

NextGen POS



Problem Description

The POS system is a computerized application used (in part) to record sales and handle payments; it is typically used in retail stores. It includes hardware components such as computer and bar code scanner, and software to run the system.

It interfaces to various service applications, such as third-party tax calculator and inventory control. These systems must be relatively fault-tolerant; that is, even if remote services are temporarily unavailable (such as the inventory system), they must still be capable of capturing sales and handling at least cash payments (so that the business is not crippled).

A POS system increasingly must support multiple and varied client side terminals and interfaces. These include a thin-client Web browser terminal, a regular personal computer with something like Java Swing graphical user interface, touch screen input, wireless PDAs, and so forth.

Furthermore, we are creating a commercial POS system, that we will sell to different clients with disparate needs in terms of business rule processing. Each client will desire a unique set of logic to execute at certain predictable points in scenarios of using the system, such as when a new sale is initiated or when a new line item is added. Therefore, we will need a mechanism to provide this flexibility and customization.

Source: "Applying UML and Patterns 2nd Ed.", Craig Larman, Prentice Hall, 2002

[Use Cases](#) | [Case Studies](#)

Last modified: 26.02.2002 , [M.Kropp](#)