

At the end of the lesson, students should be able to:

1. Describe systems design and contrast it with systems analysis
2. List documents and models used as inputs to or output from systems design
3. Explain each major design activity
4. Describe security methods and controls

Overview

This topic focuses on the solution system. During analysis, the focus is on *understanding* what the system should do (i.e., the requirements), whereas during design, the focus is on the *solution* (i.e., specifying how the system will be built and what the structural components of the new system will be). New developers often ask, “When are these tasks carried out in a real project?” Unfortunately, there is no single answer.

Many projects begin with some of the design decisions already being made, particularly about the deployment environment, when companies already have a strong technology infrastructure. For other projects, the new system may be the result of a new thrust for the organization, and thus the decisions are wide open. However, it is normal for the project team to start thinking about these issues very early in development and to begin making preliminary decisions as requirements are being defined.

The topics discussed in this lesson are solution-oriented design topics. However, it would help if you didn’t try to come up with a complete solution until you understand the problem.

What is Systems Design?

- Analysis provides the starting point for the design
- Design provides the starting point for implementation
- Analysis and design results are documented to coordinate the work
- The objective of design is to define, organize, and structure the components of the final solution to serve as a blueprint for construction

Analysis Design Implementation

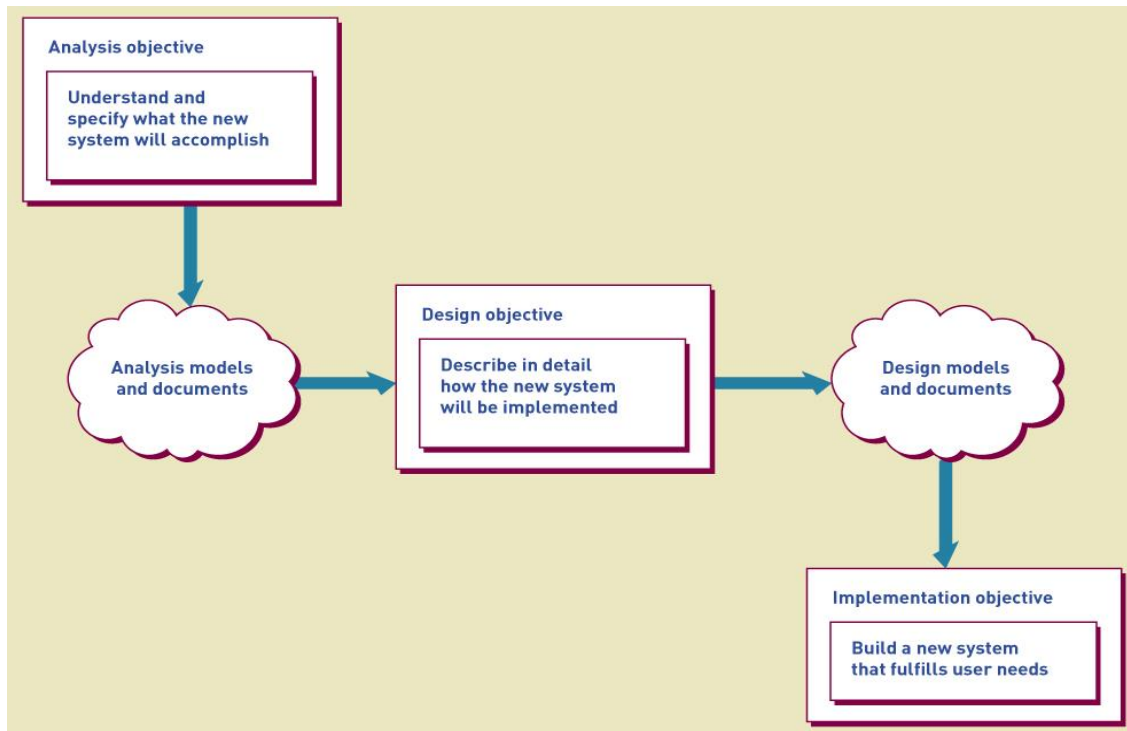


Fig. 1. Models and documents link analysis, design, and implementation

Design Models

- Design is a model-building activity
- The formality of the project will dictate the type, complexity, and depth of models
- Agile/iteration projects typically build fewer models, but models are still created
- Jumping to programming without design often causes less than optimum solutions and may require rework

