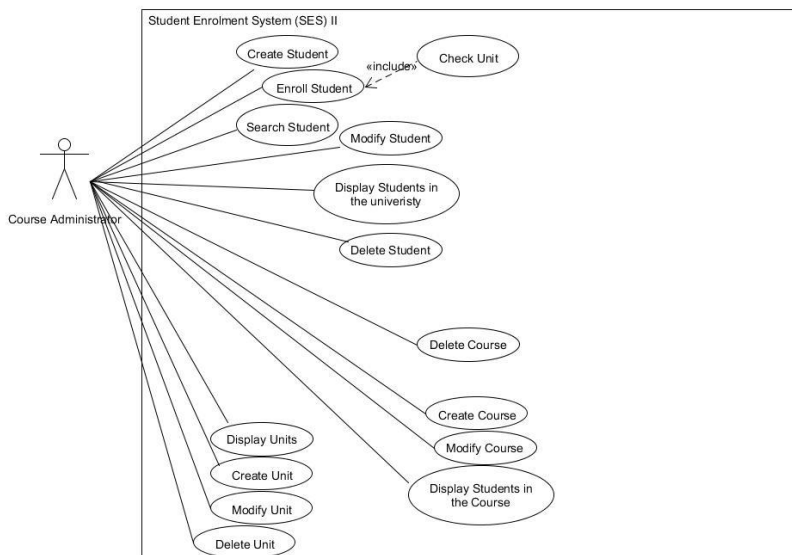


The Business Model

- Description of the business processes of an organization
- Developer can obtain this understand by conducting interviews
 - Types of interviews
 - **Structured**: specific pre-planned questions, tend to be close-ended (used to get specific info)
 - **Unstructured**: may start with 1 or 2 pre-planned questions but the follow up questions relate to the answers given, tend to have more open –ended questions (asked to encourage the interviewee to speak out)
 - Interviewer must be
 - Familiar with application domain
 - Open to new info
 - Produce written report outlining the results of the interview
- Other ways of gaining information
 - Questionnaires
 - Looking at existing forms
 - Direct observation

UML – Use Case Diagram



- Each bubble is a function of the system
- Actor is the user of the system
- Includes: A includes B. a can't be considered complete until action B is complete
- Extends: B extends A, Then B is optional variation of A
- Use case is not
 - A flowchart
 - Depiction of internal logic

Initial Requirements = functional requirements + non-functional requirements

- Functional Requirements
 - Specifies an action the target product must be able to perform
 - The functionality
 - Handled within the requirements and analysis workflows

- Non-functional Requirements
 - Specifies the properties of the product
 - Platform constraints
 - Response times
 - Reliability
 - May have to wait till the devising workflow to deal with this
- Gathering of requirements
 - Understanding the structure of the organization
 - Identifying very broad IT goals
 - Strategic plan
 - Identifying 'events'
 - Identifying 'things'

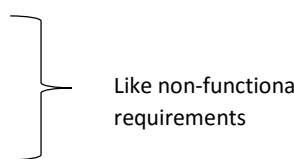
The Analysis Workflow

- Use case driven
- During this workflow, classes are extracted

Aim:

- Obtain deeper understanding of requirements
- Describe them in a way that requirements can be maintained

The Specification Document

- Informal enough for the client, but formal enough for the developers (to use for the design)
- A contract between the client and the developers
- Typical constraints
 - Deadline
 - Parallel running
 - Portability
 - Response time
- If final product passes tests, deemed to have satisfied specifications
- Informal Specification
 - Written in a natural language
 - These are nucleuse, ambiguous, poor style

