Chapter 6 : Foundation for Systems Design

Tannaz R.Damavandi
Cal Poly Pomona

Credit for most of the slides :Systems Analysis and Design in a Changing World, 7th Edition ©2016. Cengage Learning. All rights reserved.
Image credit : Google image, dreamstime.com

Outline

- What Is Systems Design?
- Design Activities
- System Controls and Security

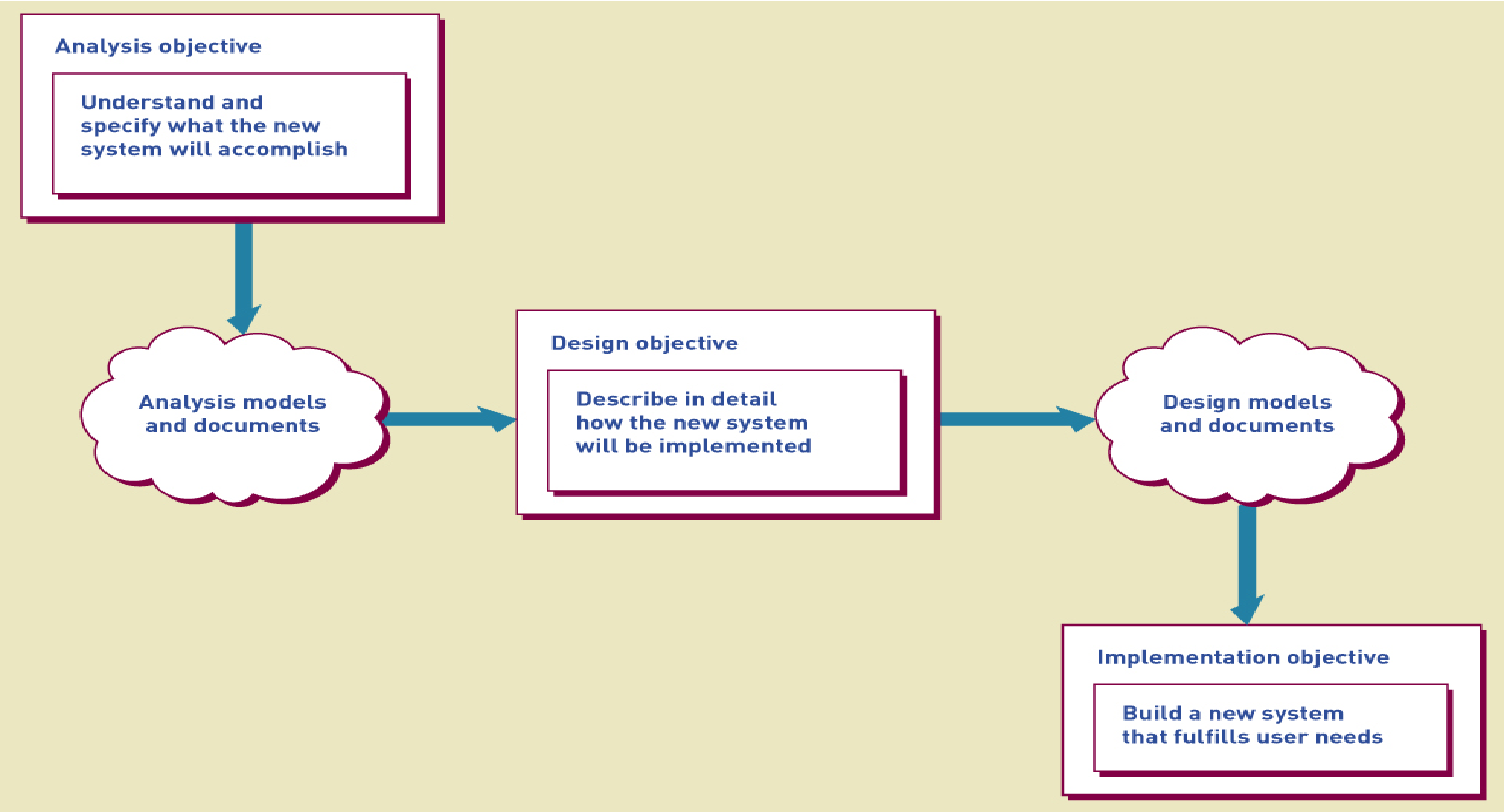
Overview

- Analysis says “what” is required and design tells us “how” the system will be configured and constructed
- Chapters 2, 3, 4 and 5 covered systems analysis activities (requirements)
- This chapter introduces system design and the design activities involved in systems development
- Design bridges the gap between requirements to actual implementation

