



IT243 – System Analysis and Design

Assignment 4 (40 Points)

Note:

- **Submission Deadline: 31st December, 2016 (11:59 pm)**
- The following questions are from Chapters: 10, 11 and 14.
- Copied assignment will be graded zero mark.
- Assignment submitted after due date will not be accepted, it will be considered to be late and will have zero mark.
- Assignments must be submitted through BB only.
- Emailed assignments will NOT be considered.

Part 1

Question 1: Explain and give an example for the following concepts (6 Marks) **accept all correct examples of these three concepts.**

1. Encapsulation.

Encapsulation is simply the combination of process and data into a single entity. In this context the **principle of information hiding** suggests that only the information required to use a software module be published to the user of the module.

2. Inheritance

Classes are arranged in a hierarchy

- Superclasses or general classes are at the top
- Subclasses or specific classes are at the bottom (Subclasses inherit attributes and methods from the superclasses above them)

3. Polymorphism.

Polymorphism is the characteristic of being able to assign a different meaning or usage to something in different contexts - specifically, to allow an entity such as a variable, a function, or an object to have more than one form.

Question 2: Why is it desirable for a structure chart to be highly cohesive and loosely coupled? (6 Marks)

Good structure charts, and well-designed systems are highly cohesive and loosely coupled. Cohesion in modules refers to how many functions a particular module performs. A module that only performs one function—for instance, a module that calculates the total price of goods sold would be a highly cohesive module. The reason high cohesion is desirable is because it means that the module is more portable, and adaptable as the application develops. A module that calculates the total price can be applied to a cash

register sale in a physical store, or a catalog, or Internet sale. Whereas, a low cohesion module—like one that calculates the total sale and the sales tax could not be used outside of the physical store.

Coupling refers to how independent each module is from the others. Modules that are able to perform a function without relying on other modules are loosely coupled. Loose coupling is desirable because it prevents issues with a change in one module cascading through the application, and causing problems. Using the in-class example, if a module calculates the date with a two-digit year, if other modules rely on that function, and the date format is changed to a four-digit year, it could affect the ability for the other functions to calculate.

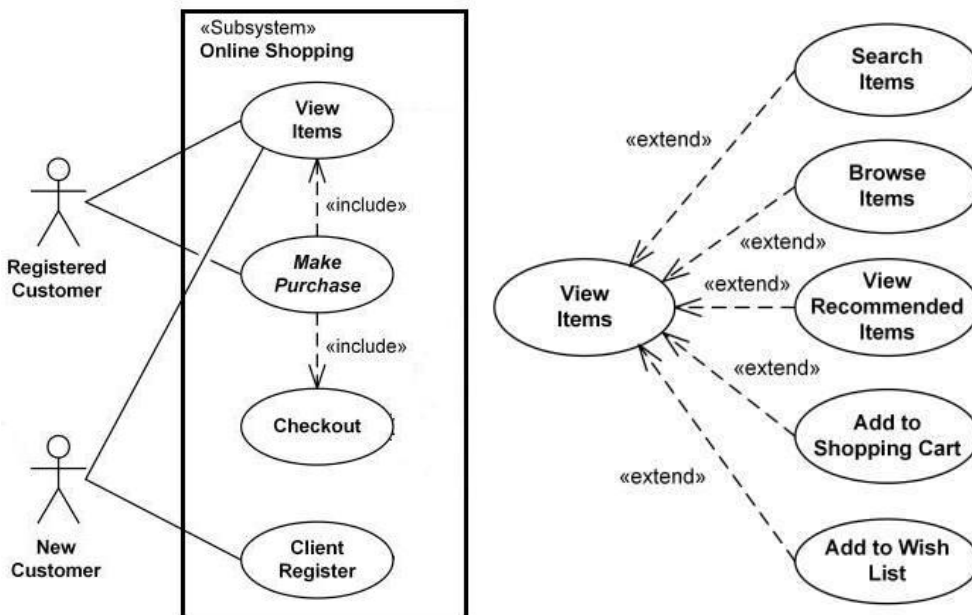
Part 2

Question 3: Draw a use case diagram for the system described below: (10 Marks)

Web Customer uses our web site to make purchases online. Top level functionality given by the web site are View Items, Make Purchase and Client Register. View Items use case could be used by customer as top level use case if customer only wants to find and see some products before registration. This use case could also be used as a part of Make Purchase use case after registration. Client Register use case allows customer to register on the web site to Make Purchase. Note, that Checkout use case is part of making purchase.