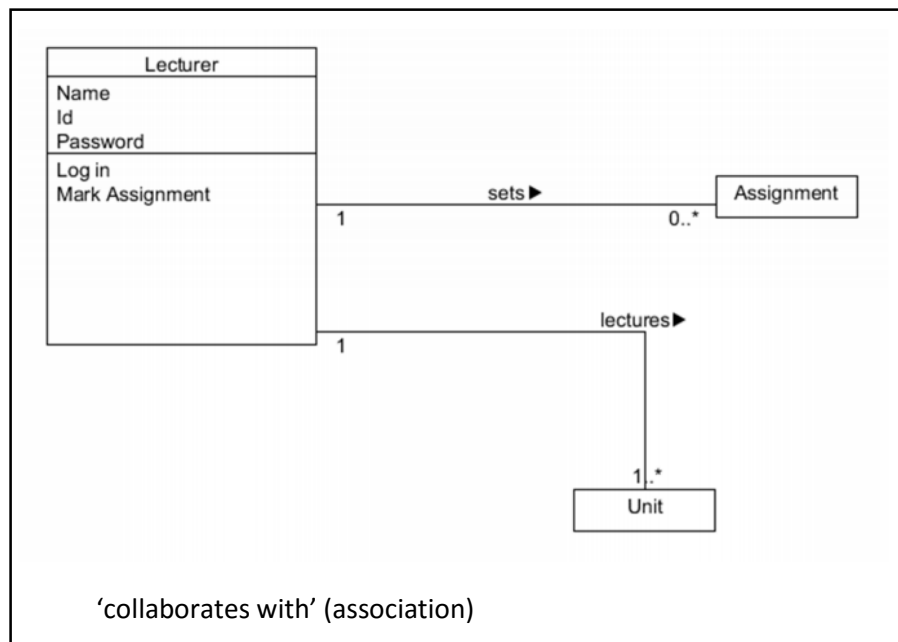
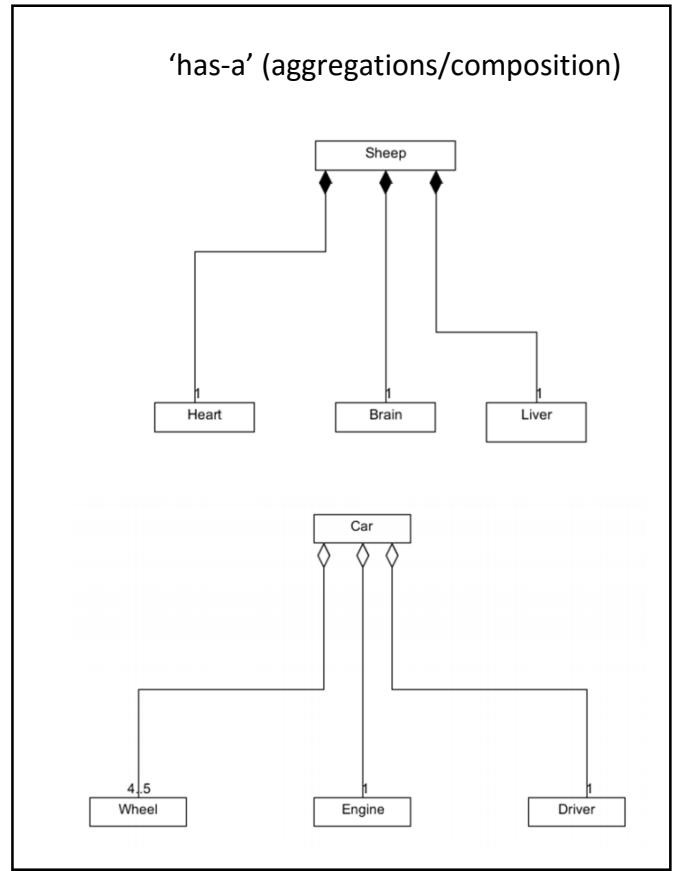
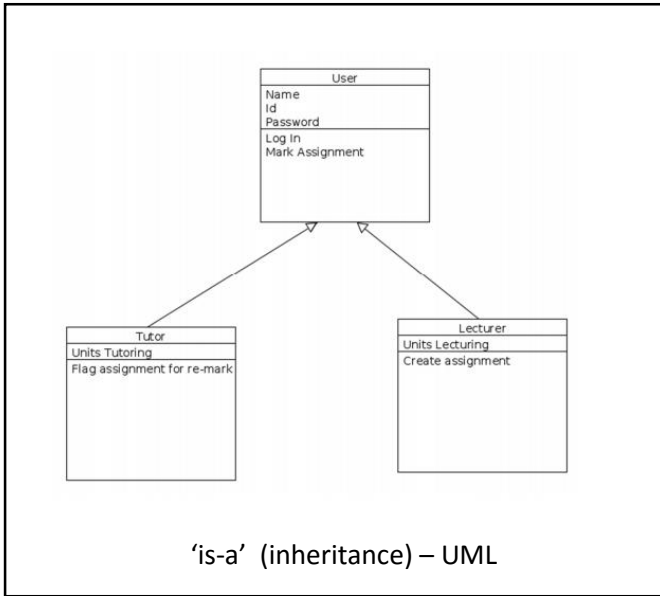
Extracting the Classes