What Is Systems Design

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

Analysis to Design to Implementation


Analysis objective

Understand and specify what the new system will accomplish

Analysis models and documents

Design objective

Describe in detail how the new system will be implemented

Design models and documents

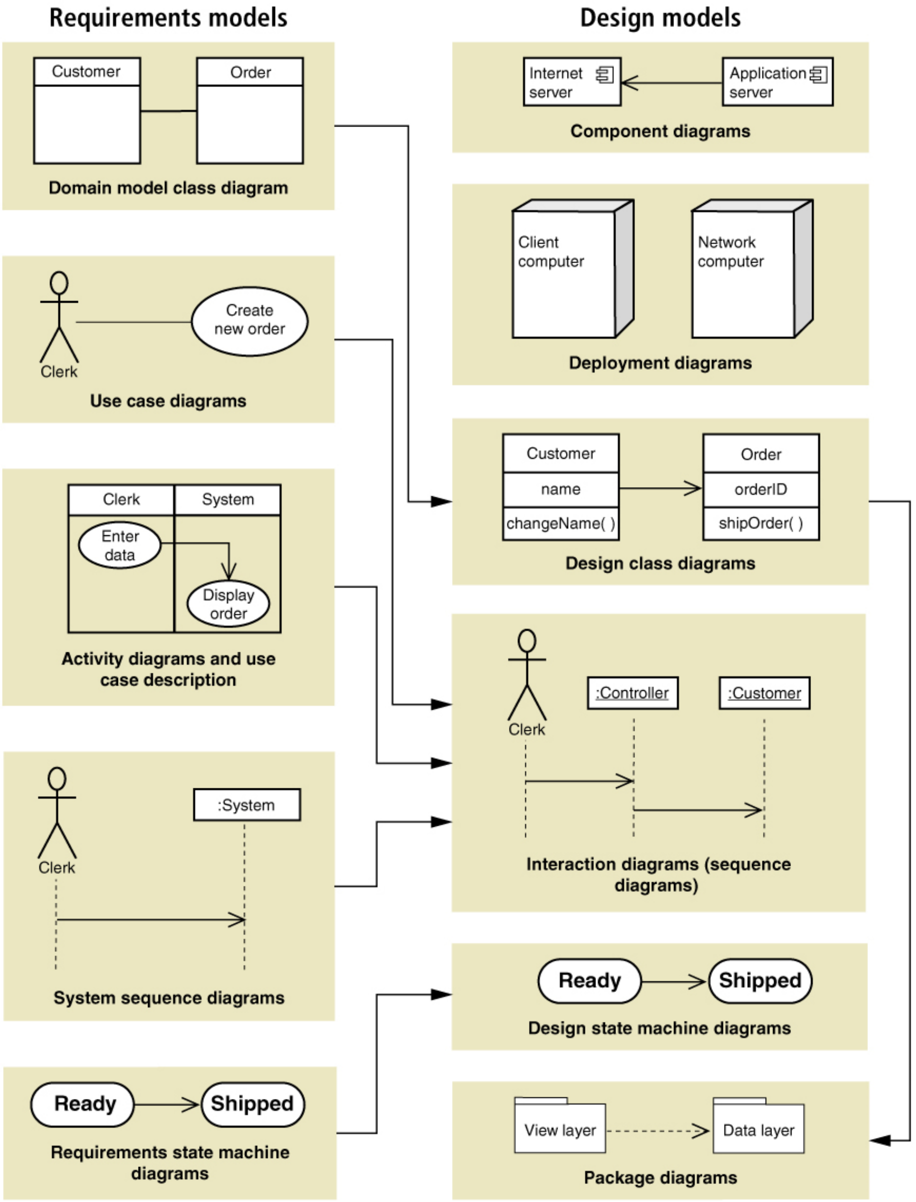
Implementation objective

Build a new system that fulfills user needs

Design Models

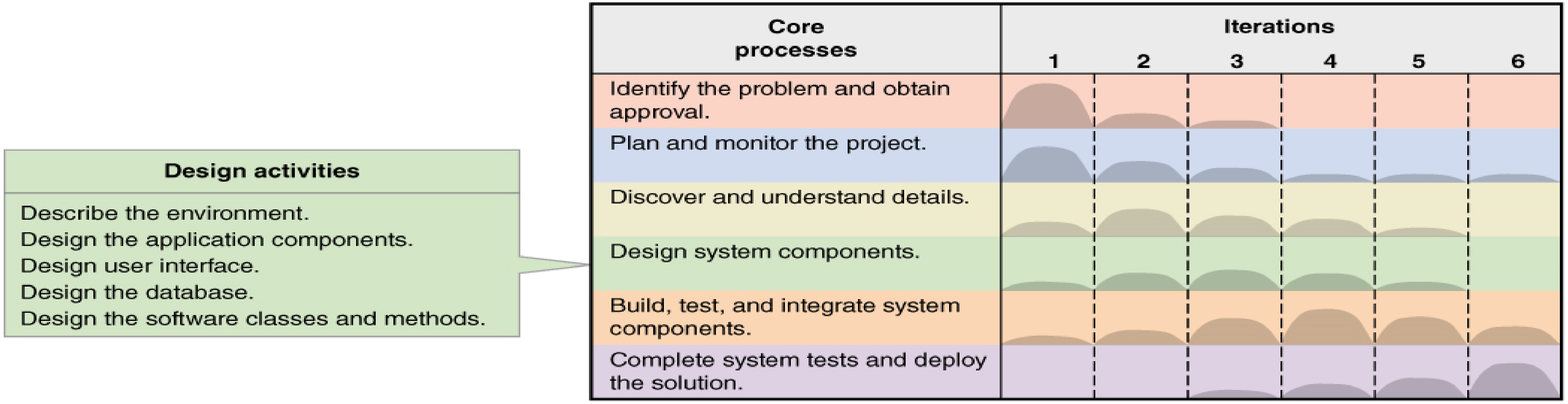
- Design is a model building activity
- The formality of the project will dictate the type, complexity, and depth of models
- *Agile/iteration projects typically build fewer models, but models are still created*
- ***Jumping to programming without design often causes less than optimum solutions and may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

