

Getting Started

Important: After you have completed your assignments or have written the final exam, do not discuss the contents on the forums (or post it anywhere else on the Internet). Posting discussion regarding the contents of final exams or completed assignments will be considered cheating and will be dealt with accordingly.

The 16-week schedule of this course is intended to provide guidance on the sequence and timing of your course activities. It is not expected that every student will adhere strictly to this schedule, and you may complete the course in less or more time than suggested (please consider your contract end date when planning your schedule!).

Before you start:

- Read the *Welcome* letter in the course materials package.
- If you encounter persistent problems with the course materials access, first check the COMP 361 General Discussion Forum. If all else fails, contact the [Virtual Helpdesk](#).
- Continue to read the other sections of this unit. It is very important that you read this information!
- Familiarize yourself with the course materials, the course study plan, and the requirements for completing COMP 361.
- Email your tutor to let him or her know that you have all your materials in place and that you have read the requirements for completing the course. Report any problems, and ask for clarification on any aspect of the course materials or instructions that you do not understand.

Using Moodle

All SCIS undergraduate courses are now delivered through Moodle. You can log in to Moodle through [myAU](#). You may want to bookmark your course site.

Assignments are submitted through the upload feature at the bottom of each assignment page. The first step is to attach multiple files, as well as delete or overwrite uploaded files. Before you proceed to the next step, which is to submit the assignment, make sure you have completed uploading all files and all edits are done. Once you hit the **Send for marking** button and confirm your submission, your tutor is alerted that you have submitted your assignment and you will no longer be able to add, delete, or edit your assignment files (unless your tutor reverts your submission to Draft status).

You have access to your active courses in Moodle until the course end date. After this, you will continue to have access for 4 more months, but you will not have access to the tutor and will not be able to post in the discussion forums.

This interface should be fairly easy to use, but if you have any questions or concerns, please contact Computing Services Help desk at <https://secure.athabasca.ca/inter/helpdesk/contacthd.htm>. Improvements and updates to Moodle are being done on a continuous basis.

Course Overview

The Subject Matter: Systems Analysis and Design

The analysis, design, and implementation of information systems (information systems development) is the application of problem solving to a particular area of human activity. As such, there is nothing inherently difficult about the subject. There is, however, substantial complexity because of the broad range of concerns and activities under the systems development umbrella.

This version of COMP 361 breaks new ground: it facilitates learning in a community and through discussions with your peers, focuses on the object-oriented approach, is use-case driven, and is compliant with UML (Unified Modeling Language) 2.0 modelling standards.

Systems analysis and design as a subject deals with the concepts, skills, methodologies, techniques, tools, and perspectives essential for systems analysts. This course takes an integrated approach to the subject:

- Discussion of concepts, skills, methodologies, techniques, tools and perspectives in the textbook is supplemented by case studies such as the Rocky Mountain House Outfitters and Reliable Pharmaceutical Service to provide experience with problem-solving techniques and issues addressed in the textbook.
- The assignments in the course reinforce the textbook materials by applying systems analysis and design concepts, skills, methodologies, techniques, tools, and perspectives to specific issues and subjects.
- The assignments support each other: Assignments 1–5 successively refine the requirements and design for an information system for the Car Sharing case study.

COMP 361 and the Web

COMP 361 uses web-based resources extensively because of the currency of the materials posted on a multitude of academic and commercial websites. Access is organized as follows:

- Supplementary readings. Websites with content relevant to specific course topics are provided to demonstrate various viewpoints on the respective topics.
- Worked examples. Additional examples of problem-solving or modelling exercises are identified and made available to you.
- Directed searches. Topic-based search strings are supplied to direct your attention to a range of web-based resources on specific topics.

How to Proceed and How Not to Proceed

Because of the integrated nature of the treatment of the subject, you are advised to follow as closely as possible the suggested sequence. Furthermore, you are strongly encouraged to communicate and discuss topics with other students through blogs and discussion forums!

You are strongly discouraged from taking the assignment-oriented approach to the course, that is, seeing the course as a collection of assignments with supporting materials. This will almost inevitably lead to two outcomes: first, an incomplete understanding of the course materials because the structure implicit in the materials is ignored; and second, difficulty completing the assignments because of incomplete and/or incorrect understanding of the applicable concepts, skills, methodologies, techniques, tools, and perspectives.

The suggested sequence assumes part-time study: that you will spend 10–15 hours per week on the course and will take approximately 16 weeks to complete it. Students who intend to study full time, or whose personal schedules are likely to be problematic or significantly different from this norm, are encouraged to develop their own schedules. You should review your schedule on a regular basis in case reorganization is required. [Extensions are available](#) should it become apparent that six months will not be sufficient to complete the course.

Course Materials

Study Guide

The seven units of the study guide provide you with information about how and what to learn in COMP 361. Each unit consists of a sequence of **sections** based on the textbook structure. It is therefore strongly suggested that you follow the sequence. The basic components of the sections are **learning objectives**. Learning objectives are used to organize the learning

resources from the textbook. A learning objective provides a focus for the content and activities: it is a statement of what you should be learning in each section. Each objective usually includes reading materials from the textbook, activities to check your understanding, and supplementary reading materials.

Each unit and each section include a brief **introduction**, which provides an overview of the unit/section. Please note that these introductions are strongly based on text from the textbook but also include important information specifically related to COMP 361.

Each unit further includes **unit activities**, which are based on the exercises and case studies provided in the textbook. The study guide points out the exercises and case study questions you are responsible to complete. Completing these review components at the end of each unit should be part of your approach to mastering the learning objectives. Solutions for the problems and exercises activities and the case studies are available through each Summary and Unit Activities section.

NOTE: Answers for Review Questions are not provided because they simply repeat the textbook material. Use the Review Questions to organize your reading of the textbook material.

A Word about Learning Objectives

Learning objectives are used to define the preferred outcomes of studying these course materials. If you understand the language of these learning “requirements,” you will be able to develop the right strategies for studying and completing assignments. The language used to express learning objectives is defined and explained in what is commonly referred to as *Bloom’s Taxonomy*.

Benjamin Bloom created this taxonomy for categorizing the types of questions that commonly occur in educational settings. The taxonomy is useful for categorizing the type of learning expected in a course—whether cognitive (mental skills, knowledge), affective (growth in feelings or emotional areas, attitude), or psychomotor (manual or physical skills).

You can find further information at Bloom’s Taxonomy of [Learning Domains](#).

Textbook

Analysis and Design in a Changing World, 6th edition, by J.W. Satzinger, R.B. Jackson, and S.D. Burd.

In order to get familiar with the textbook and its features, please read “Features” on pages x–xv and “Preface” on pages xvi–xxv.

Note: This is a digital textbook (eTextbook). Access and download it through the link on the course home page. Because this course has transitioned from a print textbook to an eText, you may notice minor discrepancies between the textbook page numbers referred to in the course and the page numbers in the eText.

Case Studies in the Textbook

The textbook provides running case studies at the end of each chapter. In the unit activities at the end of each unit of COMP 361, you will be asked to read through some of these case studies and do some of the exercises provided in the textbook. Solutions for the exercises are provided.

These case studies are a very good way of learning how to apply the concepts and theories you will learn in this course. Please do not underestimate the importance of carefully following the case studies. They provide clear examples of the kinds of analysis and modeling activities you are asked to complete for the assignments based on the Car Sharing case study.

Online Companion Website

The [Online Companion website](#) provides you with some materials to supplement the textbook such as crossword puzzles, flashcards, and a glossary.

To access the student companion website, check the envelope on the inside front cover of the textbook. In the envelope, you will find instructions as well as a PIN that will provide you with access privileges. After you have used your PIN, you can also access the Online Companion via the link in the **Important Resources** block on the right of the course home page.

Blogging

COMP 361 facilitates learning in a community and through discussions with your peers. Through reflecting about your learning and discussing about the learning material, you can learn much more than by only reading through the material.

Therefore, you are expected to create a blog on the Landing in the COMP 361 group. Your blog should reflect on the activities in the assignments and discuss what worked well and what was difficult; it is a space for sharing problems, thoughts, and contributions. Thus, the last part of each assignment is to write a posting in your blog (see assignment descriptions for further details).

How to get started?

Follow these steps to start your blog:

1. Go to <https://landing.athabascau.ca/> and log in using your AU username and password.
2. Read through the welcome page, and accept the conditions.
3. At the top, you will find “My Profile” where you can add details to your profile.

4. Next, go to [https://landing.athabasca.ca/pg/groups/71651/COMP 361-systems-analysis-and-design/](https://landing.athabasca.ca/pg/groups/71651/COMP%20361-systems-analysis-and-design/). On the left side, you will find a “Request membership” link. Click on this link to be added to the group “COMP361 – Systems Analysis and Design.”
5. The COMP361: Systems Analysis and Design group will be your access point for your peers’ blogs. Once someone has posted in his/her blog, read through the blog, and write comments in response.
6. To start your own blog, first go to your own dashboard (You/Your dashboard on the top navigation bar); then click on “Create & Share.” Click on “Write a blog posting.” Please use the following settings:
 - a. Title: Include “COMP 361” in your title. An example of a title might be: “COMP 361 – Reflection about Assignment 1.”
 - b. Excerpt: A short descriptive/introductory sentence from your posting.
 - c. Upload icon: This is optional.
 - d. Body: This is where your posting goes.
 - e. Tags: **Please write “COMP361” in this field!** (This is very important, since otherwise your posting will not be listed in the COMP361 group)
 - f. Access: In this field, you can select who will be able to read your posting. Use “Group: COMP361 – Systems Analysis and Design.” If you want your postings to be read by other users of the Landing as well, you can use “Logged in users.” The “Public” setting will give everyone on the Internet access to your blog. If you don’t want anyone to read your blog, you can use the setting “Private” (however, you will still have to submit a copy of your blog postings to your tutor for the participation grade).
 - g. Comments: Choose “On.” If you don’t want your peers to comment on your blog posting, you can choose “Off.”
 - h. To publish your posting, choose “Published” as the status.
7. Your first post should be a short message where you introduce yourself to your peers and tutor by writing a few words about yourself.

What shall I write in my blog?

Use your blog to reflect on your problems, insights, and contributions during and after your task-related activities for the course. For example, you could ask yourself the following questions: What causes/caused me problems (and why)? Which solutions have I found/tried that could also be useful for my colleagues? Where was I unable to find a solution (and why)? How do/did I approach the current problem? etc.

The minimum requirement is one blog posting per person per assignment. Each of these blog postings should be at least 300 words long. However, you are encouraged to use your blog more often. Furthermore, please look into what your peers write, and comment on their blogs as well.

What shall I NOT write in my blog?

You are not allowed to include your solution for any assignment in your blog posting! Furthermore, you are not allowed to discuss the questions and solutions of the exam in your blog posting!

Does blogging count towards my final marks?

Yes, blogging counts for your final marks in two ways. First, when you write a blog posting in your own blog to reflect about an assignment, this will count towards your assignment mark. Second, when you comment on a blog posting of one of your peers, this will count towards your participation mark.

In both cases, you need to follow the requirements of the assignment/participation marks (e.g., length of posting, etc.) in order to make your posting count toward your marks.

Discussion Forums

As mentioned before, COMP 361 facilitates learning in a community and through discussions with your peers. Therefore, you will find several discussion forums in the course. Following is a brief description of the forums and their purpose:

COMP 361 General Discussion Forum

This forum is intended for general discussions on the topics of COMP 361 or any issues you face during the course.

Discussion about DRR Resources

Use this forum to discuss resources in the Digital Reading Room (DRR). In this forum, you can share your thoughts, opinions, and ideas about these resources. Posting in this forum will contribute to your participation marks.

Other Discussion Forums within the Units

Each unit includes one discussion forum. These discussion forums hide the answers of your peers until you have posted your answer. Posting in this forum will contribute to your participation marks.

Participation Marks

NOTE: You need to achieve at least 50% on the participation marks to pass COMP 361.

In order to highlight the importance of participation in discussions with your peers as an element of this course that is vital to your successful learning, participation marks are given. **To get full marks, you are expected to write at least 10 postings.** These postings can be

- an answer to the questions asked in the discussion forums within the units.
- a comment on a peer's answer in a discussion forum within the units.
- a posting to the Discussion about DRR Resources forum.
- a comment to a peer's posting in the Discussion about DRR Resources forum.
- a comment to a blog posting from one of your peers.

In order to count for participation marks, the posting has to fulfill the following requirements:

- The posting has to be at least 300 words long.
- For answers to the questions asked in the discussion forums within the units, the posting has to actually answer the questions asked.
- For postings to the Discussion about DRR Resources forum, the posting has to reflect about the respective resource and describe positive and/or negative points of the resource. Contributions such as "That's a great resource!" or "This resource is not so helpful" are very welcome, but would not count towards your participation marks.
- For comments to postings of your peers in an discussion forum within the units, the Discussion about DRR Resources forum, or a blog entry of your peers, your comment has to be related to their postings and provide an argument or your viewpoint based on literature from the Internet or the AU library. Comments like "I fully agree" or "I don't agree" or "Same for me" are very welcome, but would not count toward your participation marks.