View Items use case have several optional use cases - customer may search for items, browse items, view items recommended for him/her, add items to shopping cart or wish list.

Solution

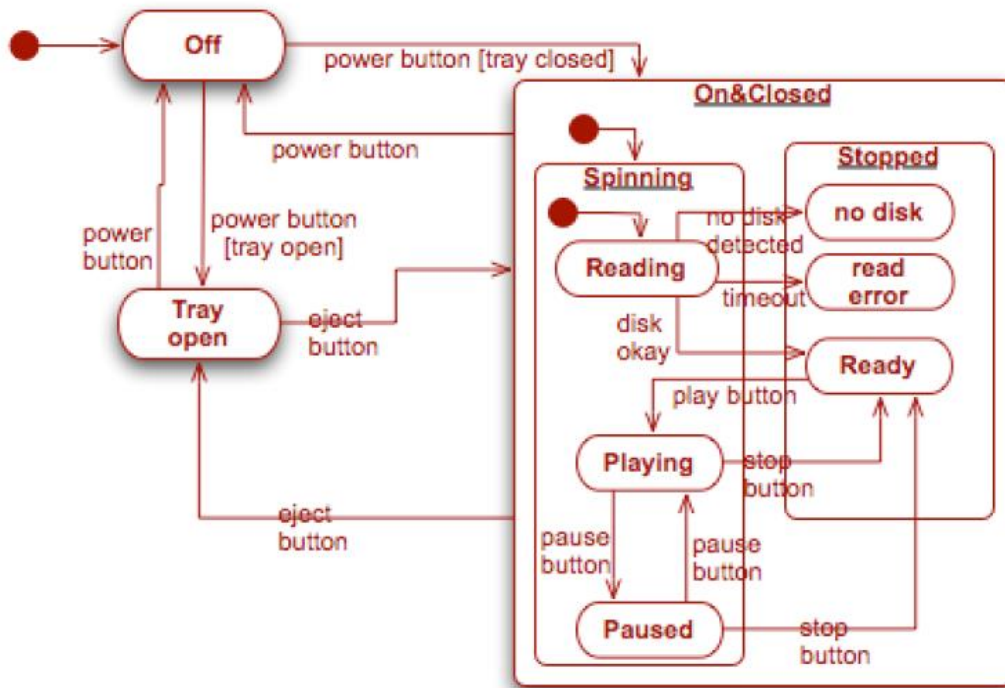


Question 4: Draw a State Machine Diagram to show the following behaviors of a DVD player. (9 Marks)

Be sure to label all transitions with the events that cause them. The DVD player is initially off. It can be turned on or off by pressing the power button. When it is off, none of the other buttons do anything. When it is on, the tray can be opened or closed by pressing the eject button. When it is open, only the power and eject buttons will do anything. When the tray is closed, the motor is either spinning or stopped. When you turn on the power or close the tray, the machine first spins up the motor to read the disk. It then gets one of three signals:

- (a) no disk – it stops the motor, and disables all buttons other than eject and power.
- (b) timeout, indicating it was unable to read the disk – it stops the motor, displays an error message, and disables all buttons other than eject and power.
- (c) disk okay – it stops the motor, but enables the play control buttons. There are three play control buttons. Play starts the DVD playing, spinning up the motor if necessary. Pause toggles between playing and paused, without stopping the motor. Pause does nothing if pressed when the motor is stopped. Stop stops the motor, if it was spinning.

Solution



Question 5: Draw a sequence diagram to illustrate the normal operation of the Use Case for “Change DVDs” for the DVD player described in the previous question. (9 Marks)

Your sequence should start in the Playing state, and describe the scenario in which the user ejects the current disk, inserts a new one, and selects play again. Assume the DVD player software has to send signals to the motor (to tell it when to spin) and to the tray (to open and close), and in both cases needs to await confirmation from the device before doing anything else. Use activation bars to indicate when the devices are active. State any further assumptions you make.

Solution