Analysis Models to Design Models



Design Activities

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

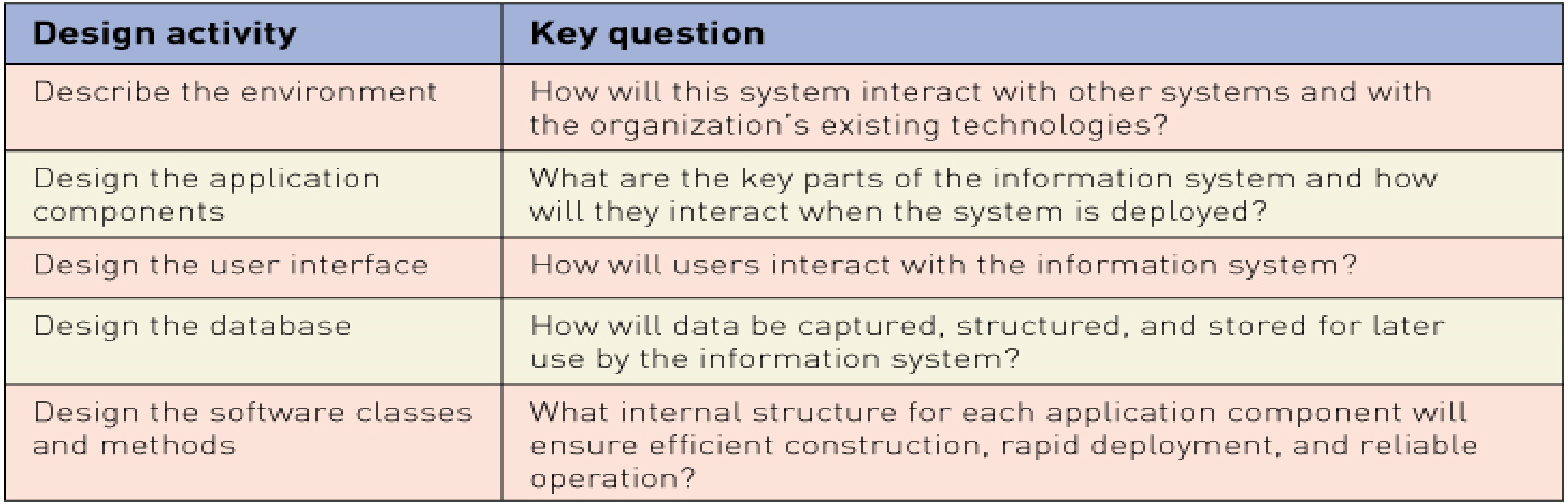


Design activities

Describe the environment.
Design the application components.
Design user interface.
Design the database.
Design the software classes and methods.

Core processes
Iterations
1
2
3
4
5
6
Identify the problem and obtain approval.
Plan and monitor the project.
Discover and understand details.
Design system components.
Build, test, and integrate system components.
Complete system tests and deploy the solution.

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?

Design the Environment

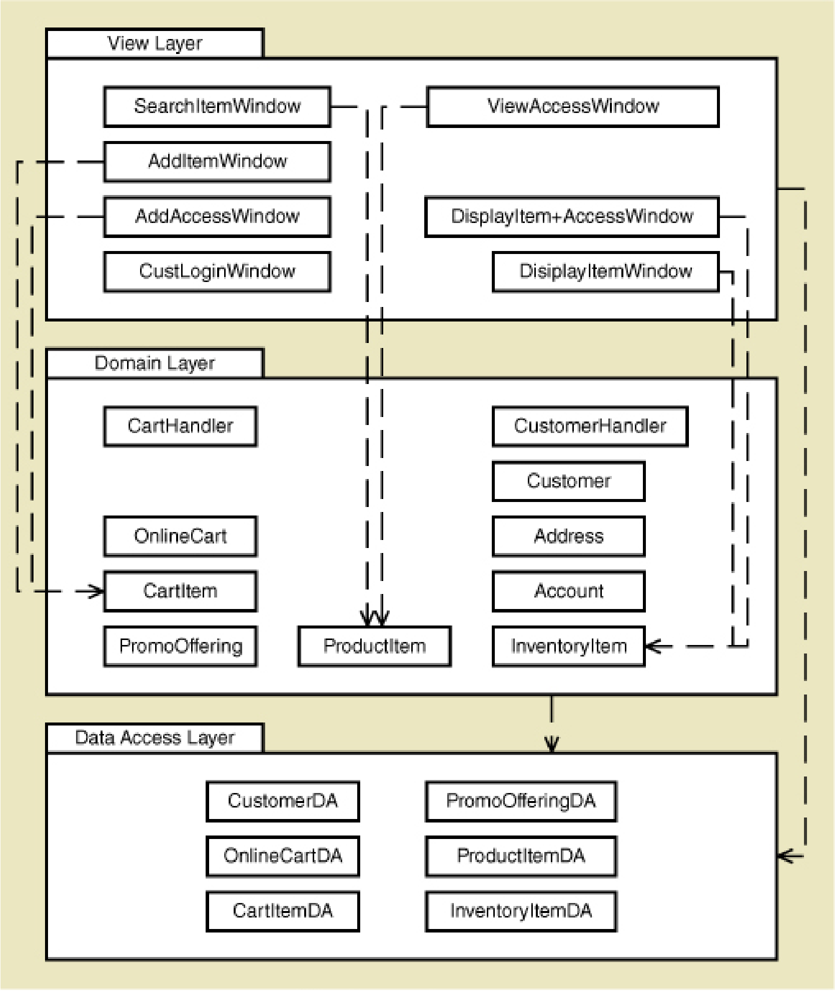
The environment is all of the technology required to support the software application

- Servers, Desktop computers , mobile devices ..
- Operating systems
- Communication capabilities, Input and output capabilities