Analysis Models to Design Models

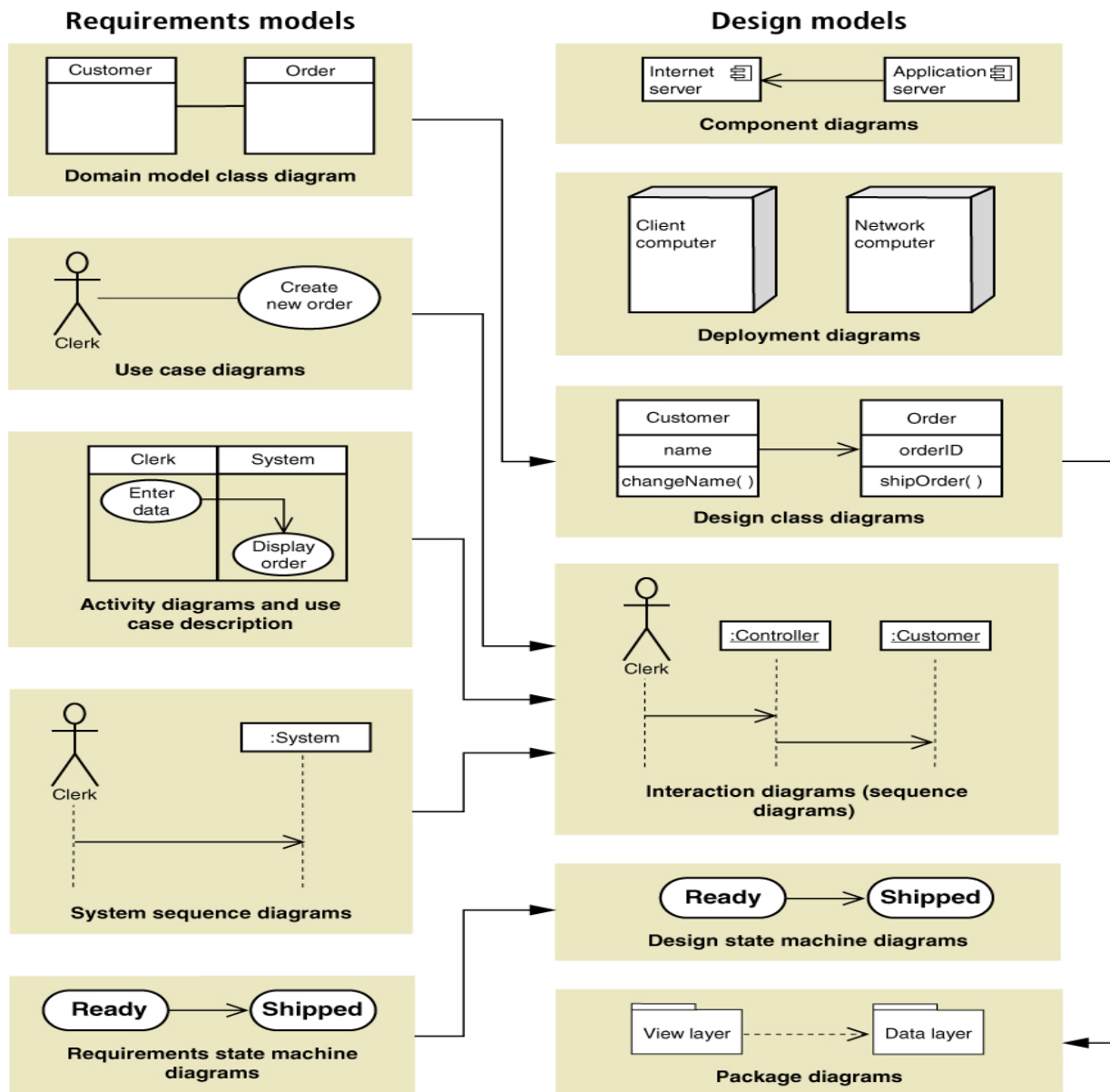
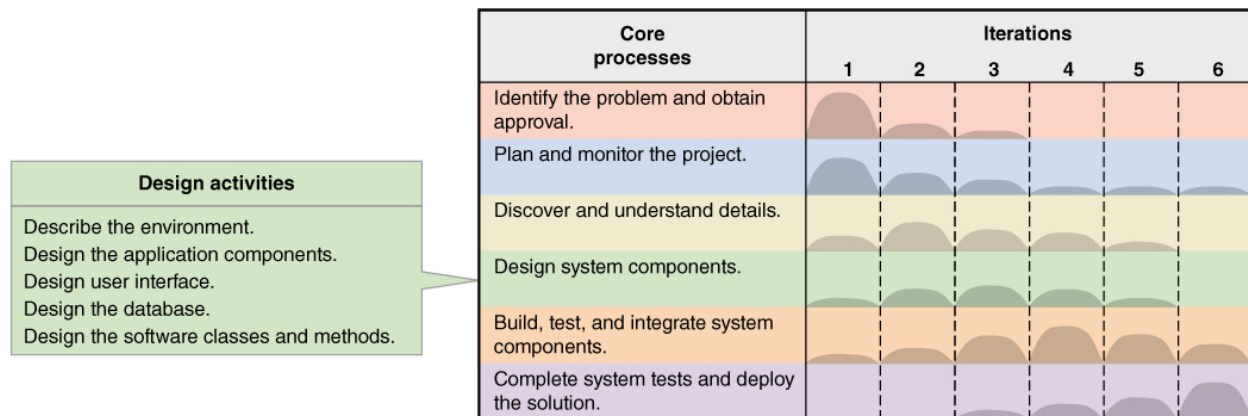


Fig. 2.

Design Activities

- Design activities correspond to components of the new system
 - The environment
 - Application components
 - User interface
 - Database
 - Software classes and methods

Design Activities and Iterations



Key Design Questions for each Activity

Design activity	Key question
Describe the environment	How will this system interact with other systems and with the organization's existing technologies?
Design the application components	What are the key parts of the information system and how will they interact when the system is deployed?
Design the user interface	How will users interact with the information system?
Design the database	How will data be captured, structured, and stored for later use by the information system?
Design the software classes and methods	What internal structure for each application component will ensure efficient construction, rapid deployment, and reliable operation?