- **Functional Modelling**
 - Use case: generic description of overall functionality
 - Scenario: specific instantiation of a use case
 - One use case can have multiple scenarios

- Normal Scenario
 - Set of interactions between user and system that corresponds to the way we understand how the system should be used towards a good
- Exception Scenario
 - Scenario that describes how to handle an undesired event, an even that interferes with progress towards a good

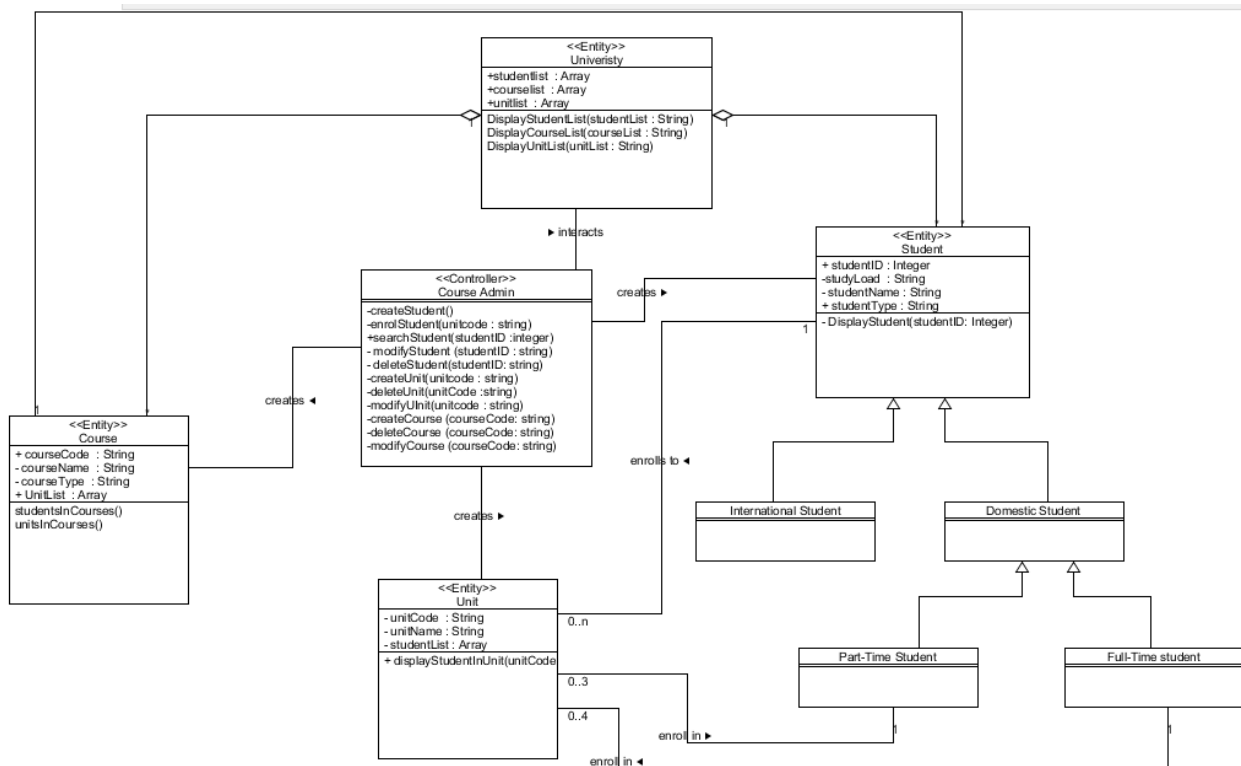
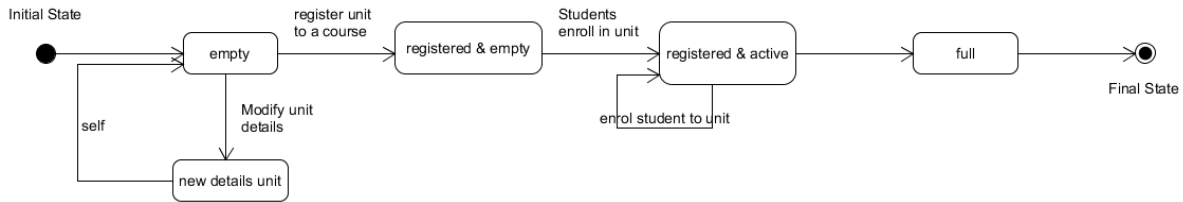
Use Case Name	Enrol Student	
Scenario	Normal scenario	
Trigger	Administrator requests enrolling a student	
Brief description	The administrator enrolls a student by choosing a unit from the units registered to a course type in which the student is enrolled	
Actors	Administrator	
Related use cases	N/A	
Preconditions	The student record exists . Also, courses and units must exist.	
Postconditions	Student is successfully enrolled into units	
Flow of events	Actor	System
	1. Administrator requests a new student enrolment 3. Administrator enters student id 5. Administrator enters one unit code one by one	2. System prompts the administrator to enter a student id 4. System displays applicable units to students particular course. System also checks whether given student is a part-time or full-time student and displays how many units the student can be enrolled into. 6. After each unit code is entered , display that student has been successfully enrolled into the unit
Exception conditions	3.1 If students is not present or have invalid characters in the student record the system will display "invalid student id. Please try again" and it should return back to the prompt "Enter student id" 5.1 If students tries to enroll in more than their expected number of units a warning sign will be displayed saying "You are unable to enroll in any more units." 5.2 If the Administrator tries to enter more than one unit at a time the system will display" Enter units one by one." 5.3 If the Administrator enters an invalid unit the system will display "Invalid Unit. Please enter unit code."	