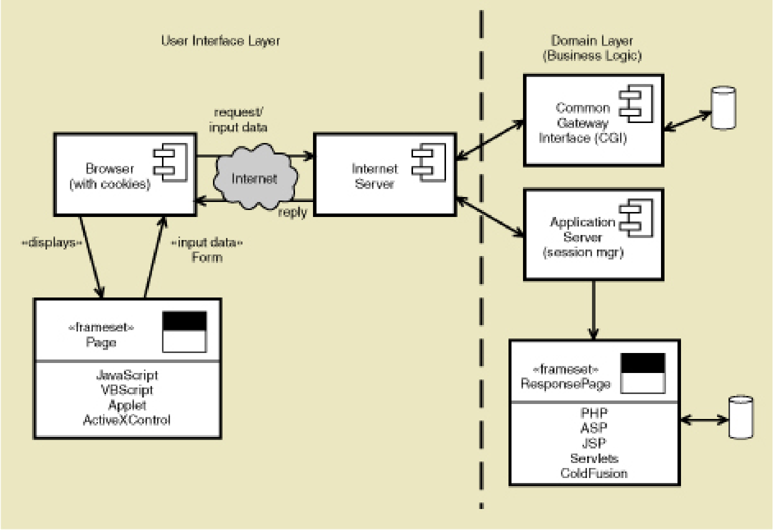
Design the Application Components

- ***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, interfaces?
  - Programming language – what are the accepted languages?
  - Build or buy – is an acceptable version available to purchase?

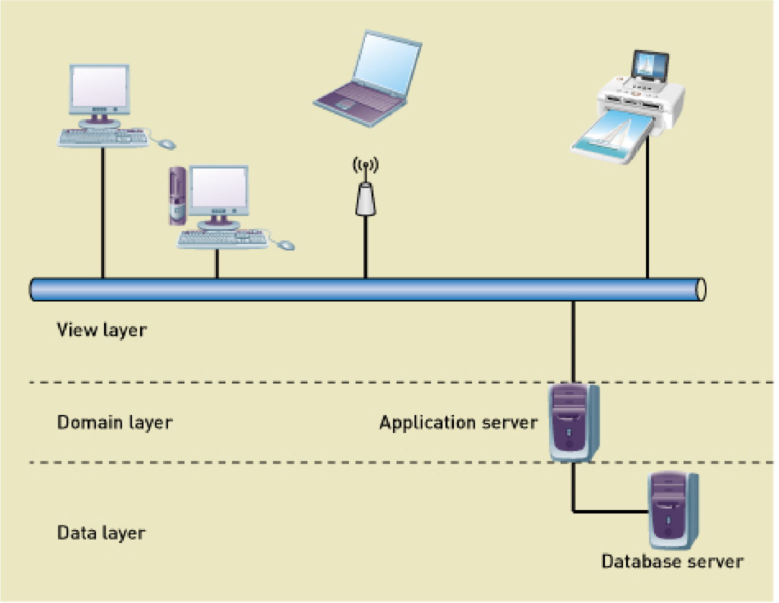
Typical Models for Designing Application Components



Package diagram



Component diagram

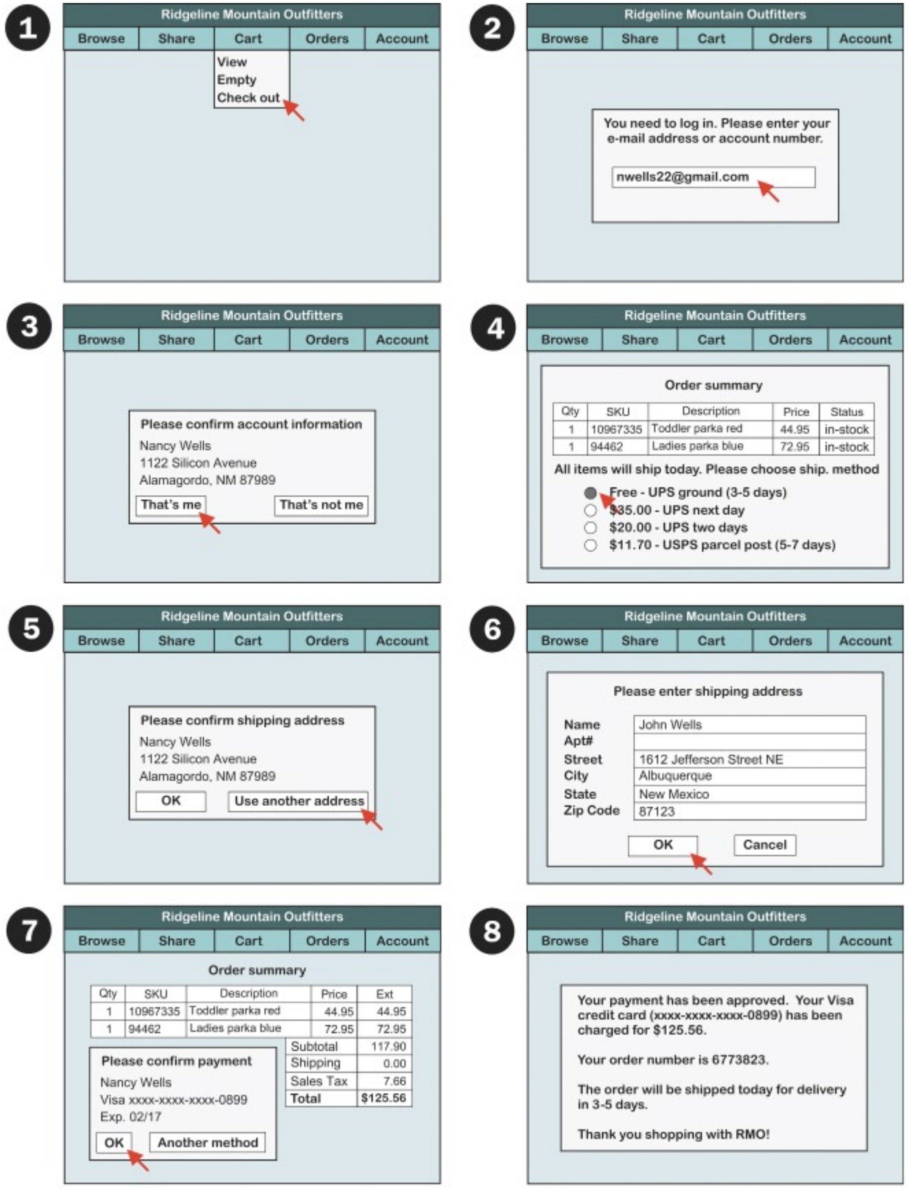


Deployment diagram

Design the User Interface

- To the user, the User Interface **is** the system.
- The user interface has large impact of user productivity
- Includes both Analysis and Design tasks
  - Requires heavy user involvement