Describe the Environment

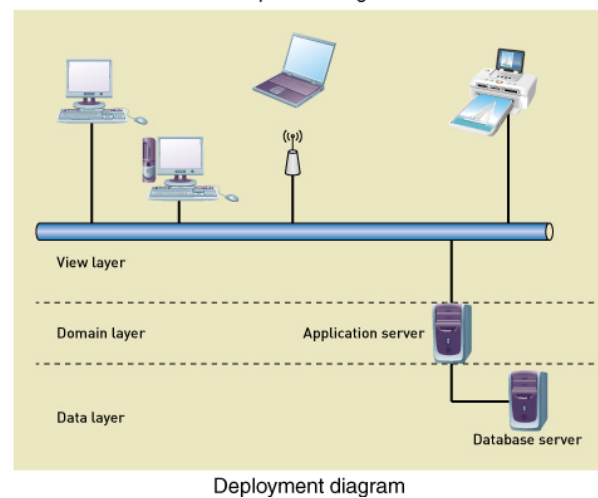
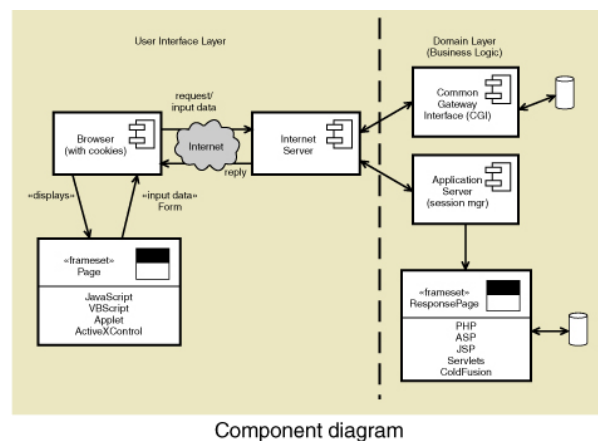
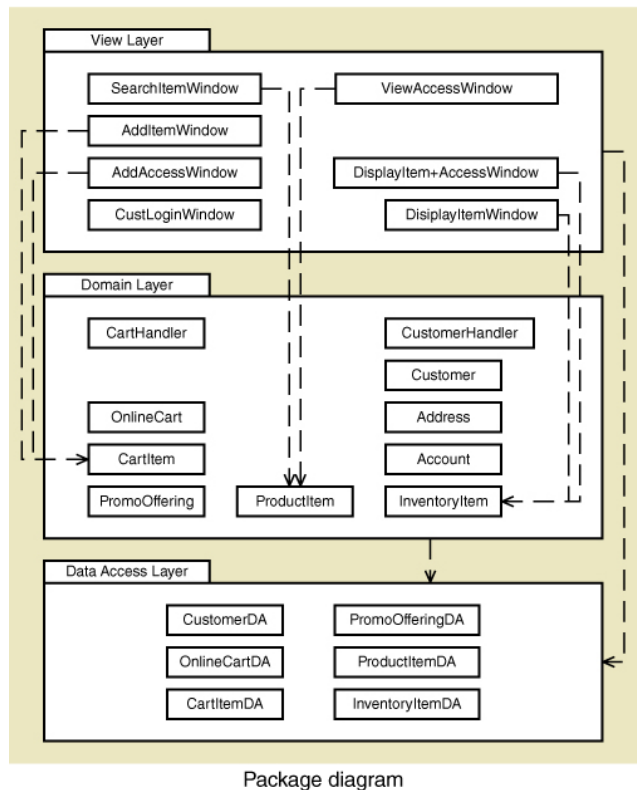
- Two key elements in the environment
 - Communications with External Systems
 - Message formats
 - Web and networks
 - Communication protocols
 - Security methods
 - Error detection and recovery
 - Conforming to an existing Technology Architecture

- Discover and describe the existing architecture
- Chapter 7 provides more details

Design the Application Components

- Application component is a well-defined unit of software that performs some function(s)
- Issues involve how to package components including
 - Scope and size – what are the functions, boundaries, and interfaces?
 - Programming language – what are the accepted languages?
 - Build or buy – is an acceptable version available to purchase?

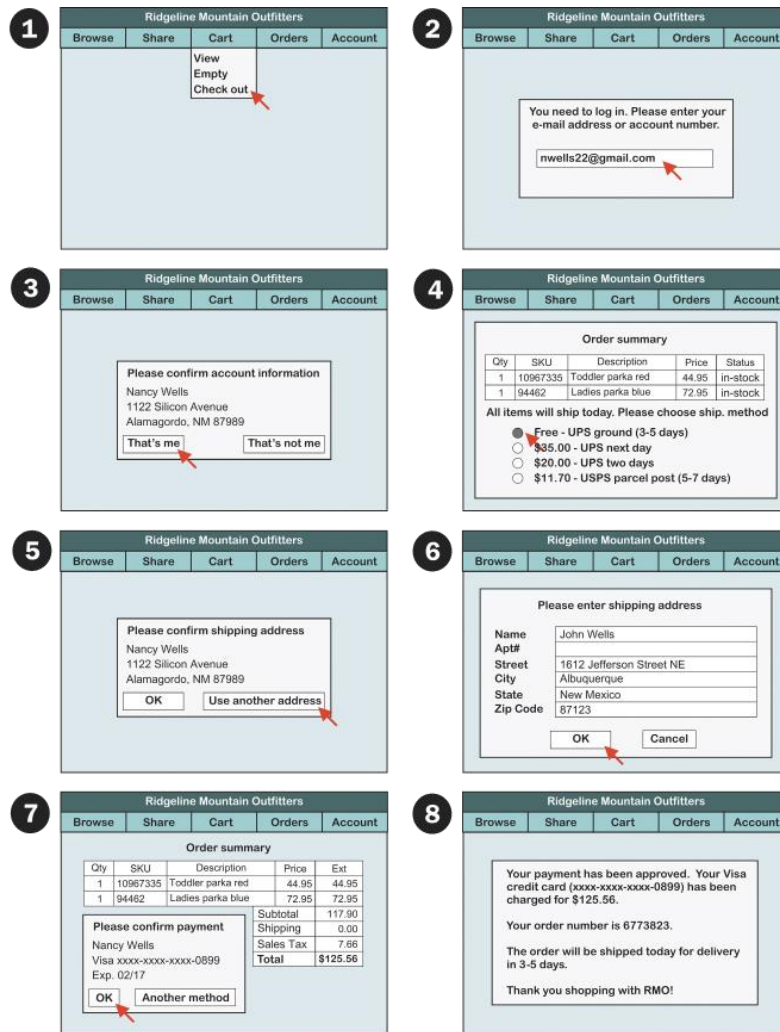
Typical models for defining application components