At the end of the course, you will submit a portfolio of your postings/comments to your tutor. This portfolio will include a copy of all your relevant postings and comments to discussion forums and your relevant comments on your peers' blogs.

About the Assignments

You will complete five assignments in COMP 361. The weights of the assignments are as follows:

Assignment	Description	Weight
Assignment 1	Business Cases and Events	15%
Assignment 2	Domain Classes and Use Cases	20%
Assignment 3	User and System Interfaces	8%
Assignment 4	Sequence Diagrams and Design Class Diagrams	15%
Assignment 5	Databases, Controls, and Security	12%

The remainder of the marks for the course is 5% for participation and 25% for the final exam.

Each assignment consists of at least **three** parts: 1) a practical part where you are asked to apply analysis, modeling, and design techniques; 2) an essay part where you are asked to answer an essay question; 3) a blogging part where you are asked to write a blog posting.

The practical parts of the five assignments required to complete this course are all related to a single overarching learning objective: When you have completed the course you should have a theoretical and practical understanding of the analysis and design techniques and tools used in a systems development project.

Tutors normally will mark assignments in about 7-8 business days. Because the assignments are related, you should leave sufficient time between assignment submissions to allow for tutor feedback. Since the assignments are based on each other, your tutor may suggest simple changes or may decide that a complete rewrite is required before you proceed to the next assignment.

All assignments involve the Car Sharing case study. You can find information about the Car Sharing case study in the Important Resources block on the right side of the course home page. Please download the file and read through it before starting your assignments!

Assignments will be assessed on content and presentation. Content will be judged on the correctness and completeness of the information presented. Presentation will be judged on the organization, format, readability, and the effectiveness in communicating the requested information.

NOTE: You need to achieve a mark of at least 50% on **each** assignment to pass COMP 361.

Modelling Tools

During COMP 361, you will create a number of different UML diagrams and models. Such diagrams and models can be created through the use of particular software. Please take a look at the tools recommended below, and use one of them for creating diagrams and models for your assignments.

Visual Paradigm

(<http://www.visual-paradigm.com/>)

Download:

Download Visual Paradigm at <http://www.visual-paradigm.com/solution/freeumltool/> and install the newest version of Virtual Paradigm for UML (selecting “Community Edition”).

Tutorials:

Check the resources page (<http://www.visual-paradigm.com/product/vpuml/resources.jsp>) and tutorial page (<http://www.visual-paradigm.com/product/vpuml/tutorials.jsp>) for information on how to use Visual Paradigm.

ArgoUML

(<http://argouml.tigris.org/>)

Download:

Download ArgoUML at <http://argouml.tigris.org/>. Check the Quick Guide for further information on the installation process.

Tutorials:

Check the Documentation page (<http://argouml.tigris.org/documentation/>) for further information. You can find a user manual, tour of ArgoUML, FAQ, etc., there.

Keep in mind that ArgoUML is an open source UML modelling tool, with an active developer community, where some of the documentation is still under development.

Modelio

(<http://archive.modeliosoft.com/>)

Download:

Download the free edition at: <http://archive.modeliosoft.com/en/products/modelio-free-edition.html>

Note:

- After you register, you will need to go back to the link above to download the software.
- Before starting with the installation process, please make sure that you run the setup as administrator.

Tutorials:

There are video tutorials available at <http://archive.modeliosoft.com/en/quick-tour/modelio-video-tutorials.html>. You can take a look at other tutorials at <http://archive.modeliosoft.com/en/tutorials.html>

Microsoft Visio (commercial tool)

If you have MS Visio available, you can also use it for creating UML diagrams and models. You should find a template for UML diagrams in the Software & Database category. Alternatively, you can check and install the stencils at <http://softwarestencils.com/uml/index.html>

If you have any questions about installing and/or using the above-mentioned tools, please let your tutor know!

If you would like to use a particular tool that is not in the list above, you have to ask your tutor for permission **before** you start working on your assignments!

Digital Reading Room

The Digital Reading Room (DRR) provides you with suggested readings. These offer additional information, different perspectives on the same material, or interesting points of view on the topics you are learning. Within each unit of the study guide, you will find links to DRR materials. You can also access the DRR directly in the **Important Resources** block on the right of the course home page.

Note: You will not be assessed, either on assignments or exams, for the contents of these readings.

Journal

Whenever you complete an activity, read web resources, etc., it is suggested that you make notes. These notes can be posted in your blog, or you can write them in a (private) journal. In any case, we strongly suggest that you make study notes in whatever format you prefer and either share them with your peers (e.g., through your blog) or keep them for yourself (e.g., in a text file, MS Word file, or an HTML file). You will find these notes valuable when you study for the final exam.

Unit 1: An Introduction to Systems Development

Information systems are strategic assets to modern businesses and organizations. Existing systems are constantly re-examined and redeveloped, and new applications are implemented to achieve or maintain competitive advantage. The key to successful system development is using *systems analysis and design* to understand what the business requires from the information system.

- **Systems analysis** means understanding and specifying in detail what the information system should accomplish.
- **Systems design** means specifying in detail how the many components of the information system should be physically implemented.

This unit provides an overview of systems analysis, systems design, and systems development.

Section 1: From Beginning to End – An Overview of Systems Analysis and Design

Systems analysis and design is, first and foremost, a practical field grounded in time-tested and rapidly evolving knowledge and techniques. This section discusses how systems development and systems design and analysis are related; provides an overview of the *systems development life cycle* (SDLC) and *iterative system development*, which are both core techniques in systems analysis and design; introduces the Ridgeline Mountain Outfitter case study, which will be used throughout the course as one of the examples; and discusses the further content of the course.

Learning Objectives

When you have completed this section, you should be able to do the following:



1. Describe the purpose of systems analysis and design in the development of information systems.
2. Explain the six core processes of the SDLC.
3. Describe the characteristics of iterative system development.
4. Explain key documents and diagrams in systems analysis and design as well as why use cases and object classes need to be identified.

Learning Objective 1

Describe the purpose of systems analysis and design in the development of information systems.

Objective Readings/Activities



Readings: Please read the following section in Chapter 1 of the textbook:

- Software Development and Systems Analysis and Design (pages 4–5)



Review: Please check your learning with Review Questions 1–3 on page 28 of the textbook.



Supplementary readings: Please check the Digital Reading Room for additional reading materials on the topic of this section.

Learning Objective 2

Explain the six core processes of the SDLC.

Objective Readings/Activities



Readings: Please read the following section in Chapter 1 of the textbook:

- Systems Development Life Cycle (pages 5–6)



Review: Please check your understanding with Review Questions 4–6 on page 28.

Learning Objective 3

Describe the characteristics of iterative system development.

Objective Readings/Activities



Readings: Please read the following section in Chapter 1 of the textbook:

- Iterative Development (pages 8–9)



Review: Please check your understanding with Review Questions 20 and 21 on page 29 of the textbook.

Learning Objective 4

Explain key documents and diagrams in systems analysis and design as well as why use cases and object classes need to be identified.

Objective Readings/Activities



Readings: Please read the following subsections in Chapter 1 of the textbook:

- Introduction to Ridgeline Mountain Outfitters (pages 6–8)
- Developing RMO's Tradeshaw System (pages 9–25)
- Where You Are Headed – The Rest of This Book (pages 26–28)



Review: Please check your understanding with Review Questions 7–19 on pages 28–29 of the textbook.

Unit 1– Summary and Unit Activities



Readings: Please read the Chapter 1 Summary on page 28 of the textbook.



Review: Review the Key Terms on page 28, and check if you know what they mean.

Unit 2: Approaches to Systems Development and Project Management

In this unit we look at approaches to systems development and project management. Although the textbook presents these topics at a later stage (Chapter 8 and Chapter 9), we are discussing these concepts now, since they are the basis for system analysis and design activities, and will help you better understand the next units.

This unit reviews the main approaches to system development that are currently used to develop business systems. This includes more details about the systems development life cycle (SDLC) as well as a discussion about the support phase. In addition, terms such as *methodologies*, *models*, *tools*, and *techniques* are introduced and discussed in the context of software development, and two approaches to system development are explained in more detail: the *structured approach* and the *object-oriented approach*. Further, *Agile development* is briefly outlined. This unit also deals with the main approaches to project management, discussing the principles of project management as well as the core activities.

Readings/Activities



Please read the following sections in Chapter 8 of the textbook:

- Development Approach at Ajax Corporation Consolidated Concepts, and Pinnacle Manufacturing (page 226)
- Overview (pages 226–227)

Section 1: Approaches to System Development

The entire process of developing an information system is more than just an SDLC. A system development methodology includes specific instructions for completing the activities of each core process by using specific *models*, *tools*, and *techniques*.

This section also reviews two main approaches to defining the information system technology and software development used for business systems: the *traditional approach* and the *object-oriented approach*.

- The traditional approach refers to structured software development, which describes software as a hierarchy of programs and modules, and uses structured analysis, structured design, and structured programming.
- The object-oriented approach refers to object-oriented software development, which describes software as a set of interacting objects. It uses such models as object class diagrams, sequence diagrams, state charts, and object-oriented programming (OOP).

Finally, Agile development is discussed as a philosophy that guides a development project. It focuses on techniques and methods that encourage more user involvement and allow for more flexible projects with changing requirements.

Learning Objectives

When you have completed this section, you should be able to do the following:



1. Compare the underlying assumptions and uses of a predictive and an adaptive SDLC.
2. Describe the key activities and tasks of information system support.
3. Explain what comprises a system development methodology—the SDLC as well as models, tools, and techniques.
4. Describe the two overall approaches used for software construction and modeling: the structured approach and the object-oriented approach.
5. Describe the key features of Agile development.

Learning Objective 1

Compare the underlying assumptions and uses of a predictive and an adaptive SDLC.

Objective Readings/Activities



Readings: Please read the following section in Chapter 8 of the textbook:

- The System Development Life Cycle (pages 227–232)



Review: Please check your learning with Review Questions 1–16 on page 248 of the textbook.

Learning Objective 2

Describe the key activities and tasks of information system support.

Objective Readings/Activities



Readings: Please read the following section in Chapter 8 of the textbook:

- The Support Phase (pages 232–233).



Review: Please check your learning with Review Questions 17 and 18 on page 248 of the textbook.

Learning Objective 3

Explain what comprises a system development methodology—the SDLC as well as models, tools, and techniques.

Objective Readings/Activities



Readings: Please read the following section in Chapter 8 of the textbook:

- Methodologies, Models, Tools, and Techniques (pages 233–236)



Review: Please check your learning with Review Questions 19–22 on page 248 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 4

Describe the two overall approaches used for software construction and modeling: the structured approach and the object-oriented approach.

Objective Readings/Activities



Readings: Please read the following section in Chapter 8 of the textbook:

- Two Approaches to Software Construction and Modeling (pages 236–244)



Review: Please check your learning with Review Questions 23–28 on page 248 of the textbook.

Learning Objective 5

Describe the key features of Agile development.

Objective Readings/Activities



Readings: Please read the following section in Chapter 8 of the textbook:

- Agile Development (pages 244–247)



Review: Please check your learning with Review Questions 29–32 on page 248 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Section 2: Project Planning and Project Management

Project management encompasses the skills and techniques that are necessary to succeed in planning and managing the development of a new system. As a knowledgeable worker and problem solver, you will need both technical and management skills to be a contributing member of a system development team. This unit provides you with the fundamentals of *project planning* and *project management*.

Learning Objectives

When you have completed this section, you should be able to do the following:

1. Describe the factors that cause a software development project to succeed or fail.
2. Describe the responsibilities of a project manager.
3. Describe the knowledge areas in the project management body of knowledge (PMBOK).
4. Describe the Agile approach to the project management knowledge areas.
5. Explain the activities required to get a project approved (Core Process 1).
6. Explain the activities required to plan and monitor a project (Core Process 2).



Learning Objective 1

Describe the factors that cause a software development project to succeed or fail.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 9 of the textbook:

- Blue Sky Mutual Funds: A New Development Approach (page 254)
- Overview (pages 254–255)
- Principles of Project Management (page 255)
- The Need for Project Management (pages 255–256)



Review: Please check your learning with Review Questions 1 and 2 on page 283 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 2

Describe the responsibilities of a project manager.

Objective Readings/Activities



- Readings: Please read the following sections in Chapter 9 of the textbook:
 - The Role of the Project Manager (pages 256–257)
 - Project Management and Ceremony (pages 257–258)



- Review:
Please check your learning with Review Questions 3–7 on page 283 of the textbook.



- Supplementary Readings:
Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 3

Describe the knowledge areas in the project management body of knowledge (PMBOK).

Objective Readings/Activities



Readings: Please read the following section in Chapter 9 of the textbook:

- Project Management Body of Knowledge (pages 258–259)



Review: Please check your learning with Review Questions 8 on page 283 of the textbook.

Learning Objective 4

Describe the Agile approach to the project management knowledge areas.

Objective Readings/Activities



Readings: Please read the following section in Chapter 9 of the textbook:

- Agile Project Management (pages 259–262)



Review: Please check your learning with Review Questions 9 and 10 on page 283 of the textbook.