Typical Models for User Interface Design



1. View
Empty
Check out

2. You need to log in. Please enter your e-mail address or account number.
nwells22@gmail.com

3. Please confirm account information
Nancy Wells
1122 Silicon Avenue
Alamagordo, NM 87989
That's me | That's not me

4. Order summary

Qty	SKU	Description	Price	Status
1	10967335	Toddler parka red	44.95	in-stock
1	94462	Ladies parka blue	72.95	in-stock

All items will ship today. Please choose ship. method
● Free - UPS ground (3-5 days)
○ $35.00 - UPS next day
○ $20.00 - UPS two days
○ $11.70 - USPS parcel post (5-7 days)

5. Please confirm shipping address
Nancy Wells
1122 Silicon Avenue
Alamagordo, NM 87989
OK | Use another address

6. Please enter shipping address

Name	John Wells
Apt#	
Street	1612 Jefferson Street NE
City	Albuquerque
State	New Mexico
Zip Code	87123

OK | Cancel

7. Order summary

Qty	SKU	Description	Price	Ext
1	10967335	Toddler parka red	44.95	44.95
1	94462	Ladies parka blue	72.95	72.95
			Subtotal	117.90
			Shipping	0.00
			Sales Tax	7.66
			Total	**$125.56**

Please confirm payment
Nancy Wells
Visa xxxx-xxxx-xxxx-0899
Exp. 02/17
OK | Another method

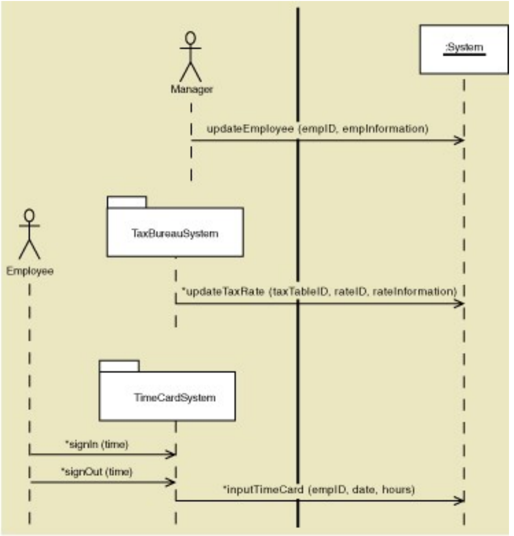
8. Your payment has been approved. Your Visa credit card (xxxx-xxxx-xxxx-0899) has been charged for $125.56.

Your order number is 6773823.

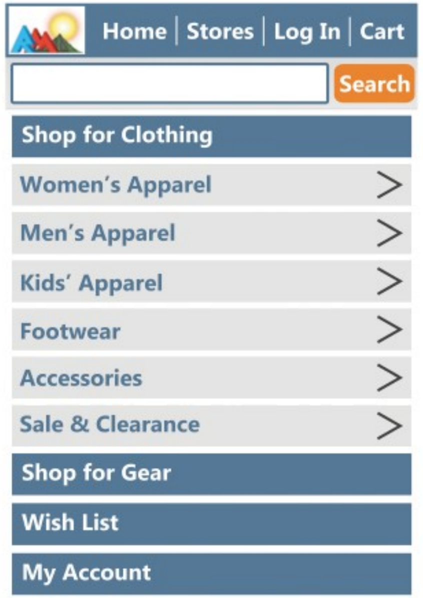
The order will be shipped today for delivery in 3-5 days.

Thank you shopping with RMO!

Storyboard


Manager
:System
updateEmployee (empID, empInformation)
TaxBureauSystem
Employee
*updateTaxRate (taxTableID, rateID, rateInformation)
TimeCardSystem
*signIn (time)
*signOut (time)
*inputTimeCard (empID, date, hours)