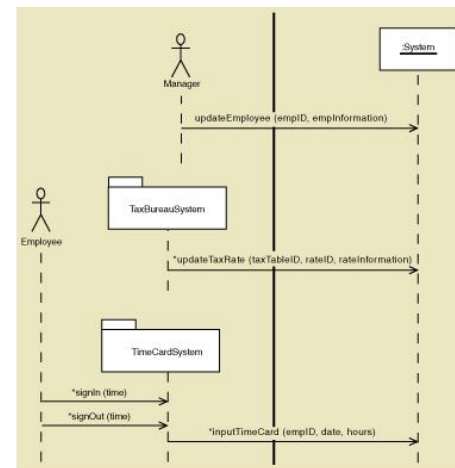
Design the User Interface

- To the user, the User Interface **is** the system.
- The user interface has a large impact on user productivity
- Includes both Analysis and Design tasks
 - Requires heavy user involvement
- Current needs require multiple user interfaces
- Many different devices and environments

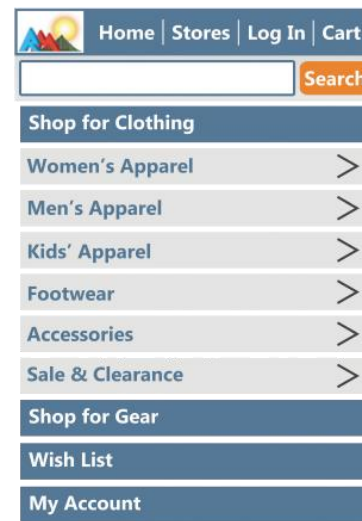
Typical Models for User Interface Design



Storyboard



System sequence diagram

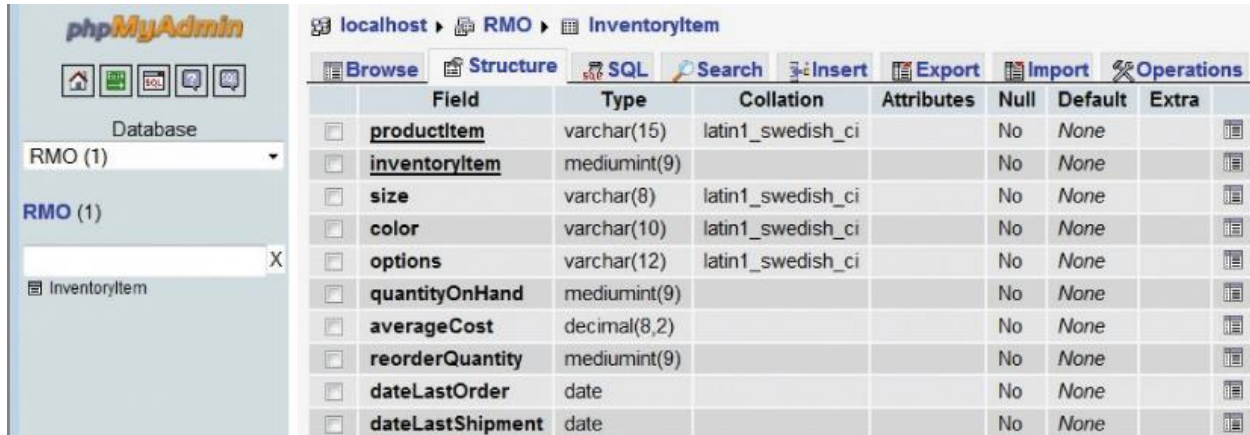


Small screen menu prototype

Design the Database

- By definition, an Information System requires data – usually in a database
- Current technology frequently uses Relational Database Management Systems (R D B M S)
- Requires converting the data model to a relational database
- Requires addressing of many other technical issues
 - Throughput and response time
 - Security

