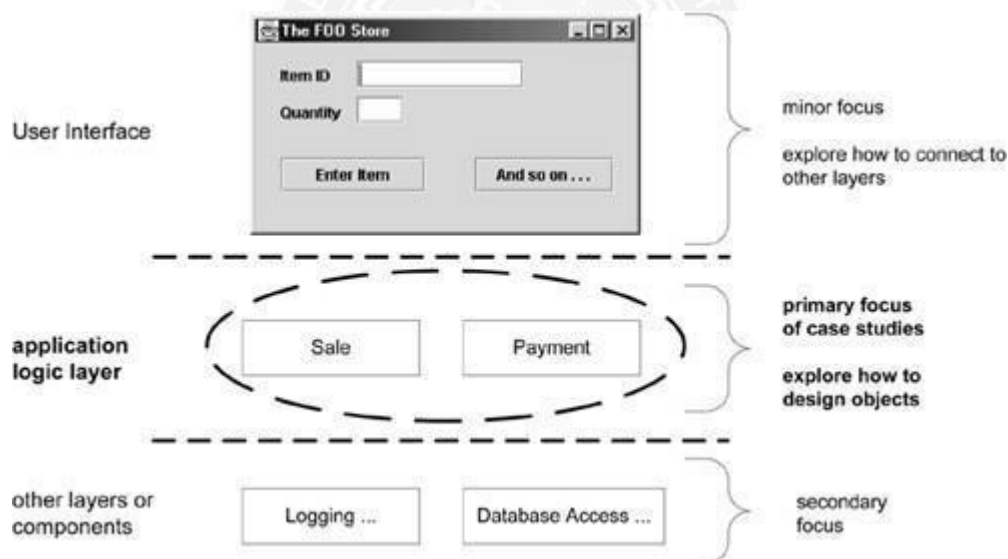


CASE STUDY

Applications include UI elements, core application logic, database access, and collaboration with external software or hardware components. This introduction to OOA/D focuses on the core application logic layer, with some secondary discussion of the other layers.

Why focus on OOA/D in the core application logic layer?

- Other layers are usually technology/platform dependent. For example, to explore the OO design of a Web UI or rich client UI layer in Java, we would need to learn in detail about a framework such as Struts or Swing.
- In contrast, the OO design of the core logic layer is similar across technologies.
- The essential OO design skills learned in the context of the application logic layer are applicable to all other layers or components.
- The design approach/patterns for the other layers tends to change quickly as new frameworks or technologies emerge.



Sample layers and objects in an object-oriented system, and the case study focus.

NEXTGEN POS SYSTEM

The NextGen POS(Point of Sale) System

- A POS system is a computerized application used (in part) to record sales and handle payments; it is typically used in a retail store.
- It includes hardware components such as a computer and bar code scanner, and software to run the system.
- It interfaces to various service applications, such as a 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 a Java Swing graphical user interface, touch screen input, wireless PDAs, and so forth.
- 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.

Using an iterative development strategy, we are going to proceed through requirements, object-oriented analysis, design, and implementation.

INCEPTION

Inception : The purpose of the inception phase is not to define all the requirements, or generate a believable estimate or project plan. Most requirements analysis occurs during the elaboration phase, in parallel with early production-quality programming and testing.

- Inception (in one sentence) - Envision the product scope, vision, and business case.
- It may include the first requirements workshop, planning for the first iteration, and then quickly moving forward to elaboration. Common inception artifacts and indicates the issues they address.

- For example, the Use-Case Model may list the names of most of the expected use cases and actors, but perhaps only describe 10% of the use cases in detail done in the service of developing a rough high-level vision of the system scope, purpose, and risks.
- Note that some programming work may occur in inception in order to create "proof of concept" prototypes, to clarify a few requirements via UI-oriented prototypes, and to do programming experiments for key "show stopper" technical questions. **Table -Sample inception artifacts.**

Artifact	Comment
Vision and Business Case	Describes the high-level goals and constraints, the business case, and provides an executive summary.
Use-Case Model	Describes the functional requirements. During inception, the names of most use cases will be identified, and perhaps 10% of the use cases will be analyzed in detail.
Supplementary Specification	Describes other requirements, mostly non-functional. During inception, it is useful to have some idea of the key non-functional requirements that have will have a major impact on the architecture.
Glossary	Key domain terminology, and data dictionary.
Risk List & Risk Management Plan	Describes the risks (business, technical, resource, schedule) and ideas for their mitigation or response.
Prototypes and proof-of-concepts	To clarify the vision, and validate technical ideas.
Iteration Plan	Describes what to do in the first elaboration iteration.
Phase Plan & Software Development Plan	Low-precision guess for elaboration phase duration and effort. Tools, people, education, and other resources.
Development Case	A description of the customized UP steps and artifacts for this project. In the UP, one always customizes it for the project.

Activities in Inception:

Inception is a short step to elaboration. It determines basic feasibility, risk, and scope, to decide if the project is worth more serious investigation.

Activities and artifacts in inception include:

- a short requirements workshop

- most actors, goals, and use cases named
- most use cases written in brief format; 10-20% of the use cases are written in fully dressed detail to improve understanding of the scope and complexity
- most influential and risky quality requirements identified
- version one of the Vision and Supplementary Specification written
- risk list
- Technical proof-of-concept prototypes and other investigations to explore the technical feasibility of special requirements
- user interface-oriented prototypes to clarify the vision of functional requirements
- recommendations on what components to buy/build/reuse, to be refined in elaboration
 - For example, a recommendation to buy a tax calculation package.
 - high-level candidate architecture and components proposed
 - This is not a detailed architectural description, and it is not meant to be final or correct. Rather, it is brief speculation to use as a starting point of investigation in elaboration. For example, "A Java client-side application, no application server, Oracle for the database," In elaboration, it may be proven worthy, or discovered to be a poor idea and rejected.
- plan for the first iteration