Learning Objective 5

Explain the activities required to get a project approved (Core Process 1).

Objective Readings/Activities



Readings: Please read the following section in Chapter 9 of the textbook:

- Activities of Core Process 1: Identify the Problem and Obtain Approval (pages 262–271)



Review: Please check your learning with Review Questions 11–18 on pages 283–284 of the textbook.

Learning Objective 6

Explain the activities required to plan and monitor a project (Core Process 2).

Objective Readings/Activities



Readings: Please read the following section in Chapter 9 of the textbook:

- Activities of Core Process 2: Plan and Monitor the Project (pages 271–282)



Review: Please check your learning with Review Questions 19–23 on page 284 of the textbook.

Unit 2 – Summary and Unit Activities



Readings:

- Read the Chapter 8 Summary on page 247 of the textbook.
- Read the Chapter 9 Summary on page 283 of the textbook.



Review:

- Review the Key Terms on pages 248 and 283, and check if you know what they mean.
 - Answer question 12 from Problems and Exercises in Chapter 8 on page 249 of the textbook.
([Suggested solutions are available here.](#))
 - Read “Community Board of Realtors” in Chapter 8, and answer questions 1–3 on page 250 of the textbook
([Suggested solutions are available here.](#))
 - Answer questions 1 and 2 from Problems and Exercises in Chapter 9 on page 284 of the textbook.
([Suggested solutions are available here.](#))
-

Unit 2: Approaches to Systems Development and Project Management

Answers for Unit Activities

Problems and Exercises (Chapter 8)

- 12. Visit the Web sites of a few leading information systems consulting firms. Try to find information about their approaches to developing systems. Are their SDLCs described? Do the sites mention any tools, models, or techniques?**

Answers will vary. If it is a consulting firm, they might publish a lot about their methodology (or even sell their methodology to clients). Some will brag about the state-of-the-art approach that they use. Some will be vague about it.

Running Case Study: Community Board of Realtors (Chapter 8)

- 1. Compared to the Tradeshow application described in Chapter 1, how long might this project take, and which approach to the SDLC would be most appropriate?**

Answers will vary.

In the Tradeshow system in Chapter 1, we illustrated an iteration as a week-long mini-project. In reality it would probably require a normal iteration of 3 or 4 weeks. The other subsystem would probably also require an iteration of 3 or 4 weeks. The Community Board of Realtors is a comparable sized system – maybe a little larger with about 12 to 15 use cases. It will probably require three or four iterations.

The Community Board of Realtors system is also a well defined requirement. So rather than doing it as an adaptive project with iterations, it could very easily be done as a single predictive type of project. In fact, probably most developers would proceed with a normal modified waterfall approach to this system.

- 2. If you use a predictive SDLC, how much time might each phase of the project take? How much overlap of phases might you plan for? Be specific about how you would overlap the phases.**

Answers will vary.

Using a predictive approach with a modified waterfall approach some estimates (guesses) of time required:

Planning: Schedule, cost estimate, work team – probably a week.
Analysis: Use cases and domain models – probably around 3 or 4 weeks.
Design: Packages, network, domain model, user interfaces, use case realization (use case design) – probably another 3 or 4 weeks.
Implementation: Programming and testing – probably about 6 to 8 weeks.

Overlapping the last three phases is desirable. There are three primary users, and working on the new system can be partitioned by those use cases. The analysis phase could focus first on the domain model, which impacts all use cases. Then it could focus on the use cases associated with the MLS office and begin designing and implementing those use cases. Additional analysis, design, and implementation could then follow for the Real Estate Office and the RE Agent.

- 3. If you use an adaptive SDLC, how many iterations might you plan to include? What use cases would you analyze, design, and implement in the first iteration? What use cases would you work on in the second iteration? In additional iterations? Think in terms of getting the core functionality implemented early and then building the supporting functionality.**

Answers will vary.

Assuming about 12 to 15 use cases, this project would probably be handled in three or four iterations. (The list of use cases developed in Chapter 2 will need to be revisited and probably expanded. CRUD analysis would indicate if enough use cases have been defined to maintain and report data. There may need to be some more 'R'eporting use cases defined.) The most important data for this application is the property listing data. However, the property listing information is dependent on real estate offices and real estate agents. The interdependencies are quite tight, and it does not make sense to try to deploy part of the system early. The core functionality is to build and maintain the database. The supporting functionality is to report and view the data. Hence a possible solution might be as follows:

Iteration 1: Use cases to create and update real estate offices and agents
Iteration 2: Use cases to create and update property listings
Iteration 3: Use cases to produce views and reports of the data

Problems and Exercises (Chapter 9)

- 1. Read this description and then make a list of expected business benefits that the company might derive from a new system:**

Especially for You Jewelers is a small jewelry company in a college town. Over the last couple of years, it has experienced a tremendous increase in its business. However, its financial performance hasn't kept pace with its growth. The current system, which is partly manual and partly automated, doesn't track accounts receivables sufficiently, and the company is finding it difficult to determine why the receivables are so high. It runs frequent specials to attract customers, but it has no idea whether these are profitable or if the benefit—if there is one—comes from associated sales. Especially for You wants to increase repeat sales to its existing customers; thus it needs to develop a customer database. It also wants to install a new direct sales and accounting system to help solve these problems.

- Reduce the level of accounts receivables.
- Determine which type of specials and promotions increased sales.
- Increase repeat sales to existing customers.
- Closely track financial performance of the store.

2. Read this narrative and then make a list of system capabilities for the company:

The new direct sales and accounting system for Especially for You Jewelers will be an important element in the growth and success of the jewelry company. The direct sales portion needs to track every sale and be able to link to the inventory system for cost data to provide a daily profit and loss report. The customer database needs to be able to produce purchase histories to assist management in preparing special mailings and special sales to existing customers. Detailed credit balances and aged accounts for each customer would help solve the problem with the high balance of accounts receivables. Special notice letters and credit history reports would help management reduce accounts receivable.

- Track individual sales.
- Report on cost data for inventory items.
- Produce daily profit and loss reports.
- Track purchase histories of individual customers.
- Produce special mailings.
- Maintain accounts aging with reporting.

Unit 3: Systems Analysis Activities

This unit focuses on investigating and modelling system requirements, and gives an overview of the activities of systems analysis. In the first section, system requirements are defined, and the different types of requirements that analysts encounter are explored. In the subsequent sections, the importance of creating models and how to model system requirements in detail through the creation of activity diagrams, use case diagrams, domain model class diagrams, system sequence diagrams, and state machine diagrams is explained.

Section 1: Investigating System Requirements

In this section, we discuss how to investigate system requirements. First, an overview is provided on system analysis activities. Then, some important concepts are described such as *functional* and *non-functional requirements*, the role of *models* in system analysis, and the role of *stakeholders* and their contributions in the requirements definition process. Finally, this section discusses techniques on how to *gather information* about system requirements and how to create an *activity diagram* to model workflows.

Learning Objectives

When you have completed this section, you should be able to do the following:

1. Describe the activities of systems analysis.
2. Explain the difference between functional and nonfunctional requirements.
3. Describe the role of models in systems analysis.
4. Identify and understand different kinds of stakeholders and their contributions to requirements definition.
5. Describe information-gathering techniques, and determine when each is best applied.
6. Develop activity diagrams to model workflows.



Learning Objective 1

Describe the activities of system analysis.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 2 of the textbook:

- Mountain Vista Motorcycles (page 36)
- Overview (pages 36–37)
- The RMO Consolidated Sales and Marketing System Project (page 37–40)
- Systems Analysis Activities (pages 40–42)



Review: Please check your learning with Review Questions 1 on page 61 of the textbook.

Learning Objective 2

Explain the difference between functional and nonfunctional requirements.

Objective Readings/Activities



Readings: Please read the following section in Chapter 2 of the textbook:

- What are Requirements? (pages 42–44)



Review: Please check your learning with Review Questions 3 on page 61 of the textbook.

Learning Objective 3

Describe the role of models in systems analysis.

Objective Readings/Activities



Readings: Please read the following section in Chapter 2 of the textbook:

- Models and Modeling (pages 44–46)



Review: Please check your learning with Review Questions 2 on page 61 of the textbook.

Learning Objective 4

Identify and understand different kinds of stakeholders and their contributions to requirements definition.

Objective Readings/Activities



Readings: Please read the following section in Chapter 2 of the textbook:

- Stakeholders (pages 46–48)



Review: Please check your learning with Review Questions 6 on page 61 of the textbook.

Learning Objective 5

Describe information-gathering techniques, and determine when each is best applied.

Objective Readings/Activities



Readings: Please read the following section in Chapter 2 of the textbook:

- Information-Gathering Techniques (pages 48–57)



Review: Please check your learning with Review Questions 4, 5, 7, 8 on page 61 of the textbook.

Learning Objective 6

Develop activity diagrams to model workflows.

Objective Readings/Activities



Readings: Please read the following section in Chapter 2 of the textbook:

- Documenting Workflows with Activity Diagrams (page 57–60)



Review: Please check your learning with Review Questions 9 and 10 on page 61 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Section 2: Use Cases

Virtually all newer approaches to system development begin the requirements modelling process with the concept of a *use case*. A use case is an activity the system performs, usually in response to a request by a user. In this section, two techniques for identifying use cases are described, and guidelines for creating a *use case diagram* are provided.

Learning Objectives

When you have completed this section, you should be able to do the following:



1. Explain why identifying use cases is the key to defining functional requirements.
2. Apply the user goal technique to identify use cases.
3. Apply the event decomposition technique to identify use cases.
4. Apply the CRUD technique to validate and refine the list of use cases.
5. Describe the notation and purpose for the use case diagram.

Learning Objective 1

Explain why identifying use cases is the key to defining functional requirements.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 3 of the textbook:

- Waiters on Call Meal-Delivery System (page 68)
- Overview (pages 68–69)



Review: Please check your learning with Review Questions 1–3 on page 85 of the textbook.

Learning Objective 2

Apply the user goal technique to identify use cases.

Objective Readings/Activities



Readings: Please read the following section in Chapter 3 of the textbook:

- Use Cases and User Goals (pages 69–70)



Review: Please check your learning with Review Questions 4–8 on page 85 of the textbook.

Learning Objective 3

Apply the event decomposition technique to identify use cases.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 3 of the textbook:

- Use Cases and Event Decomposition (pages 70–76)



Review: Please check your learning with Review Questions 9–19 on pages 85–86 of the textbook.

Learning Objective 4

Apply the CRUD technique to validate and refine the list of use cases.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 3 of the textbook:

- Use Cases and CRUD (pages 77–78)



Review: Please check your learning with Review Questions 20 and 21 on page 86 of the textbook.

Learning Objective 5

Describe the notation and purpose for the use case diagram.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 3 of the textbook:

- Use Cases in the Ridgeline Mountain Outfitters Case (page 78)
- Use Case Diagrams (pages 78–84)



Review: Please check your learning with Review Questions 22–29 on page 86 of the textbook.

Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Section 3: Domain Modelling

As with use cases, an analyst should ask the users to discuss the types of “things” that they work with routinely and about which the system needs to store information. The analyst can ask about several types of things to help identify them. Many things are tangible and therefore more easily identified, but others are intangible. Different types of things are important to different users, so it is important to include information from all types of users. These things can then be modeled in different ways, depending on the development approach used. In this section, we discuss how to identify and model things in the problem domain.

Learning Objectives

When you have completed this section, you should be able to do the following:

1. Explain how the concept of things in the problem domain also defines requirements.
2. Identify and analyze data entities needed in the system and read, interpret, and create entity-relationship diagrams.
3. Identify and analyze domain classes needed in the system and read, interpret, and create domain model class diagrams.



Learning Objective 1

Explain how the concept of things in the problem domain also defines requirements.

Objective Readings/Activities



Readings: Please read the following section in Chapter 4 of the textbook:

- Waiters on Call Meal-Delivery System (page 92)
- Overview (page 92)
- “Things” in the Problem Domain (pages 92–98)



Review: Please check your learning with Review Questions 1–20 on pages 112–113 of the textbook.

Learning Objective 2

Identify and analyze data entities needed in the system and read, interpret and create entity-relationship diagrams.

Objective Readings/Activities



Readings: Please read the following section in Chapter 4 of the textbook:

- The Entity-Relationship Diagram (pages 98–101)



Review: Please check your learning with Review Questions 21–23 on page 113 of the textbook.

- Supplementary Readings:
Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 3

Identify and analyze domain classes needed in the system and read, interpret, and create domain model class diagrams.

Objective Readings/Activities



Readings: Please read the following section in Chapter 4 of the textbook:

- The Domain Model Class Diagram (pages 101–111)



Review: Please check your learning with Review Questions 24–34 on page 113 of the textbook.

Section 4: Extending the Requirements Models

The previous two sections focused on two primary aspects of modelling functional requirements, use cases and domain model classes. This section goes into more detail on how to model functional requirements, and discusses techniques and models to show additional information, which enables us to provide a more comprehensive description of the system's requirements.