Typical Table Definition as part of Database Schema



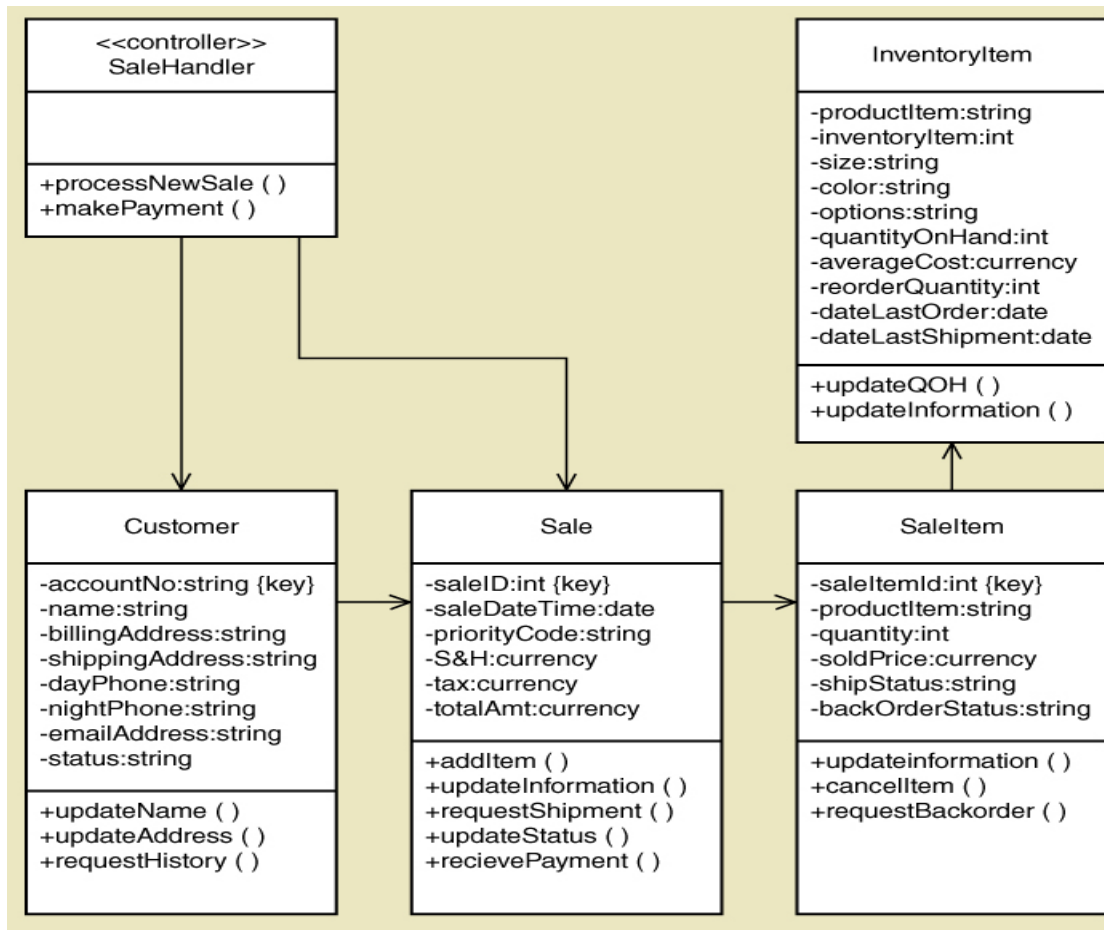
The screenshot shows the phpMyAdmin interface. On the left, the 'Database' dropdown is set to 'RMO (1)', and the 'InventoryItem' table is selected. The main panel displays the 'Structure' tab for the 'InventoryItem' table. The table has 11 fields, each with a checkbox for selection. The fields are: productItem (varchar(15)), inventoryItem (mediumint(9)), size (varchar(8)), color (varchar(10)), options (varchar(12)), quantityOnHand (mediumint(9)), averageCost (decimal(8,2)), reorderQuantity (mediumint(9)), dateLastOrder (date), and dateLastShipment (date). All fields are set to 'No' for Null, 'None' for Default, and have no Extra attributes.

	Field	Type	Collation	Attributes	Null	Default	Extra
<input type="checkbox"/>	<u>productItem</u>	varchar(15)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	<u>inventoryItem</u>	mediumint(9)			No	None	
<input type="checkbox"/>	size	varchar(8)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	color	varchar(10)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	options	varchar(12)	latin1_swedish_ci		No	None	
<input type="checkbox"/>	quantityOnHand	mediumint(9)			No	None	
<input type="checkbox"/>	averageCost	decimal(8,2)			No	None	
<input type="checkbox"/>	reorderQuantity	mediumint(9)			No	None	
<input type="checkbox"/>	dateLastOrder	date			No	None	
<input type="checkbox"/>	dateLastShipment	date			No	None	

Design Software Classes and Methods

- A K A Detailed Design
- A model building activity
 - Design Class Diagram
 - Sequence Diagrams
 - State-Machine Diagrams