- Class Modelling
 - Determine the entity classes and their attributes
 - Determine the inter-relationships and iterations between the entity classes
 - Present this info in the form a class diagram
 - Associations between them



- Dynamic Modelling

- Determine the operations performed by, or to each entity class
- Present this info in form of a statechart diagram

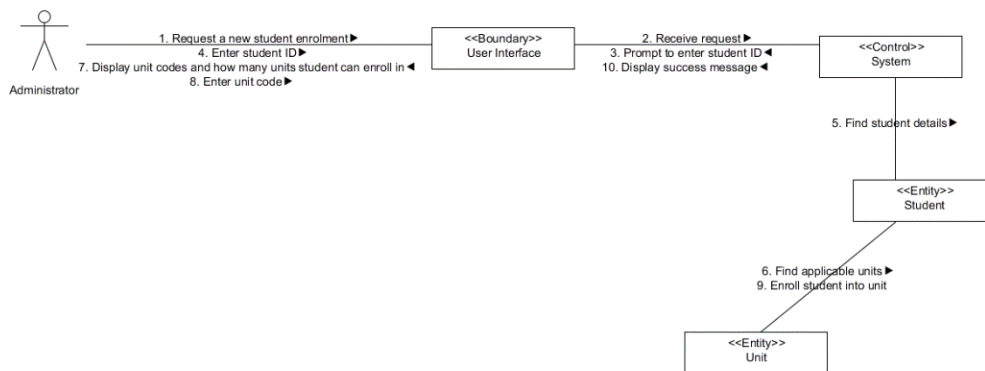


Things to look for in class diagram

- Visibility on diagrams + , - (methods)
- Return types on methods
- Ensure children class have closed arrow head
- Methods repeating in different classes mean high coupling
- If a class has no methods and few attributes its probably not a class but an attribute on another class

Use-case Realization

- The realization of a specific scenario of a use case is depicted using an interaction diagram
- Iteration Diagram
 - **Communication Diagram**
 - Client will not understand the specification so a written description of the diagram as the 'flow of Events'
 - Adv
 - Can see what the communication is between objects
 - Dis
 - Lots of step so it's hard to them clearly on the diagram
 - Hard to trace the order of steps



- **Sequence Diagram**
 - Order of step shown visually
 - Relationship between classes in not clear
 - Sometimes hard to layout classes horizontally across the top