Learning Objectives



1. Write fully developed use case descriptions.
2. Develop activity diagrams to model the flow of activities
3. Develop system sequence diagrams
4. Develop state machine diagrams to model object behavior.
5. Explain how use case descriptions and UML diagrams work together to define functional requirements.

Learning Objective 1

Write fully developed use case descriptions.

Objective Readings/Activities



Readings: Please read the following section in Chapter 5 of the textbook:

- Electronics Unlimited: Integrating the Supply Chain (page 120)
- Overview (pages 120–121)
- Use Case Descriptions (pages 121–124)



Review: Please check your learning with Review Questions 1–8 on page 144 of the textbook.

Learning Objective 2

Develop activity diagrams to model flow of activities.

Objective Readings/Activities



Readings: Please read the following section in Chapter 5 of the textbook:

- Activity Diagrams for Use Cases (pages 125–126)



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 3

Develop system sequence diagrams.

Objective Readings/Activities



Readings: Please read the following section in Chapter 5 of the textbook:

- The System Sequence Diagram—Identifying Inputs and Outputs (pages 126–132)



Review: Please check your learning with Review Questions 9–18 on page 144 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

68

Learning Objective 4

Develop state machine diagrams to model object behavior.

Objective Readings/Activities



Readings: Please read the following section in Chapter 5 of the textbook:

- The State Machine Diagram—Identifying Object Behavior (pages 132–142)



Review: Please check your learning with Review Questions 19–26 on pages 144–145 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 5

Explain how use case descriptions and UML diagrams work together to define functional requirements.

Objective Readings/Activities



Readings: Please read the following section in Chapter 5 of the textbook:

- Integrating Requirements Models (pages 142–143)



Review: Please check your learning with Review Questions 27 on page 145 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Unit 3 – Summary and Unit Activities

Readings: Please read the following sections of the textbook:



- Chapter 2 Summary (page 60)
- Chapter 3 Summary (page 85)
- Chapter 4 Summary (page 112)
- Chapter 5 Summary (pages 143–144)



Review:

- Review the Key Terms on pages 61, 85, 112, and 144, and check if you know what they mean.
- Answer question 8 from Problems and Exercises in Chapter 2 on page 62 of the textbook.
([Suggested solutions are available here.](#))
- Read “Community Board of Realtors” in Chapter 2, and answer questions 1–3 on pages 63–64 of the textbook.
([Suggested solutions are available here.](#))
- Read “Sandia Medical Devices” in Chapter 2, and answer questions 1–5 on pages 65–66 of the textbook.
([Suggested solutions are available here.](#))

- Answer questions 1 and 2 from Problem and Exercises in Chapter 3 on page 86 of the textbook.
([Suggested solutions are available here.](#))
 - Read “Community Board of Realtors” in Chapter 3, and answer questions 1–4 on page 88 of the textbook.
([Suggested solutions are available here.](#))
 - Read “Sandia Medical Devices” in Chapter 3, and answer questions 1–3 on pages 89–90 of the textbook.
([Suggested solutions are available here.](#))
 - Answer questions 1, 2 and 10 from Problem and Exercises in Chapter 4 on pages 113–115 of the textbook.
([Suggested solutions are available here.](#))
 - Read “Community Board of Realtors” in Chapter 4, and answer questions 1–3 on page 116 of the textbook.
([Suggested solutions are available here.](#))
 - Read “Sandia Medical Devices” in Chapter 4, and answer questions 1 and 2 on pages 117–118 of the textbook
([Suggested solutions are available here.](#))
 - Answer questions 1, 4, and 6 from Problem and Exercises in Chapter 5 on pages 145–146 of the textbook.
([Suggested solutions are available here.](#))
 - Read “Community Board of Realtors” in Chapter 5, and answer questions 1–3 on page 148 of the textbook.
([Suggested solutions are available here.](#))
 - Read “Sandia Medical Systems Real-Time Glucose Monitoring” in Chapter 5, and answer questions 1–3 on page 149–150 of the textbook.
([Suggested solutions are available here.](#))
-

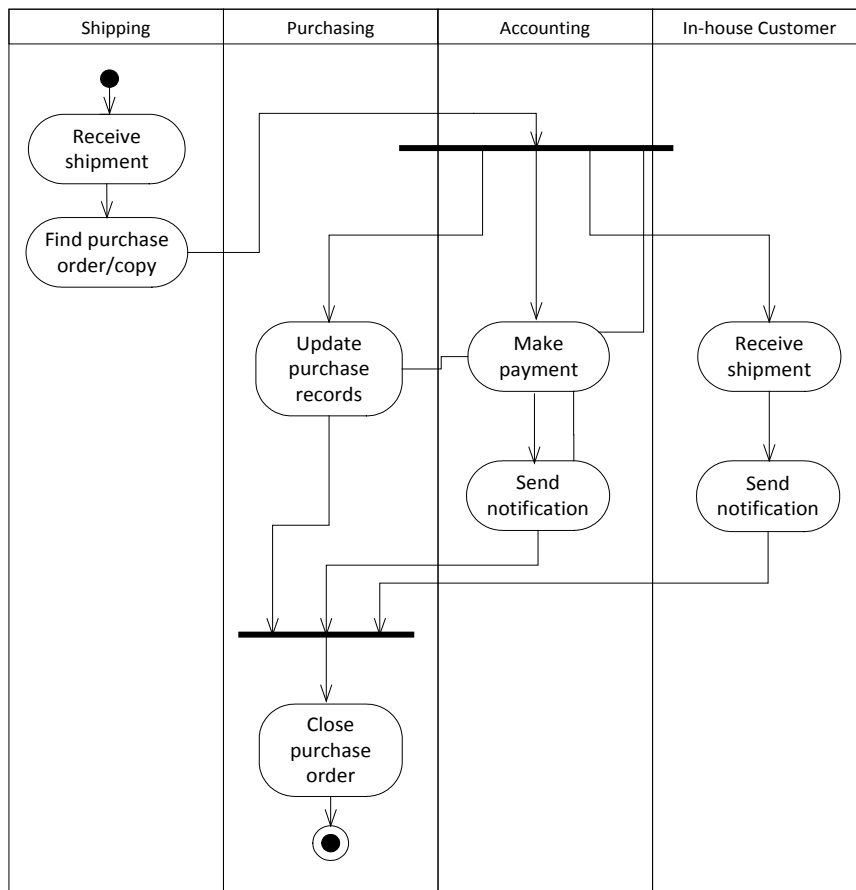
Unit 3: Systems Analysis Activities

Answers for Unit Activities

Problems and Exercises (Chapter 2)

8. Develop an activity diagram based on the following narrative. Note any ambiguities or questions that you have as you develop the model. If you need to make assumptions, also note them.

The shipping department receives all shipments on outstanding purchase orders. When the clerk in the shipping department receives a shipment, he or she finds the outstanding purchase order for those items. The clerk then sends multiple copies of the shipment packing slip. One copy goes to Purchasing, and the department updates its records to indicate that the purchase order has been fulfilled. Another copy goes to Accounting so a payment can be made. A third copy goes to the requesting in-house customer so he or she can receive the shipment. After payment is made, the accounting department sends a notification to Purchasing. After the customer receives and accepts the goods, he or she sends notification to Purchasing. When Purchasing receives these other verifications, it closes the purchase order as fulfilled and paid.



Running Case Studies: Community Board of Realtors (Chapter 2)

1. Who are the stakeholders for the issues related to real estate in your community, and what are their main interests?

Answers will vary. Usually for each state/county there are such organizations as the following:

- A division of real estate for the state, often in a state's department of commerce
- A state association of realtors
- A state multiple listing service
- Local realtor boards and associations
- Real estate offices and agents

2. What types of information does the board collect and make available to its members and to the community?

Answers will vary.

- Real estate offices:
 - Collects information about offices
 - Collects information about agents and brokers
 - Provides search and display of offices, agents, and brokers with addresses
 - Provides services to real estate offices, such as advertising, training, banking information, lender information, etc.
- Real estate listings:
 - Collects Information about property listings
 - Collects Information about sales of property
 - Provides search and display about listed properties
 - Provides maps of listed properties

3. Research the real estate industry in at least two countries other than the United States. For each of these countries, what are some of the cultural and legal issues that differ from those in the United States? If you were working on support for an international real estate cooperative system, in what ways would the information collection activity process be complicated?

Answers will vary.

Running Case Studies: Sandia Medical Devices (Chapter 2)

- 1. Who are RTGM's stakeholders? Should NMHS's patients be included in defining the system requirements? Why or why not? Should RTGM interact with medical professionals other than physicians? Why or why not?**

Stakeholders include the following:

- Patients (users of the monitoring device and of the phone application)
- Doctors (users of the information transmitted)
- Other medical staff (those who enter patient medical information on phone app)
- Medical equipment engineers (developers of the mobile monitoring equipment)
- Technical staff (developers of the central server system, and the database)
- Project team members (developers of the phone app)

Patients should be included in defining system requirements for those areas that impact their use. This would include the medical device (comfort, wearability, maintenance), and the phone app (installing, executing, user interface screen).

Medical professionals other than physicians may need to be involved. This would include medical staff that are involved in the design of the device (Is it sensitive enough to read glucose levels, etc.?). Other medical professionals may want to research results for medical studies. As such they will dictate what data should be captured.

- 2. If you were the lead analyst for RTGM, how would you determine the requirements? Be specific in your answer. List several questions you need answered.**

Assuming we are limiting the fact finding and requirements definition to the smartphone app:

Questions for the medical equipment engineers would include the specifications for the device – range, transmit parameters, data formats, occurrence of transmittal, ranges of values for normal, abnormal, and dangerous; what “test” capabilities it has; and how it is enabled.

Questions for the patient would be about the user interface for the phone app – how readable, how understandable it is; signalling for normal and abnormal conditions; message format; how usable the screens are for entering data, reading information, “testing” the equipment.

Questions for the medical professionals would include information about the data formats to be sent, how often data should be sent, how to notify the system about abnormal/emergency situations, what other patient information needs to be captured.

3. What are the primary functional requirements for the system as described so far in the case?

- Enter user (patient) information
- Test monitoring device
- Receive monitor-device data
- Send monitoring data to server
- Receive data from server
- Alert patient (user) of abnormal situation

4. Are the parameters for alerting patients and medical personnel the same for every patient? Can they vary over time for the same patient? What are the implications for the system's functional requirements?

The case does not describe medical parameter variation by patient, but it may be assumed that the severity of illness, weight of the patient, or sex of the patient may impact the acceptable and dangerous levels. Hence entering patient information may need to be done by trained medical personnel, or it needs to be accessed from the central server.

The case does not address whether alerts can change over time. But assuming that the severity of the illness will cause the parameters to change, patient information should be updated as appropriate.

The functional requirements may need to change to access history information from the server to automatically update alert levels.

5. Briefly describe some possible nonfunctional requirements for RTGM.

Usability – Patients may be completely non-technical, may also be ill, may be disabled. Usability needs to be assessed very carefully.

Reliability – Both the monitoring device and the phone app must be error free. In addition, some type of fail-safe capability should be built in. For example, if the monitoring device fails to communicate, the phone app should sound an alert. The phone app should have a “normal operation” icon showing at all times (maybe a green light).

Performance – Probably not a problem for the phone app. Performance, e.g., throughput, will need to be evaluated for the central server and the telephone connectivity.

Security – All medical information must conform to HIPAA requirements. Transmittal of data over phone lines should be encrypted. Care should be taken so that multiple devices that are located near to each other do not interfere.

The “+” requirements should also be addressed. How easy is it to install the phone app? How much memory does it require? What kinds of devices is it compatible with? How does it interface with the server application?