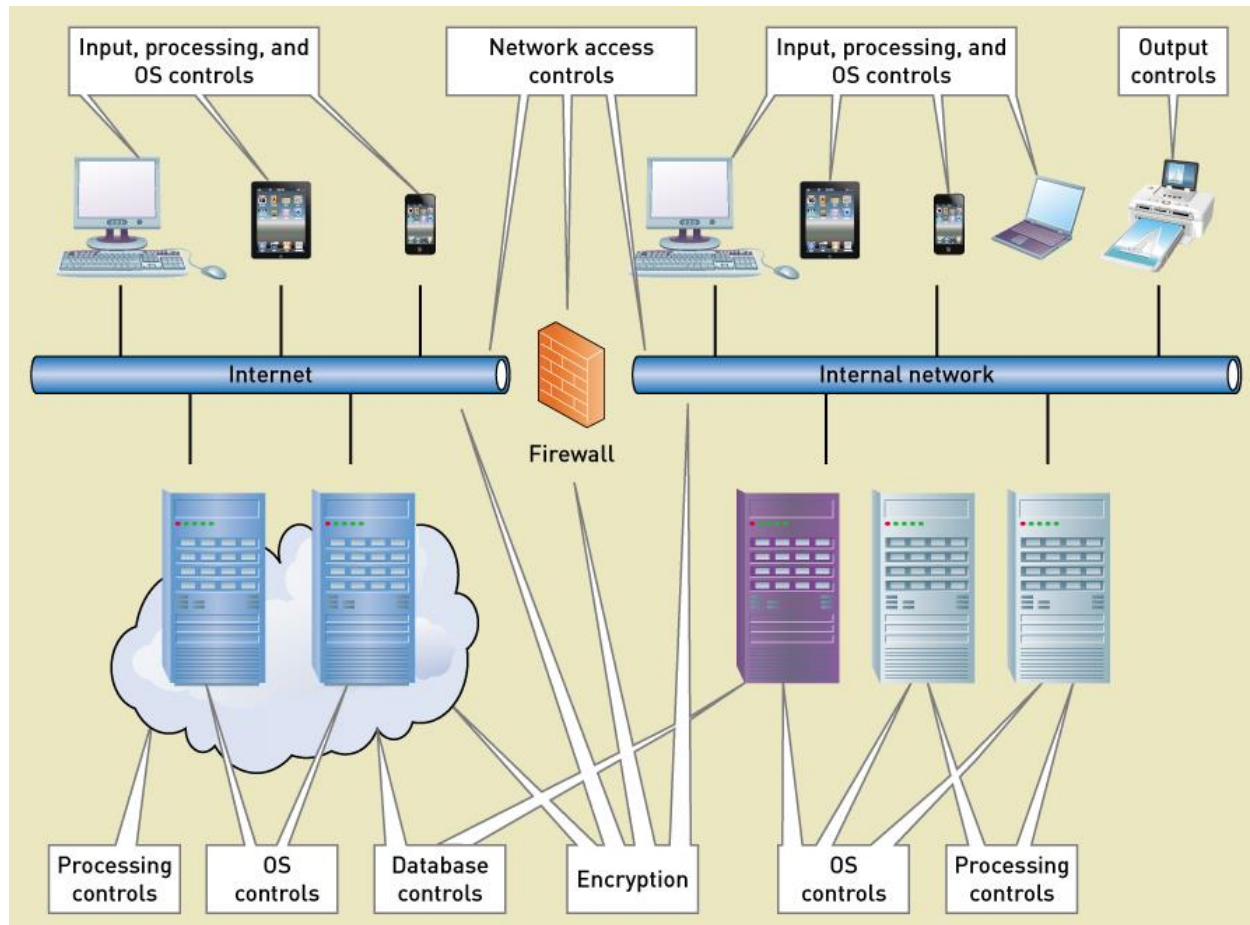
Typical Design Class Diagram



System Controls and Security

- Integrity Controls
 - Controls that maintain integrity of inputs, outputs and data and programs
- Security Controls
 - Controls that protect the assets from threats, internal and external

Integrity and Security Controls



Designing Integrity Controls

- Integrated into application programs and D B M S
- Objectives of Integrity Controls
 - Ensure that only appropriate and correct business transactions are accepted
 - Ensure that transactions are recorded and processed correctly
 - To protect and safeguard assets such as the database

Input Controls

- Prevent invalid or erroneous data from entering the system
- Value Limit Controls
 - Check the range of inputs for reasonableness
- Completeness Controls
 - Ensure all the data has been entered
- Data Validation Controls
 - Ensure that specific data values are correct
- Field Combination Controls
 - Ensure data is correct based on relationships between fields

Output Controls

- To ensure that output arrives at proper destination (for authorized eyes) and is accurate, current, and complete
- Examples
 - Physical access to printers and display devices
 - Discarded data – protect from “dumpster diving”
 - Labels on printed and electronic output to correctly identify source of data

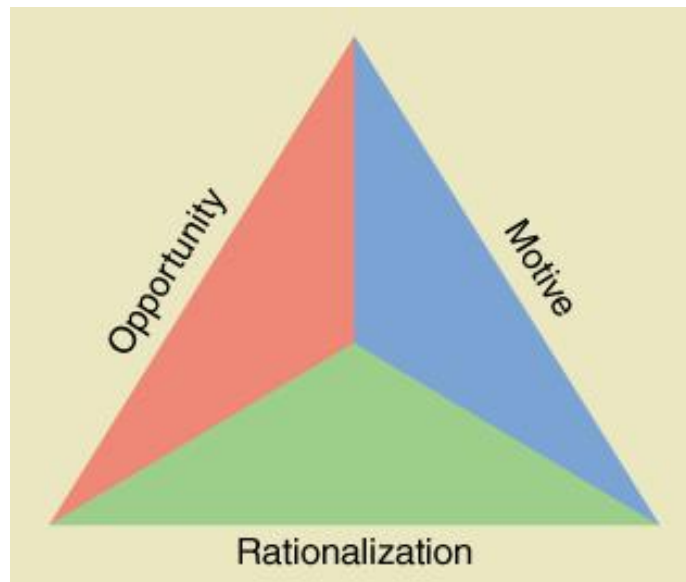
Redundancy, Backup and Recovery

- Protect data and systems from catastrophes
 - Databases
 - Hardware
 - Software applications
 - Networks
- On-site versus off-site copies

Fraud Prevention

- Critical to prevent internal fraud, embezzlement, or loss
 - Fraud triangle
 - Opportunity
 - Motive