System sequence diagram

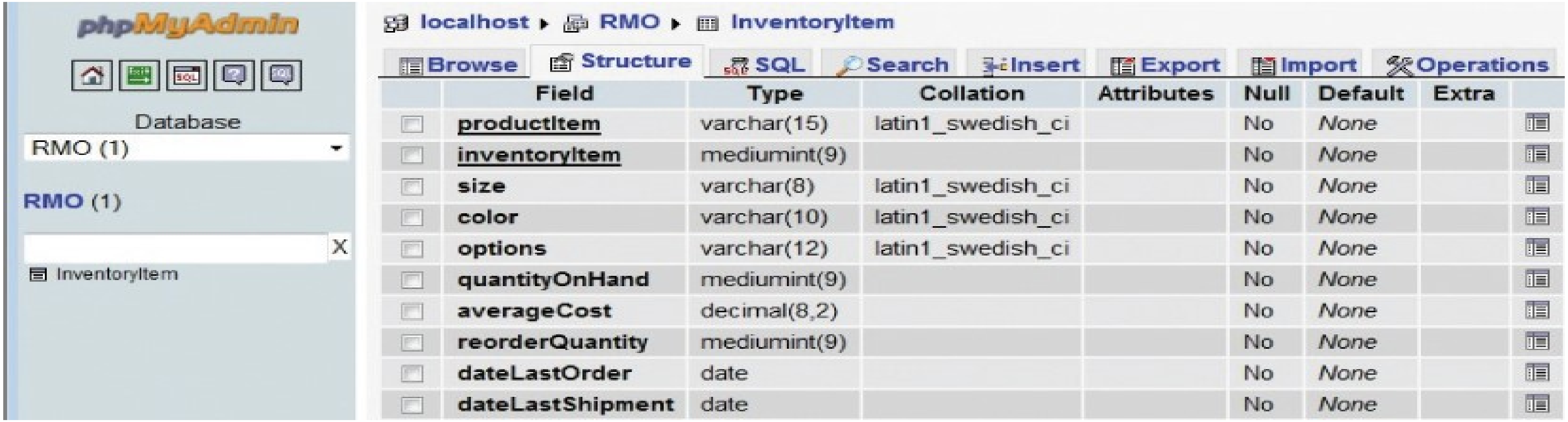


Small screen menu prototype

Design the Database

- ***By definition, an Information System requires data – usually in a database***
- Current technology frequently use Relational Database Management Systems (RDBMS)
- *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 Scheme

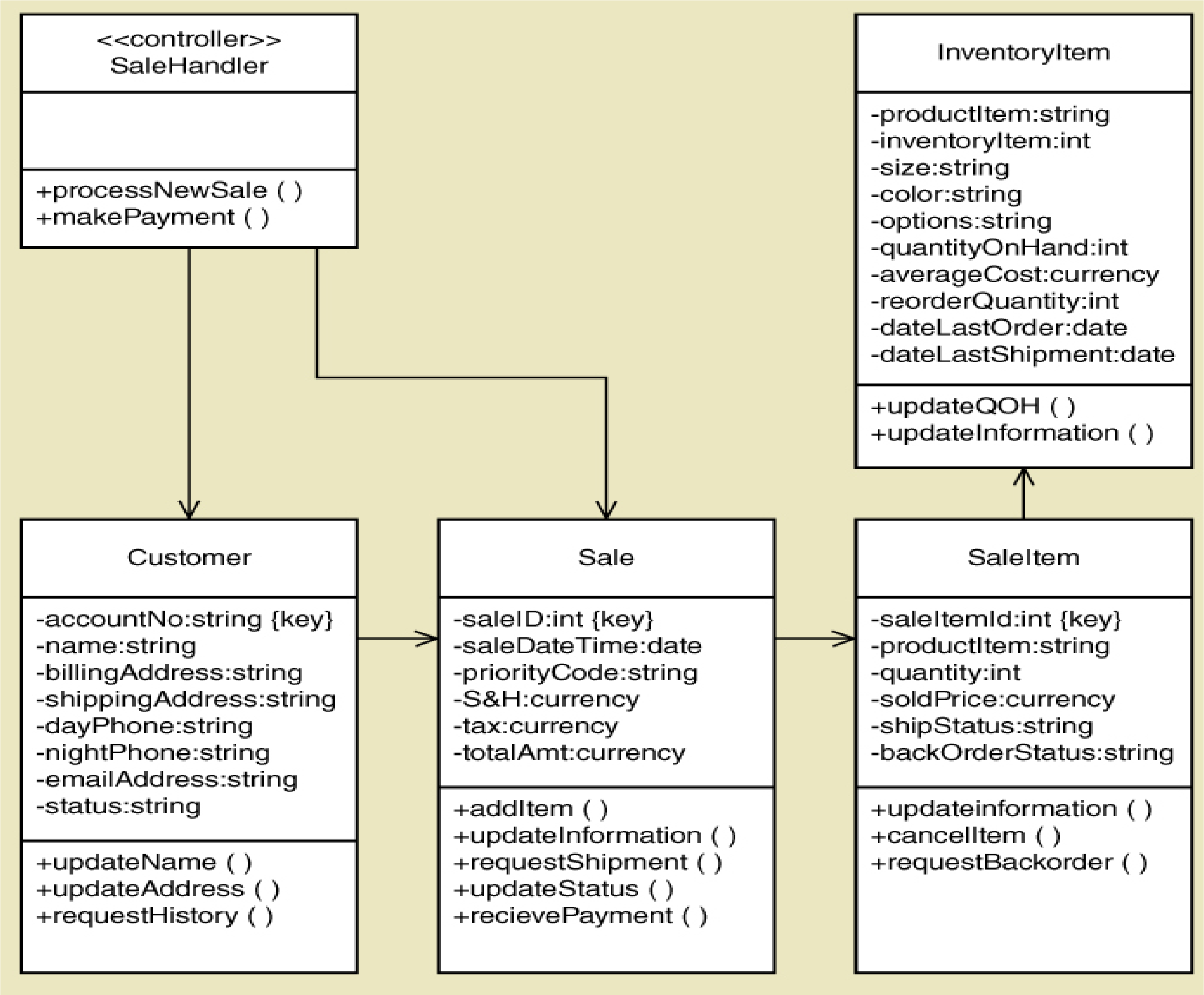


	Field	Type	Collation	Attributes	Null	Default	Extra	
□	productItem	varchar(15)	latin1_swedish_ci		No	*None*		
□	inventoryItem	mediumint(9)			No	*None*		
□	size	varchar(8)	latin1_swedish_ci		No	*None*		
□	color	varchar(10)	latin1_swedish_ci		No	*None*		
□	options	varchar(12)	latin1_swedish_ci		No	*None*		
□	quantityOnHand	mediumint(9)			No	*None*		
□	averageCost	decimal(8,2)			No	*None*		
□	reorderQuantity	mediumint(9)			No	*None*		
□	dateLastOrder	date			No	*None*		
□	dateLastShipment	date			No	*None*		

