Requirements* for advice on how to set out your assignment. These can be found on the NCC Education website. Hover over 'About Us' on the main menu and then navigate to 'Policies and Procedures' then scroll to the 'Student Support' area.
- You **must** read the NCC Education document *Academic Misconduct Policy* and ensure that you acknowledge all the sources that you use in your work. These documents are available on the NCC Education website. Hover over 'About Us' on the main menu and then navigate to 'Policies and Procedures' then scroll to the 'Student Support' area.
- You **must** complete the *Statement and Confirmation of Own Work*. The form is available on the NCC Education website. Hover over 'About Us' on the main menu and then navigate to 'Policies and Procedures' then scroll to the 'Student Support' area.
- **Please note, word count is not applicable on this assignment as the artefact to be produced is a programming code (with a word document containing screenshots of the results, a filled-out object definition sheet and a test log)**
- You must submit a paper copy and digital copy (on disk or similarly acceptable medium). Media containing viruses, or media that cannot be run directly, will result in a fail grade being awarded for this assessment.
- All electronic media will be checked for plagiarism.

Introduction

In this assignment, you will apply knowledge and skills of key elements of the Object-Oriented Systems Analysis and Design approach to software development.

You should read the information provided and each task carefully before you begin.

You must work independently and should not share your work with other students. All work produced must be your own.

Any sources of information, text, images created by others that you include in your work must be clearly identified and referenced. If you use work of others without permission or without proper acknowledgement, you may be disqualified from the assessment.

You may ask your tutor for support if you have questions about the requirements of each task, however, they cannot guide you with solutions for the tasks.

Scenario (Problem Statement)

North Yorkshire Property services (NYPS) is a real-estate agency business. The company's chief executive officer (CEO) has asked you to develop a system (Property Viewing System (PVS)) to help it run its property sales operation. The following is a brief description of the system they would like.

PVS is concerned mainly with the setting up of appointments for clients to view properties.

NYPS has five branches; each branch has a single manager and a number of representatives (part or full-time employees).

Clients may be buyers or sellers and must register with NYPS.

Sellers are property owners who use NYPS to market their property/properties. Each property has a single client as an owner. A property sale is managed from a single branch, though details are available through the system to all branches.

Buyers may browse details of properties available. When they would like to view a property, the representative makes a viewing appointment. Details of buyer's appointments and viewer's comments are kept on file.

Buyer's details are kept until they indicate that they are no longer interested.

The Property Viewing System (PVS) will provide support for NYPS by meeting the following requirements:

- Recording information about the company's employees, including their names and employee numbers and for representatives, their grade and whether part or full time.
- Recording information about the structure of the management of the company
- Recording information about the branches, including their names, locations and staff who work in them.
- Recording information about each of the properties that the company is marketing, including address, description, picture(s), price, type (bungalow, house, apartment etc.), seller, branch responsible, number of bedrooms, date

first offered for sale, status (for sale, under offer, offer accepted, contract exchanged, sale completed).

- Recording information about offers made for a property, including offer price, date, action taken (accepted / rejected), client making offer.
- Recording information about clients: names, address etc, any properties they are selling, properties viewed and offers made.
- Recording information about appointments that have been made, including the representative making the appointment.
- NYPS staff should be able to add, amend and delete property details, client details, offers, and appointments. A manager should be able to add, amend, and delete employee and branch details.
- Clients should be able to browse details of properties and search for properties (on district, price range, number of bedrooms, type)
- The system must respond to requests quickly and be secure since personal information is collected.

Task 1: Requirements Elicitation – 15 Marks

The problem statement is brief, incomplete, ambiguous and vague in some places. You should therefore have a number of questions that you want to ask to clarify your understanding. (You cannot solve a problem that you do not fully understand).

During requirements analysis, you need to discover as much ***relevant*** information about the system as you can and clarify any ambiguities.

a) Identify FOUR (4) stakeholders in the PVS system. Provide a clear justification for your choice.

b) Explain TWO (2) methods you would use to elicit the system requirements. Explain why you would use these methods for *this system*.

c) Identify at least FIVE (5) additional pieces of information you need in order to clarify the problem statement above. Explain why each piece of information is needed.

Task 2: Requirements – 10 marks

A requirements specification is an essential document for the development of a system. In this task, you will produce important elements of the specification.

a) Explain the difference between a functional and non-functional requirement and provide an example of each from the PVS system.

Analyse the problem statement and create succinct lists of the following:

b) Functional requirements.

c) Non-functional requirements.

NB. Where you consider that there are ambiguities / omissions in the problem statement, you should make reasonable assumptions and state what they are explaining why they are relevant.