Rationalization



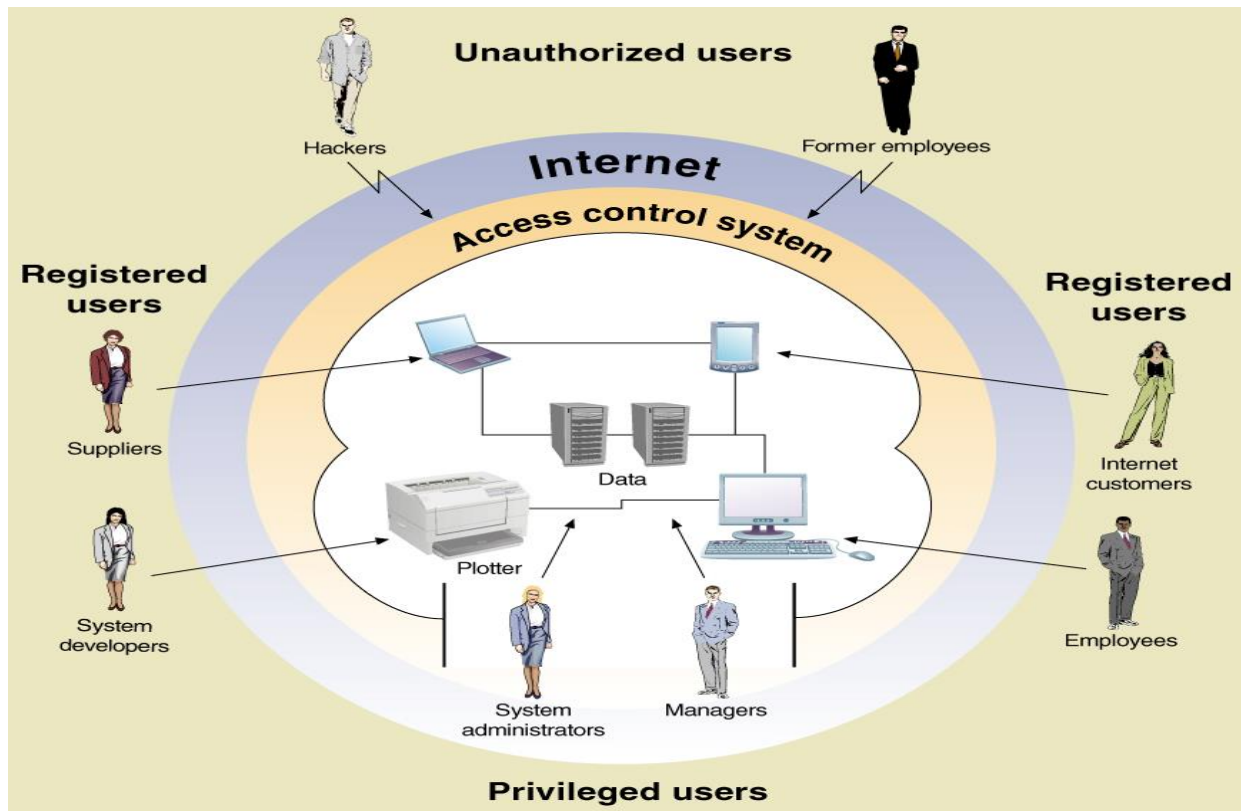
Fraud Risk – Factors and Techniques

Factors affecting fraud risk	Risk-reduction techniques
Separation of duties	Design systems so those with asset custody have limited access to related records. Also, ensure that no one has sufficient system access to commit and cover up a fraud.
Records and audit trails	Record all transactions and changes in asset status. Log all changes to records and databases, and restrict log access to a few trusted persons.
Monitoring	Incorporate regular and systematic procedures to review records and logs for unusual transactions, accesses, and other patterns.
Asset control and reconciliation	Limit physical access to valuable assets, such as inventory, and periodically reconcile physical asset counts with related records.
Security	Design security features into individual systems and supporting infrastructure. Review and test security features frequently. Use outside consultants to conduct penetration testing attack and fraud vectors from external and internal sources.

Designing Security Controls

- Protect all assets against external threats
- Other objectives
 - Protect and maintain a stable, functioning operating environment 24/7 (equipment, operating systems, D B M S s)
 - Protect information and transactions during transmission across networks and Internet
- Access Controls – Limit a person's ability to access servers, files, data, applications
 - Authentication – to identify users
 - Multifactor Authentication
 - Access control list – list of valid users
 - Authorization – authenticated user's list of permission level for each resource
- Registered Users – those with authorization
- Unauthorized Users – anyone not registered
- Privileged Users – those that maintain lists and systems

Types of Users

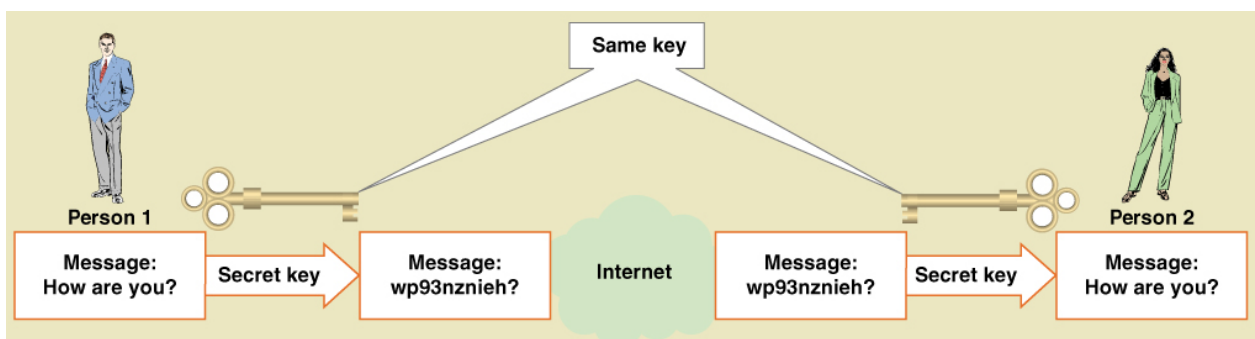


Data Encryption

- Method to secure data – stored or in transmission
- Encryption – alter data so it is unrecognizable
- Decryption – converted encrypted data back to readable format
- Encryption Algorithm – mathematical transformation of the data
- Encryption Key – a long data string that allows the same algorithm to produce unique encryptions