Design Software Classes and Methods

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

Typical Design Class Diagram with Attributes and Methods



<<controller>>
SaleHandler

+processNewSale ( )
+makePayment ( )

InventoryItem

-productItem:string
-inventoryItem:int
-size:string
-color:string
-options:string
-quantityOnHand:int
-averageCost:currency
-reorderQuantity:int
-dateLastOrder:date
-dateLastShipment:date

+updateQOH ( )
+updateInformation ( )

Customer

-accountNo:string {key}
-name:string
-billingAddress:string
-shippingAddress:string
-dayPhone:string
-nightPhone:string
-emailAddress:string
-status:string

+updateName ( )
+updateAddress ( )
+requestHistory ( )

Sale

-saleID:int {key}
-saleDateTime:date
-priorityCode:string
-S&H:currency
-tax:currency
-totalAmt:currency

+addItem ( )
+updateInformation ( )
+requestShipment ( )
+updateStatus ( )
+recievePayment ( )

SaleItem

-saleItemId:int {key}
-productItem:string
-quantity:int
-soldPrice:currency
-shipStatus:string
-backOrderStatus:string

+updateinformation ( )
+cancelItem ( )
+requestBackorder ( )

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

Designing Integrity Controls

- Integrated into application programs and DBMS
- 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

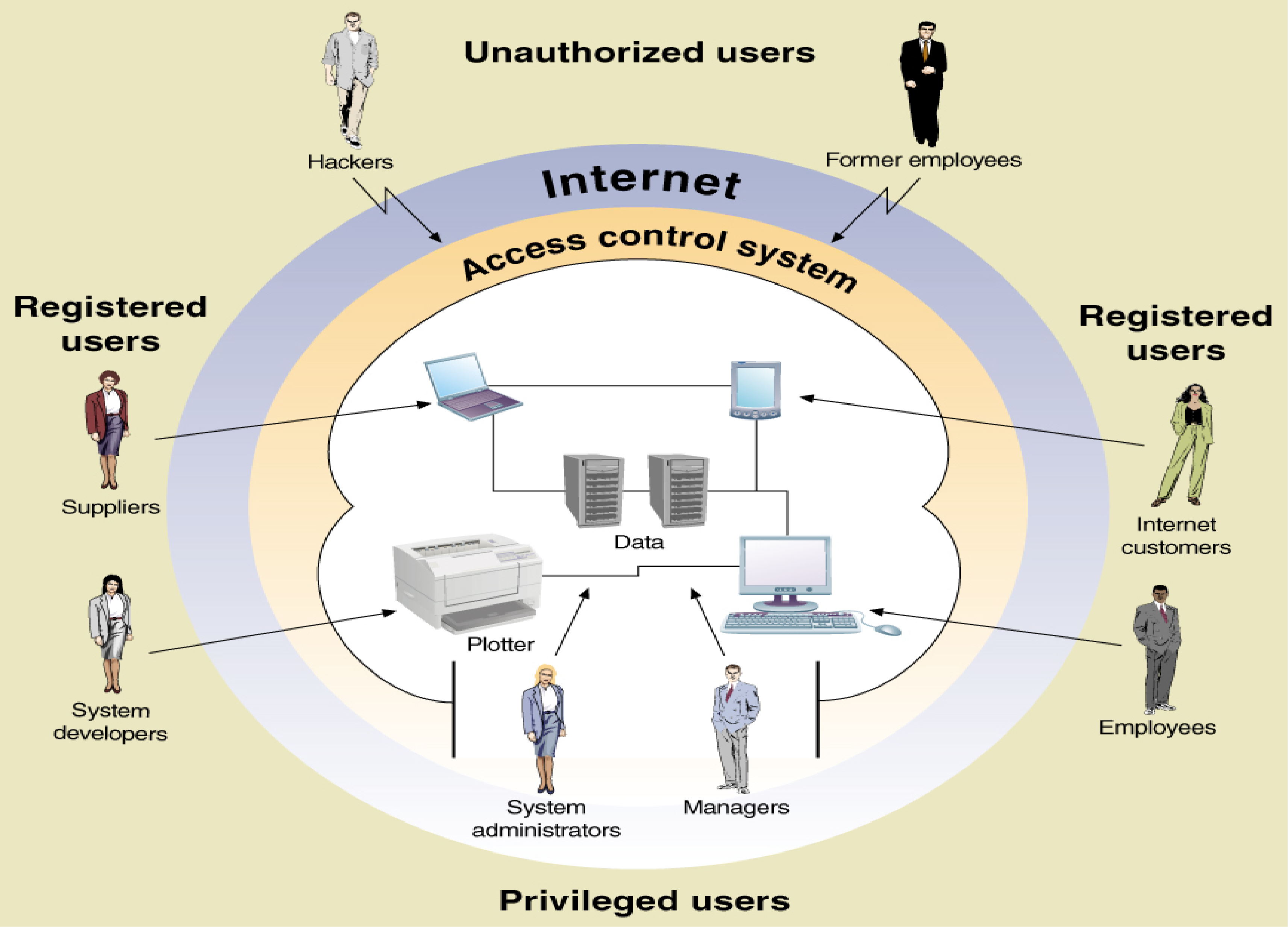
Designing Security Controls

- Protect all assets against external threats
- Other objectives
  - Protect and maintain a stable, functioning operating environment 24/7 (equipment, operating systems, DBMSs)
  - Protect information and transactions during transmission across networks and Internet