Problems and Exercises (Chapter 3)

- 1. Review the external event checklist in Figure 3-3 and then think about a university course registration system. What is an example of an event of each type in the checklist? Name each event by using the guidelines for naming an external event.**

External agent wants something – Student registers for a section of a course

External agent wants some information – Student searches for a course

Data changed and needs to be updated – Instructor assigned to teach a course section

Management wants some information – Show enrollments for all courses in a department

- 2. Review the temporal event checklist in Figure 3-4. Would a student grade report be an internal or external output? Would a class list for the instructor be an internal or external output? What are some other internal and external outputs for a course registration system? Using the guidelines for naming temporal events, what would you name the events that trigger these outputs?**

Grade report: External

Class list: Internal

Other external: Financial aid confirmation, graduation notice, employment confirmation letter

Other internal: Enrollment report and paycheck

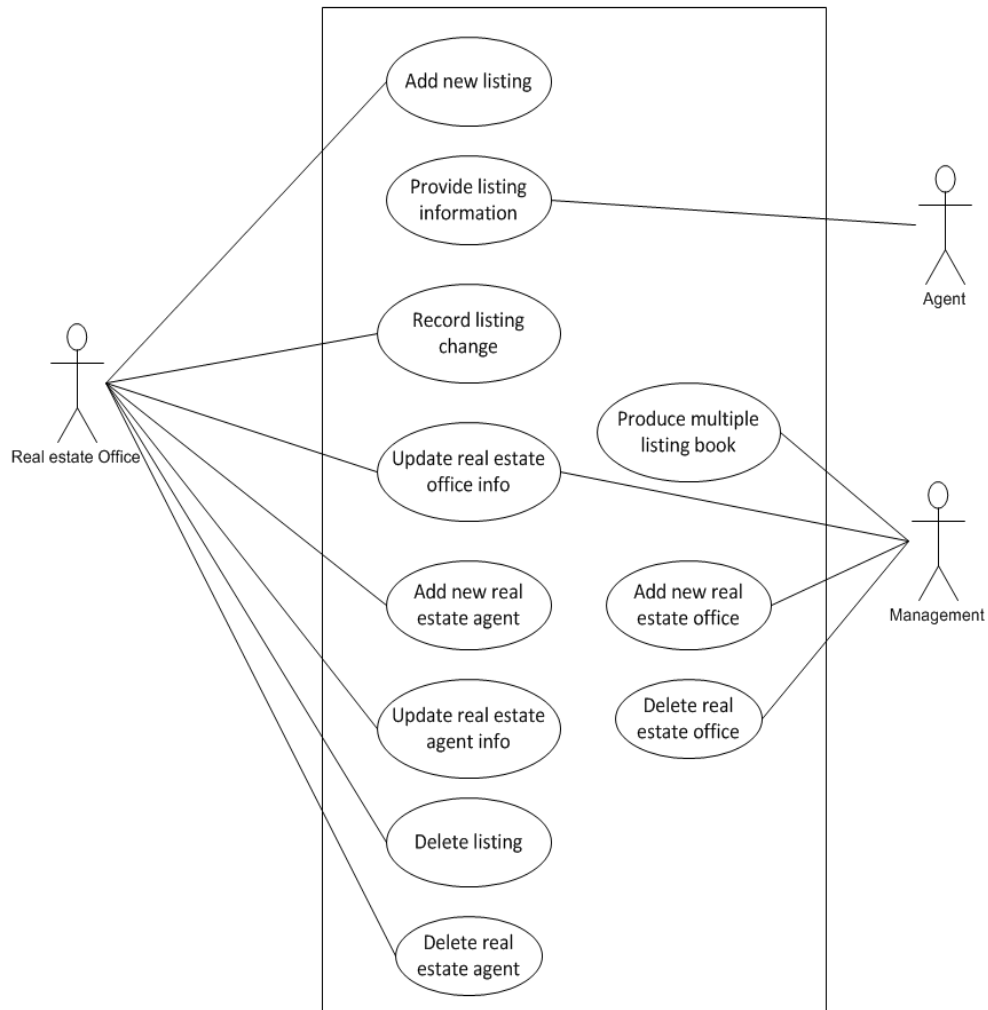
To name the temporal events, include *Time to produce* with the output name, as well as the recipient. For example, *Time to produce grade report for students*. Others are similar.

Running Case Studies: Community Board of Realtors (Chapter 3)

- 1. To what events must the MLS system respond? List each event, the type of event, and the resulting use case. Be sure to consider all the use cases that would be needed to maintain the data in the MLS system, thinking in terms of the CRUD technique.**

Event	Type	Use case
Real estate office submits new listing	External	Add new listing
Agent requests listing information	External	Provide listing information
Time to produce multiple listing book	Temporal	Produce multiple listing book
Real estate office submits listing change request	External	Record listing change
New real estate office opens (implied)	External	Add new real estate office
Change real estate office information	External	Update real estate office info
New agent is hired (implied)	External	Add new real estate agent
Change agent information	External	Update real estate agent info
House is sold (from CRUD)	External	Delete listing
Real estate office closes (from CRUD)	External	Delete real estate office
Agent retires/quits (from CRUD)	External	Delete real estate agent

2. Draw a use case diagram based on the actors and use cases you identified in question 1.

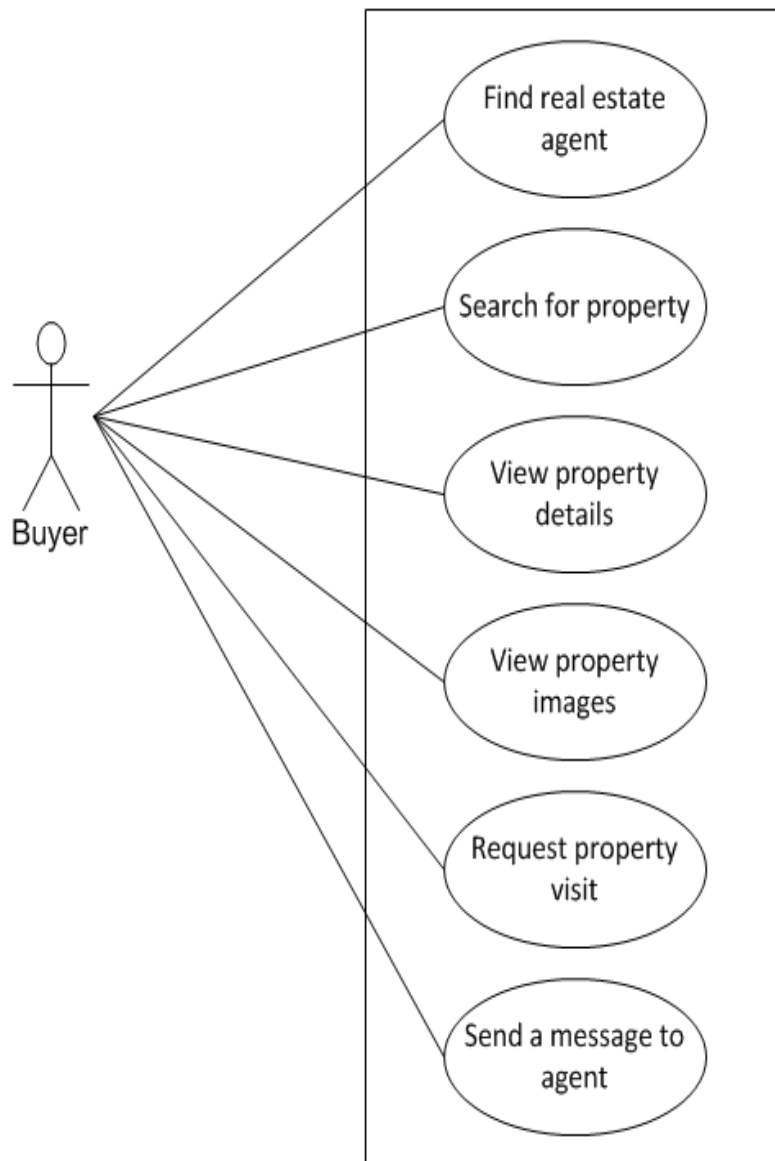


- 3. Given the information available in the system, consider yourself a potential customer looking for real estate. List as many specific use cases you would like to see based on your specific goals.**

Answers will vary.

- Find real estate agent
- Search for property (by various criteria)
- View property details
- View property images (video or pictures)
- Request a property visit
- Send a message to real estate agent

4. Draw a use case diagram for all the use cases for the potential customer you identified in question 3.



Running Case Studies: Sandia Medical Devices (Chapter 3)

1. Identify all the actors that will use RTGM.

Patient

Health-care provider (physician)

Nurse (physician assistant)

2. Using the actors that you identified in question 1, develop a list of use cases based on the user goal technique. Draw a use case diagram for these use cases.

Patient:

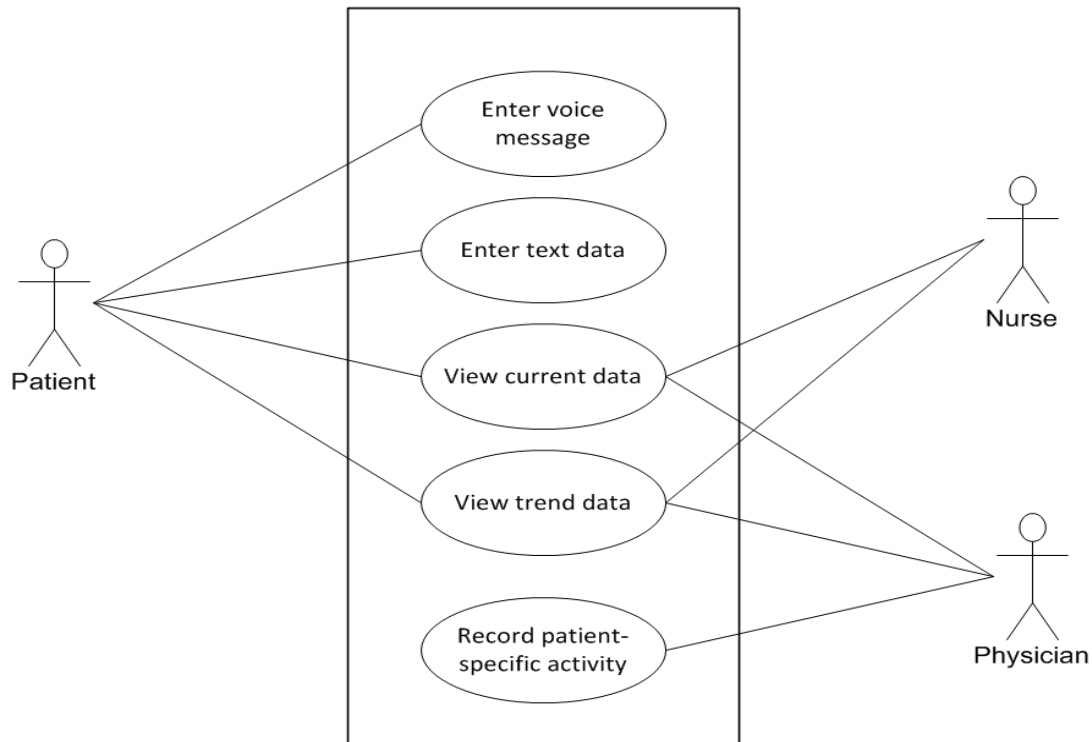
- View current data
- View trend data
- Enter text message
- Enter voice message

Nurse:

- View current data
- View trend data

Physician:

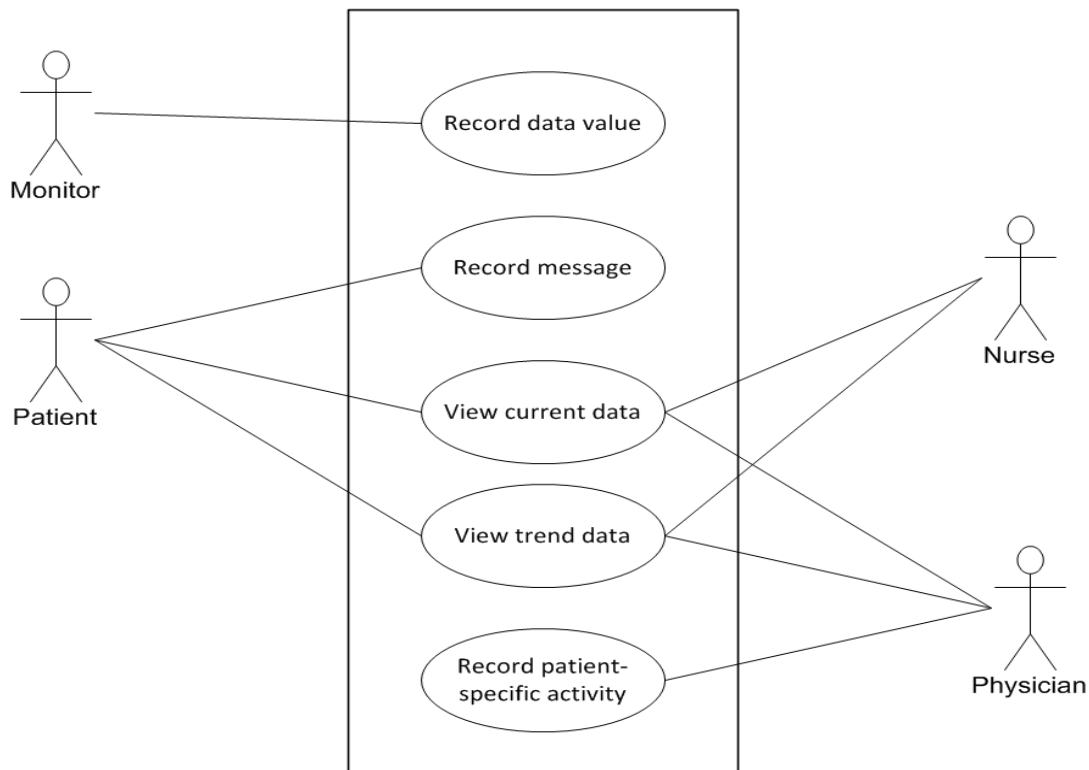
- View current data
- View trend data
- Record patient-specific activity



3. Using the event decomposition technique for each event you identified in the description, name the event, state the type of event, and name the resulting use case. Draw a use case diagram for these use cases.