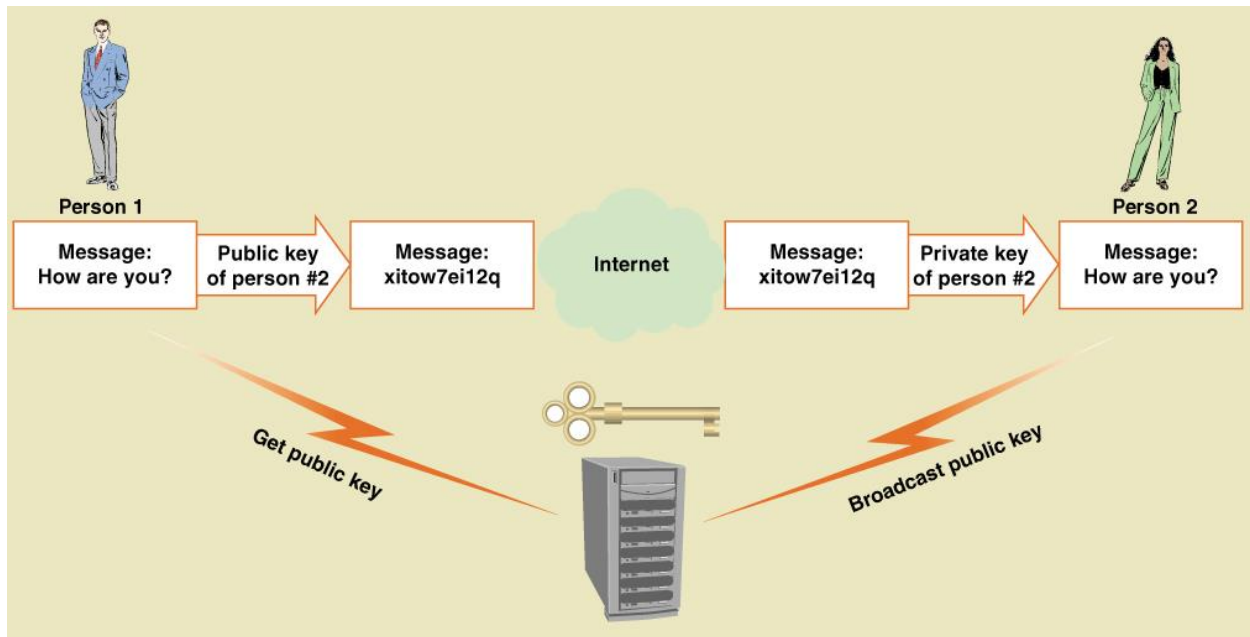
Symmetric Key Encryption

- Encryption method that uses the same key to encrypt and decrypt



Asymmetric Key Encryption

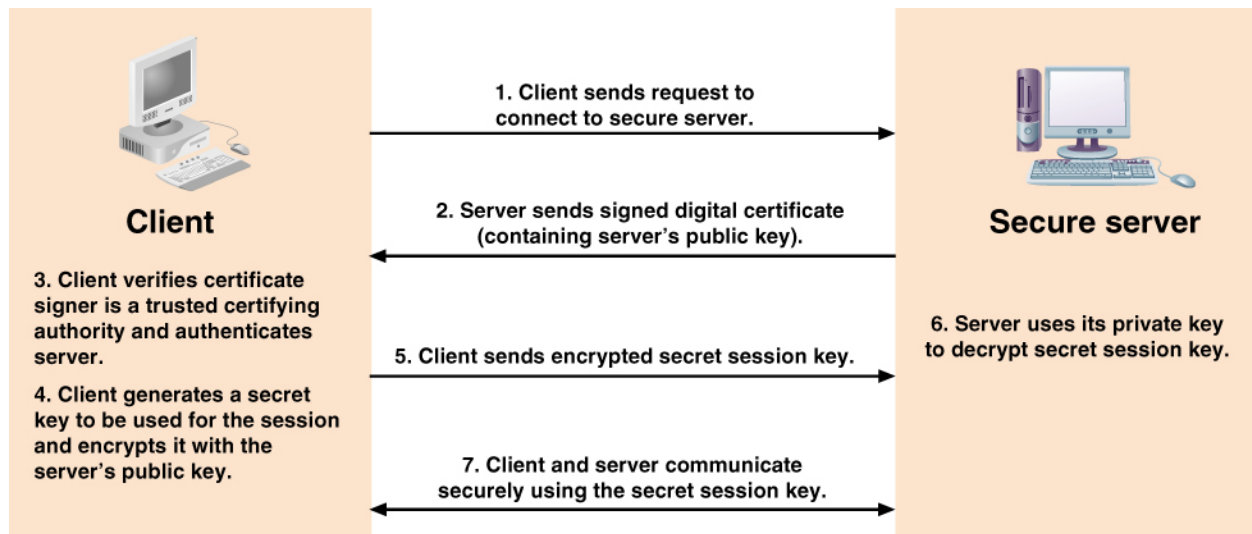
- Encryption method that uses different keys to encrypt and decrypt
 - AKA Public Key Encryption



Digital Signatures and Certificates

- Digital Signature – a technique where a document is encrypted using a private key
 - Note – implements the previous slide, but in reverse
 - Document is encrypted with the private key, but then can only be decrypted with the correct public key
- Digital Certificate – An organization name and public that is encrypted and certified by an authorized third-party
- Certifying Authority – the authorized third party
 - Widely known and accepted – built into Web browsers

How a Digital Certificate is Used



Secure Transactions

- Secure Sockets Layer (SSL) – standard set of protocols for authentication and authorization
- Transport Layer Security (TLS) – an Internet standard equivalent to SSL
- IP Security (IP Sec) – Internet security protocol at a low-level transmission
- Hypertext Transfer Protocol Secure (HTTPS) – Internet standard to transmit Web pages