Designing Security Controls

- 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

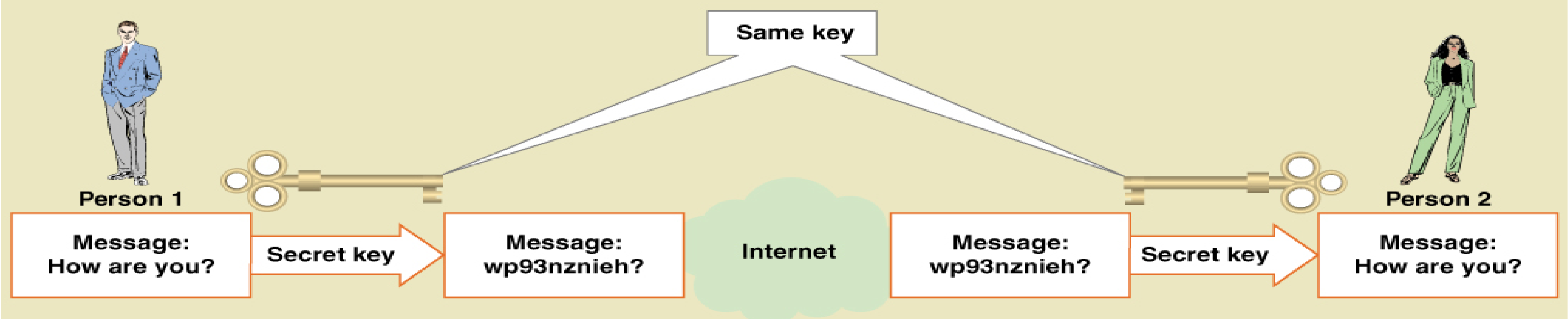


Data Encryption

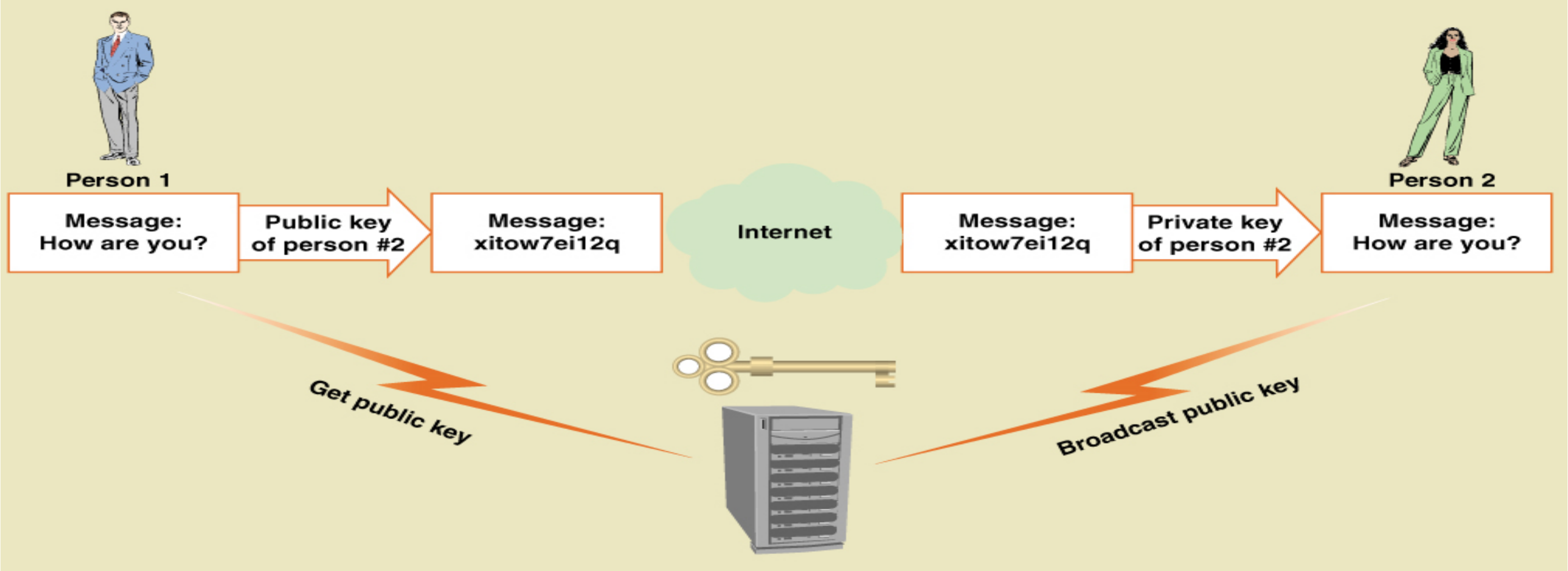
- Method to secure data – stored or in transmission
- Encryption – alter data so it is unrecognizable
- Decryption – converted encrypted data back to readable format
- Encryption Algorithm – mathematical transformation of the data
- Encryption Key – a long data string that allows the same algorithm to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

Symmetric Vs. Asymmetric Key Encryption

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



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



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 (IPSec) – Internet security protocol at a low-level transmission
- Hypertext Transfer Protocol Secure (HTTPS) – Internet standard to transmit Web pages

Summary

- This chapter introduced the concept of Systems Design
  - Analysis is fact finding and modeling
  - Design is modeling to specify how system will be implemented
  - Design is bridge between analysis an implementation
- Activities of Systems Design
  - Describe the environment
  - Design the application components
  - Design the User Interface
  - Design the database
  - Design the software classes and methods

Summary (Cont’d)

- System Controls and Security
  - Integrity Controls
    - Input controls
    - Output controls
    - Backup and recovery
    - Fraud prevention
  - Security Controls
    - Access controls
    - Data encryption
    - Digital signatures and certificates
    - Secure transactions