Event	Type	Use Case
Monitor records data value	Temporal	Record data value
Patient views current value	External	View data value
Patient views trend chart	External	View trend data
Nurse views current value	External	View data value
Nurse views trend chart	External	View trend data
Physician views current value	External	View data value
Physician views trend chart	External	View trend data
Physician responds to data	External	Record activity
Patient responds to data	External	Record message (voice or text)

Note: Physician activity and patient activity may be quite different; hence we have identified separate use cases.

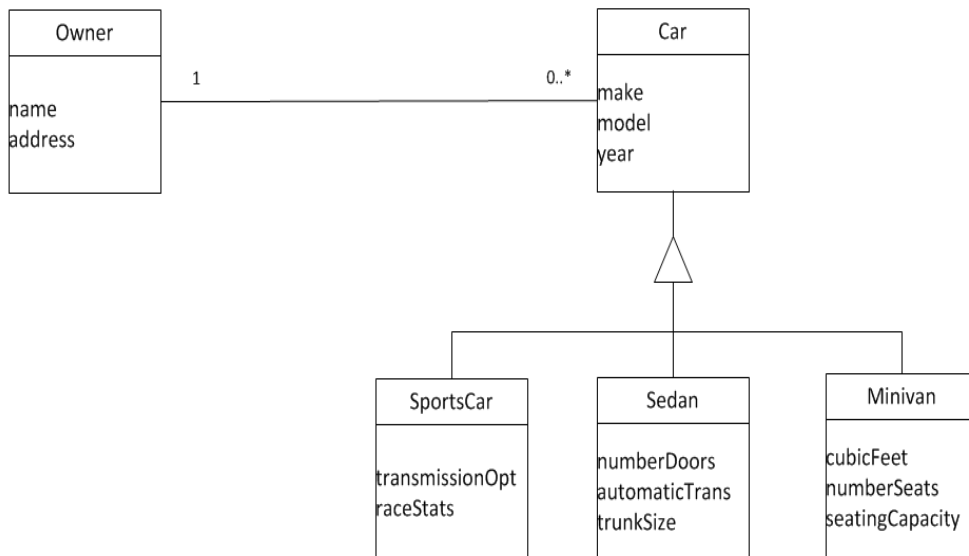


Problems and Exercises (Chapter 4)

1. Draw an entity-relationship diagram, including minimum and maximum cardinality, for the following: The system stores information about two things: cars and owners. A car has attributes for make, model, and year. The owner has attributes for name and address. Assume that a car must be owned by one owner and an owner can own many cars, but an owner might not own any cars (perhaps she just sold them all, but you still want a record of her in the system).



2. Draw a class diagram for the cars and owners described in exercise 1, but include subclasses for sports car, sedan, and minivan, with appropriate attributes.



10. Consider the domain model class diagram shown in Figure 4-24, which includes classes for college, department, and faculty members.

- a. What kind of UML relationships are shown in the model?

There are two binary association relationships.

- b. How many attributes does a “faculty member” have? Which (if any) have been inherited from another class?

FacultyMember has five attributes, none inherited.

- c. If you add information about one college, one department, and four faculty members, how many objects do you add to the system?

You add six objects.

d. Can a faculty member work in more than one department at the same time? Explain.

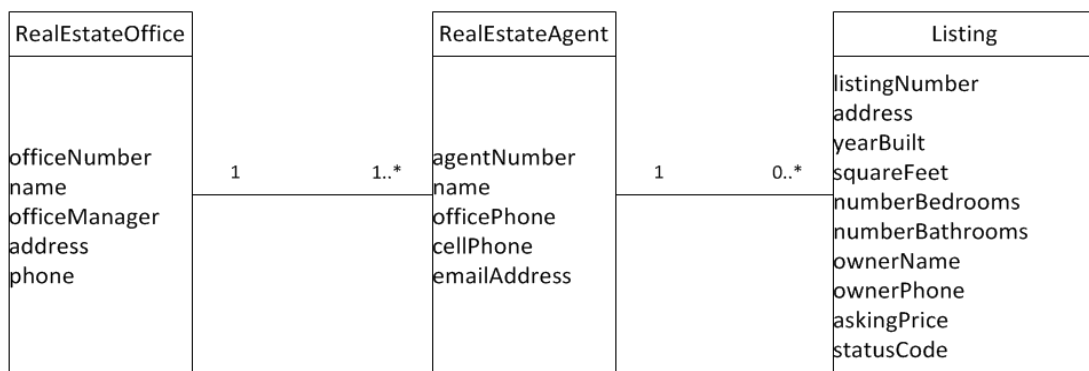
Using minimum and maximum multiplicity (cardinality), we can say that a faculty member can be in more than one department at the same time. In the real world, for example, one teacher can be part of both the computer science and computer information systems departments (a split appointment is possible).

e. Can a faculty member work in two departments at the same time, where one department is in the college of business and the other department is in the college of arts and sciences? Explain.

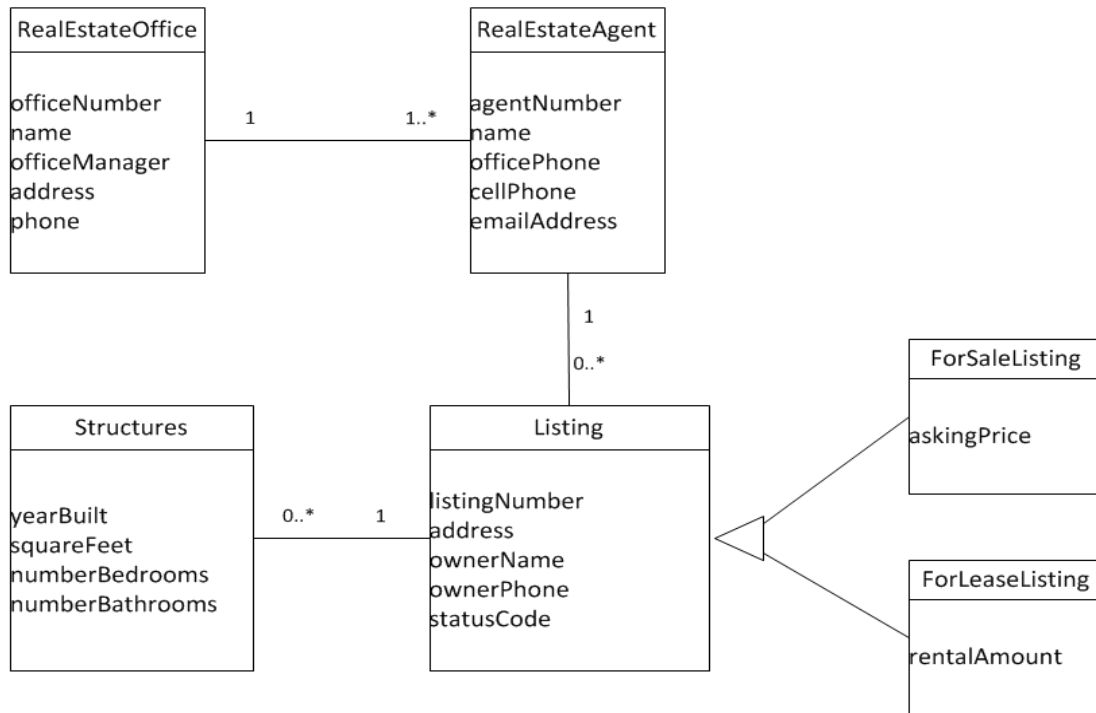
This is a good question to discuss in the Unit 3 Discussion Forum. Draw a circle for a FacultyMember object (Billy Bob) and two circles for two separate Department objects (computer science and CIS). Connect the FacultyMember to each Department with two lines. These lines associate one FacultyMember object with two Department objects. Next, draw two College objects (business and sciences). Connect computer science to sciences and CIS to business. Now it should be clear that a faculty member can be part of two departments at the same time.

Running Case Studies: Community Board of Realtors (Chapter 4)

- Based on the information here, draw a domain model class diagram for the MLS system. Be sure to consider what information needs to be included versus information that is not in the problem domain. For example, is detailed information about the owner, such as his employer or his credit history, required in the MLS system? Is that information required regarding a potential buyer?**



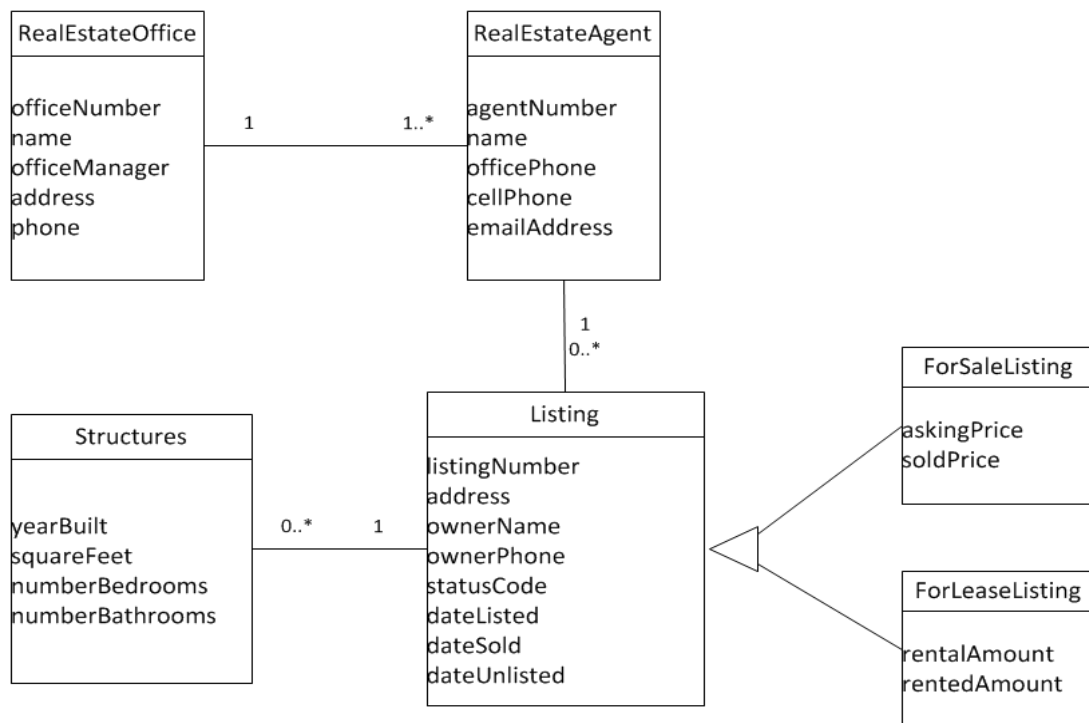
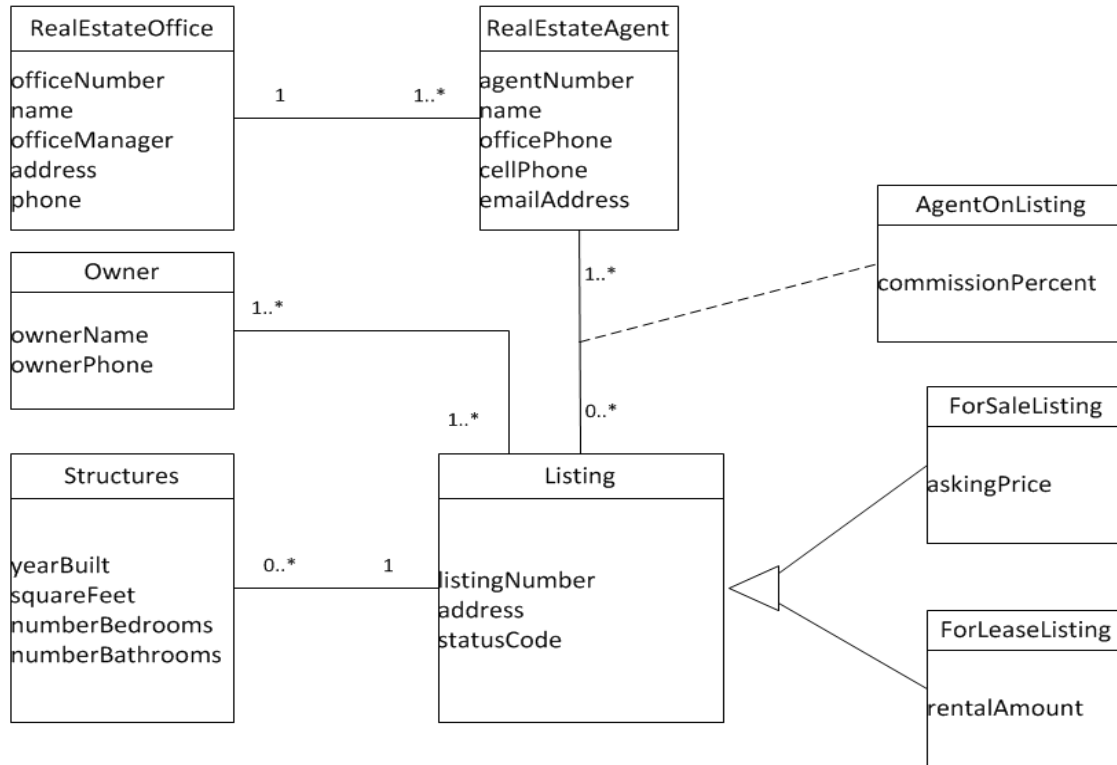
2. Draw a second domain model class diagram that adds the following specifications. First, there are two types of listings: a listing for sale and a listing for lease. Additionally, a listing might include no structures, such as vacant land, or it might include more than one structure, such as a main house and a guest house, each with separate values for square footage, number of bedrooms, and number of bathrooms.