For the highest grade you are expected to identify all requirements from the problem statement and FOUR (4) additional functional/ non-functional requirements with justification.

Task 3: Use Case Model – 30 marks

Use Case models can be used in various stages of system development.

a) Explain the purpose of a Use Case model, with an example of how it can be beneficial in the development of the above system.

Analyse the problem statement above:

b) Identify the Actors in the above system, justifying your decisions for selecting them.

c) Use a CASE tool to draw a Use Case diagram for the above system. You will gain additional credit if you correctly identify and ***appropriately*** use generalisation, <<includes>> or <<extends>>. *NB You do not have to use them all!*

d) Identify TWO (2) scenarios for the Use Case *Make appointment.*

e) Document a Use Case for *Make appointment*. You should use the template provided in the Appendix.

Task 4: Sequence Diagram– 15 marks

a) Explain the purpose of a Sequence Diagram, with an example of how it can be beneficial in the development of the above system.

b) Use the CASE tool to draw a Sequence diagram for one of your scenarios for *Make appointment.*

Task 5: Class Diagram – 30 Marks

a) Analyse the problem statement to identify classes and use a CASE tool to produce an implementation level Class Diagram for the system.

   The class diagram must show attributes, operations, and the relationship of classes to each other with multiplicity. It should include class and association names. You will have to invent suitable names for the associations.

   The use of abstract classes and sub-classes (where appropriate) will attract additional marks.

b) Provide a justification for why each class was selected for inclusion.

Appendix: Use Case description template

Here is the template that you should complete for your Use Case Description. You should use this layout.

Template:

Use Case Name	
Participating Actors	
Entry condition *(The event that triggers the Use Case)*	
Flow of events	
Exit condition *(conditions that are satisfied after the Use Case finishes)*	
Special requirements *(any constraints etc that are not related to the function of the system)*	

Example of a Use Case description for part of a Library system

An *example* of a Use Case description for *part of a Library system* is shown below. The example text provides you with an indication of **how** to express the content.

Use Case Name	*Borrow a Book*
Participating Actors	`BookBorrower`
Entry condition *(The event that triggers the Use Case)*	The `BookBorrower` selects the *Borrow a book* option on the screen menu.
Flow of events	1. The system responds by displaying a new screen (New book Loan) 2. The `BookBorrower` enters his/her member's ID number. 3. The system checks if the `BookBorrower` has already got the maximum number of Books on loan. 4. If the `BookBorrower` has already reached his/her limit, the system displays an appropriate message.

	5. If the `BookBorrower` has not reached his limit, the system asks for the ID number of the Book. 6. The system checks to see if the Book is available. 7. If the Book is not available, the system displays an appropriate message. 8. If the Book is available, then the `BookBorrower` records its loan and records the BOOK as no longer available. 9. The system asks for confirmation of the loan. 10. The `BookBorrower` confirms or cancels the loan. 11. The system returns to the main menu, with a confirmatory message of the loan made.
Exit condition *(conditions that are satisfied after the Use Case finishes)*	If the loan was confirmed then it is recorded in the system otherwise, no change is made to the system.
Special requirements *(any constraints etc that are not related to the function of the system)*	None.

Guidance

The assessment of your project will depend in part upon the quality of the documentation that you have produced.

- Restate the specification of the assignment by listing the required outcomes.
- Take the time required to design the assignment before you type any program code. Follow an appropriate design and documentation sequence.
- Always document your designs before you implement them.
- Keep the Object Definition Sheets up-to-date and do basic tests at each stage of the development.
- Provide a detailed design including, where appropriate, the design of any algorithms.
- Build in error handling to involve meaningful messages that would help with any future maintenance of the software.
- Annotate all implementations.
- Design a testing strategy.
- Justify the design of suitable comprehensive test data.
- Show evidence of testing.
- Where appropriate, detail any major corrective action that you have taken in the light of the testing process.

Submission requirements

A word-processed document must be submitted incorporating the full documentation of all the significant aspects of the development of the assignment above. The document should be submitted both in paper form and digital form.

Refer to the Guidance above when producing your final documentation.

You are required to submit a publishable copy of the compiled system together with installation notes.

This publishable copy, which includes a setup file, should be on an appropriate medium (CD, USB flash drive, etc.).

Candidate checklist

Please use the following checklist to ensure that your work is ready for submission.

Have you read the NCC Education document *Academic Misconduct Policy* and ensured that you have acknowledged all the sources that you have used in your work?	❑
Have you completed the *Statement and Confirmation of Own Work* form and attached it to your assignment? **You must do this.**	❑
Have you ensured that your work does not contain viruses and can be run directly?	❑