Note: Here is an additional solution to Question 2 with a few more attributes added that are not requested by the case, but that will be helpful for later chapters. It adds some attributes to produce more reports.

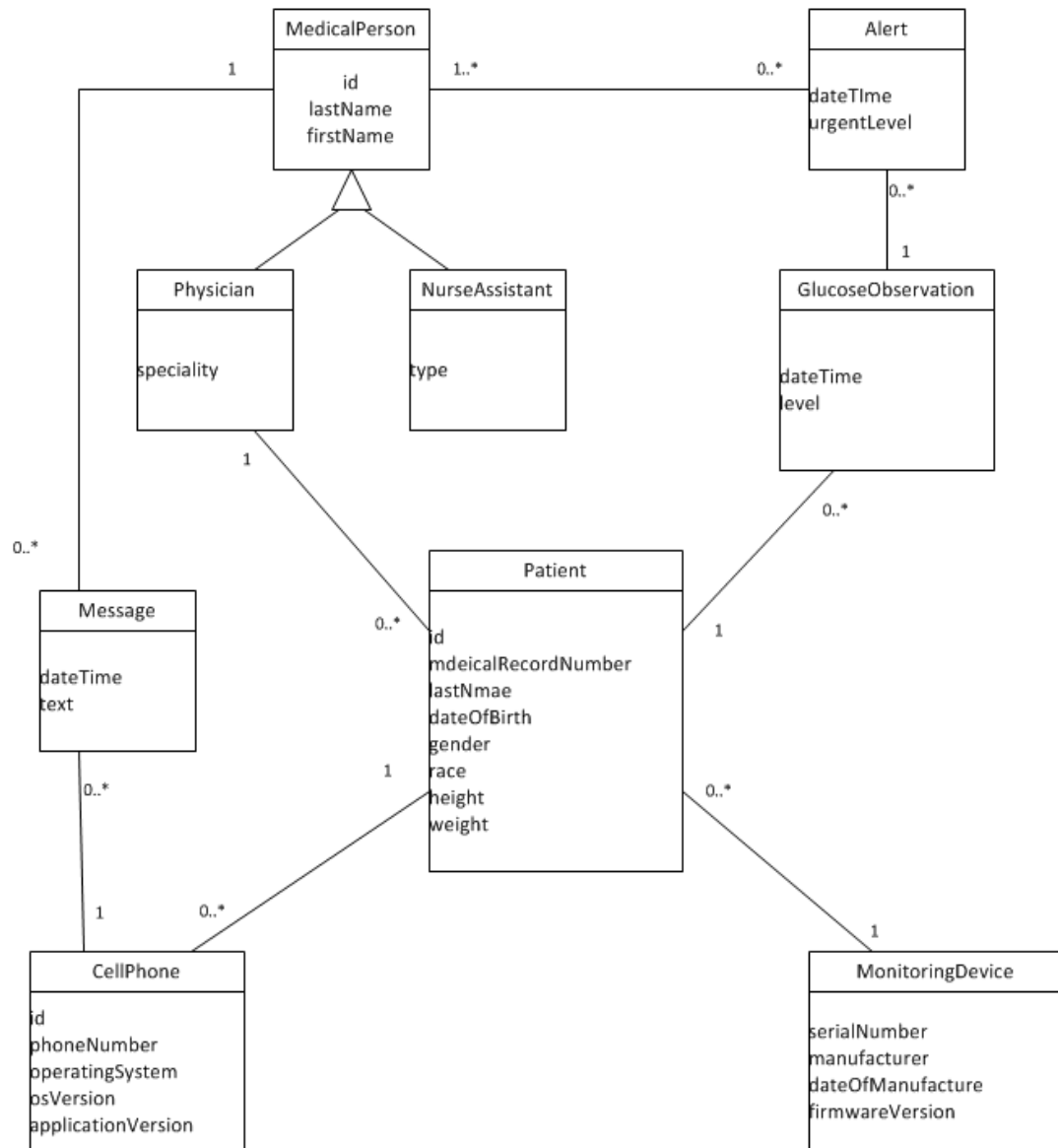
3. Draw a third domain model class diagram that assumes a listing might have multiple owners. Additionally, a listing might be shared by two or more agents, and the percentage of the commission that each agent gets from the sale can be different for each agent.

Unit 3: Systems Analysis Activities
Answers for Unit Activities



Running Case Studies: Sandia Medical Devices (Chapter 4)

1. **Modify the diagram (Figure 4-25) to incorporate the changes under consideration. You may need to use association classes and generalization/specialization (inheritance).**



2. **Are a set of abstract and concrete classes needed to represent variations among cell phones? Why or why not?**

Answers will vary.

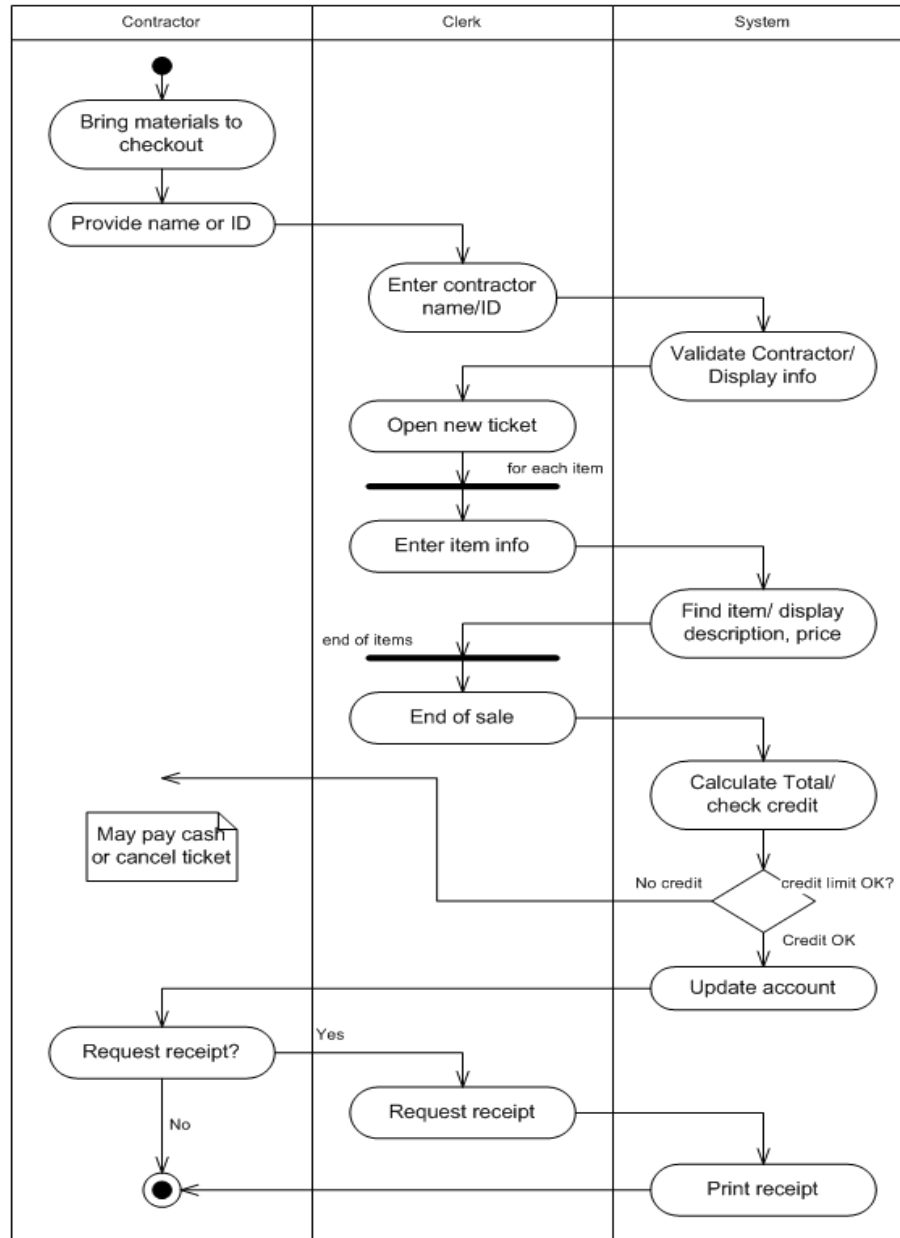
Possibly to distinguish between different types of cell phones, a general abstract class and specialized concrete classes might be necessary.

Problems and Exercises (Chapter 5)

- 1. After reading the following narrative, do the following:**
 - i. Develop an activity diagram for each scenario.**
 - ii. Complete a fully developed use case description for each scenario.**

Contractor Sale activity diagram:

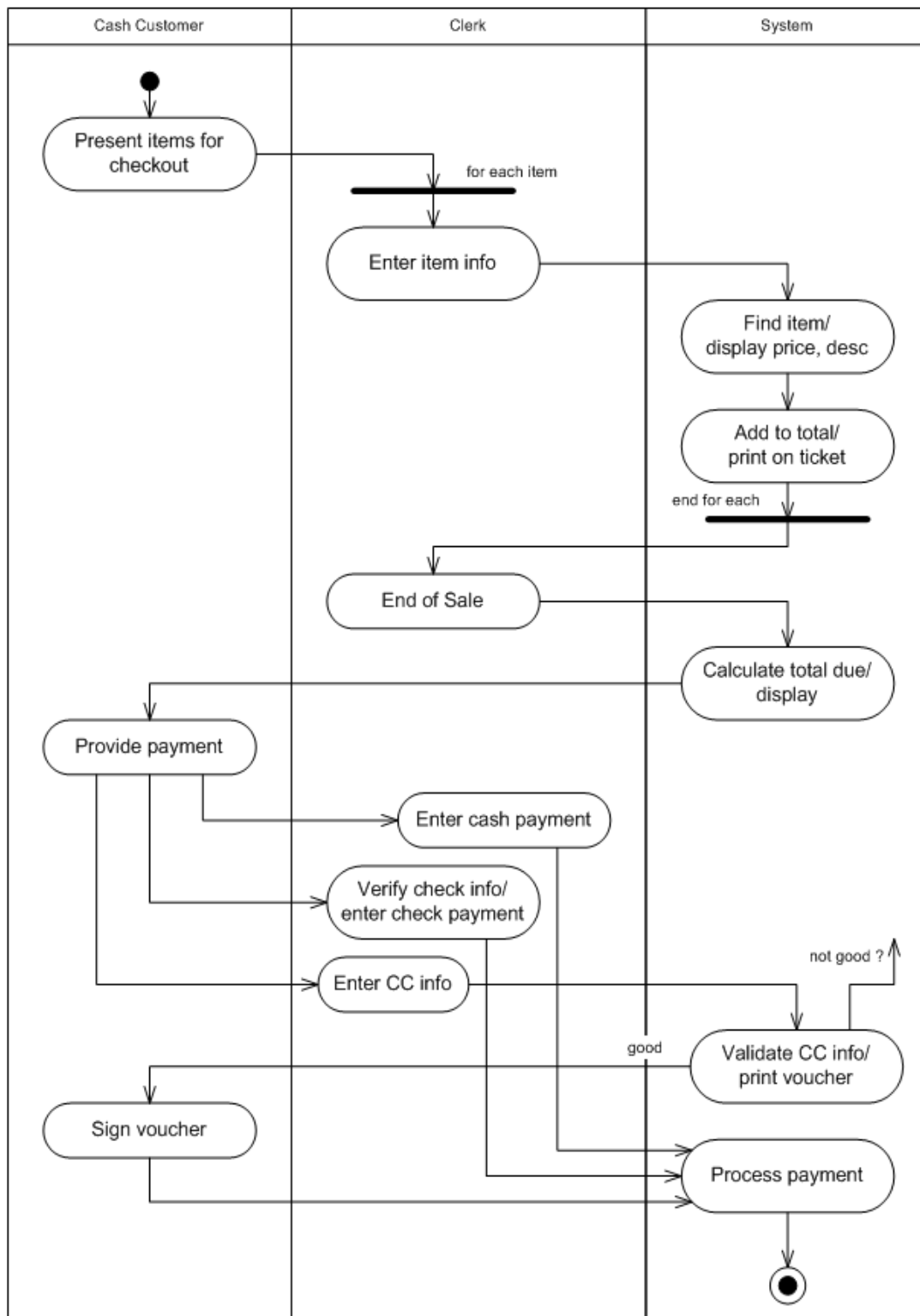
Unit 3: Systems Analysis Activities
Answers for Unit Activities



Contractor Sale fully developed use case description:

Use Case Name:	Create a new sale	
Scenario:	Create new sale to a contractor (on account sale)	
Triggering Event:	Contractor wants to purchase items.	
Brief Description:	A contractor wants to purchase items. The clerk rings up the items and then adds them to the contractor's account.	
Actors:	Sales clerk	
Stakeholders:	Sales clerk Accounting department Sales department	
Preconditions:	Customer account must exist. Inventory items must exist.	
Postconditions:	New sale is created. Sales line items are created and connected to the sale. Customer (contractor) account is updated.	
Flow of Activities:	Actor	System
	1. Clerk enters contractor ID. 2. Clerk enters each item. 3. Clerk indicates the end of the sale. 4. If contractor wants receipt, requests receipt.	1.1 System validates contractor account. 2.1 System finds item in inventory, finds price, adds to total. 3.1 System calculates total and adds to contractor account. 4.1 System prints receipt.
Exception Conditions:	1.1 If contractor account is out of balance, treat this sale as a cash sale, or stop process and send contractor to accounting clerk. 2.1 If system has information missing, sales clerk calls manager and manually enters information. 3.1 If contractor account balance is over the limit, treat as cash sale, cancel, or send contractor to accounting clerk.	

Sale to public activity diagram:

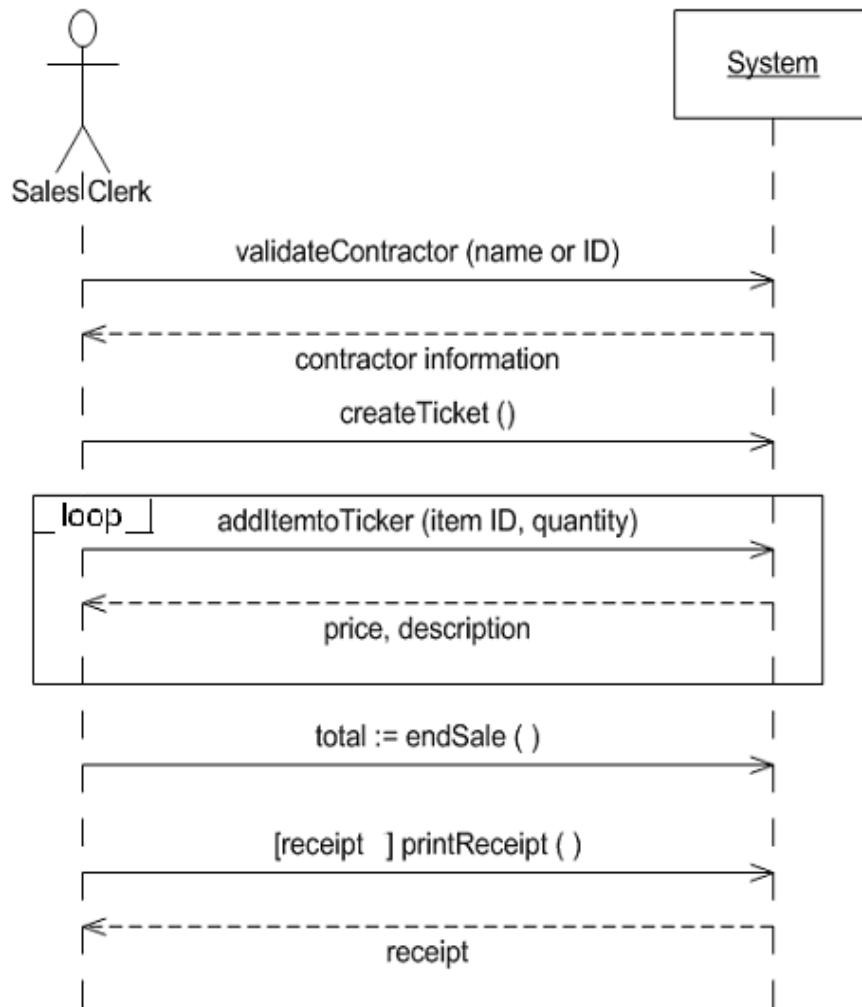


Sale to public fully developed use case description:

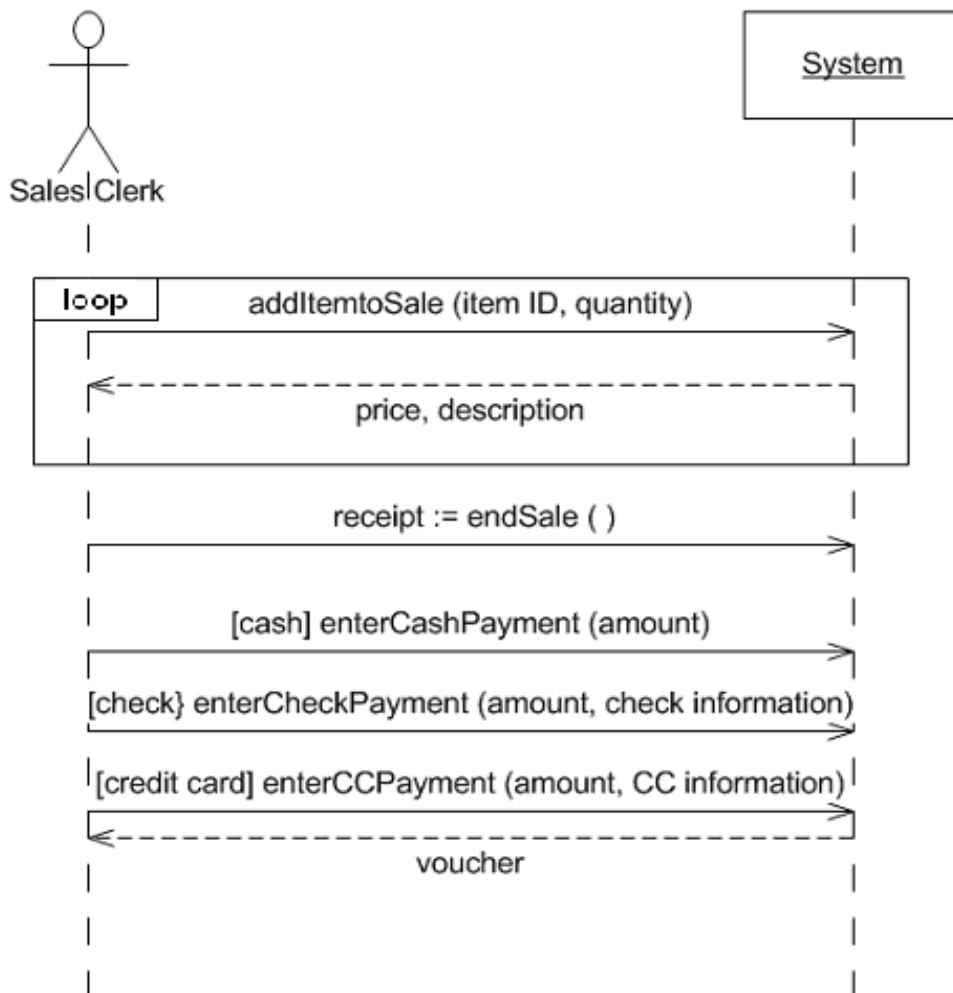
Use Case Name:	Create a new sale	
Scenario:	A new cash sale	
Triggering Event:	Cash customer wants to purchase items.	
Brief Description:	A cash customer wants to purchase items. The clerk enters the item ID, and the system creates a sales ticket. Customer pays with cash, check, or credit card.	
Actors:	Sales clerk	
Stakeholders:	Sales clerk Accounting department Sales department	
Preconditions:	Inventory items must exist.	
Postconditions:	New sale is created. Sales line items are created and connected to the sale. Payment transaction is created.	
Flow of Activities:	Actor	System
	1. Clerk starts new cash sale. 2. Clerk enters each item. 3. Clerk indicates the end of the sale. 4. Clerk indicates type of payment and enters information.	2.1 System finds item in inventory, finds price, displays information, adds to total. 3.1 System calculates total. 4.1 System processes payment and creates payment transaction.
Exception Conditions:	2.1 If system has information missing, sales clerk calls manager and manually enters information. 4.1 If customer credit card fails approval, require cash or cancel sale.	

4. Develop an SSD based on the narrative and your activity diagram for problem 1.

Contractor Sale:



Public sale:



6. Review the cellular telephone state machine diagram shown in Figure 5-21 and then answer the following questions. (Note that this telephone has characteristics not found in ordinary telephones. Base your answers only on the state machine diagram.)

i. What happens to turn on the telephone?

A person has to switch it on with some external event.

ii. What states does the telephone go into when it is turned on?

It goes into the composite state (unlabeled, but should have a label of ON). It goes into the *Quiet* state for one path. It can also go into *Charged*, *Low Warning*, or *Discharged*.

iii. What are the three ways the telephone can be turned off?

SwitchOff, *Quiet* and *Low Warning*, or *Discharged*.

iv. Can the telephone turn off in the middle of the Active (Talking) state?

Yes, if it is discharged, it leaves the *Discharged* nested state and the *On* composite state.

v. How can the telephone get to the Active (Talking) state?

Through the origin state of connecting or by answering from the origin state of ringing.

vi. Can the telephone be plugged in while someone is talking?

No. The state chart says it can only be plugged in from the *Quiet* state.

vii. Can the telephone change battery states while someone is talking? Explain which movement is allowed and which isn't allowed.

It can go from *Charged* to *Low Warning* and from *Low Warning* to *Discharged*. However, because it cannot be plugged in while someone is talking, it cannot move back up from *Discharged* to *Low Warning* or *Charged*.

viii. What states are concurrent with what other states? Make a two-column table showing the concurrent states.

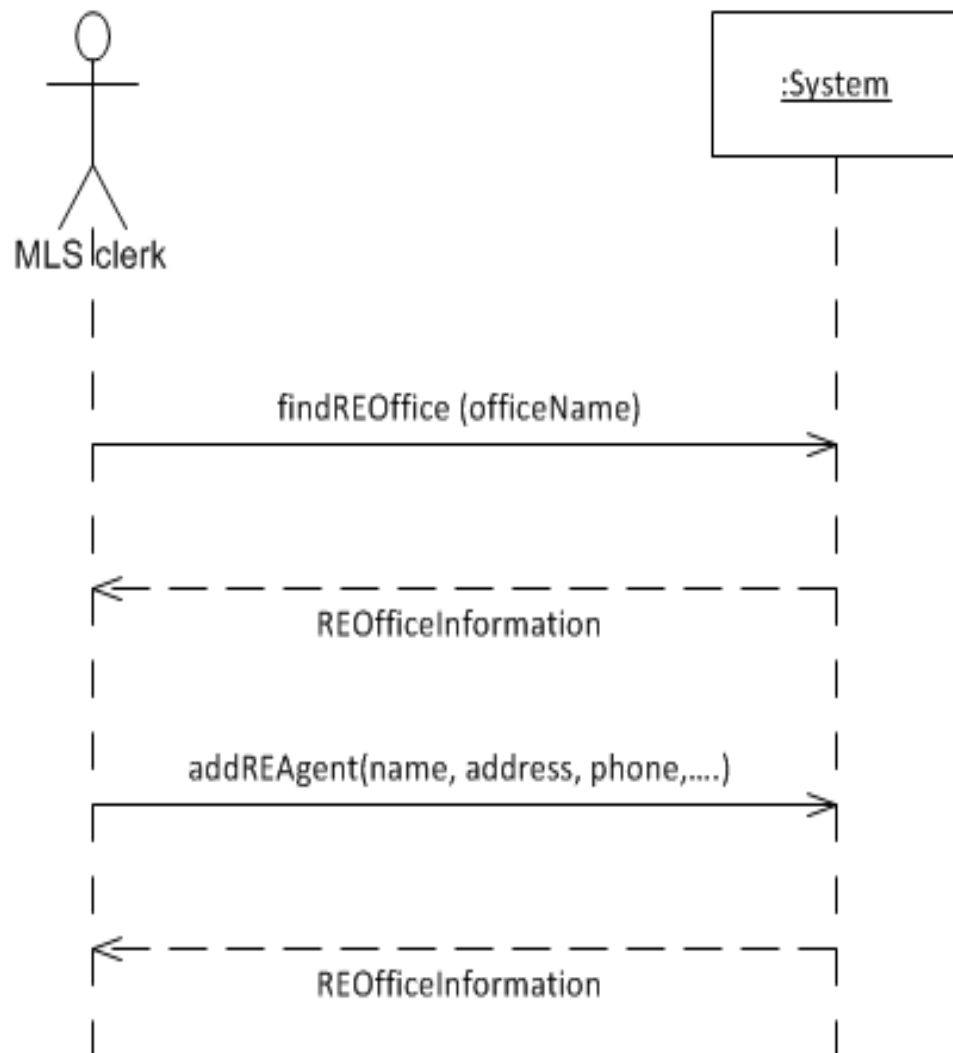
Concurrent States	
Quiet, Dialing, Connecting, Ringing, Active, and Plugged In	Charged, Low Warning, Discharged

Running Case Studies: Community Board of Realtors (Chapter 5)

The Multiple Listing Service system has a number of use cases, which you identified in Chapter 3, and three key domain classes, which you identified in Chapter 4: RealEstateOffice, Agent, and Listing.

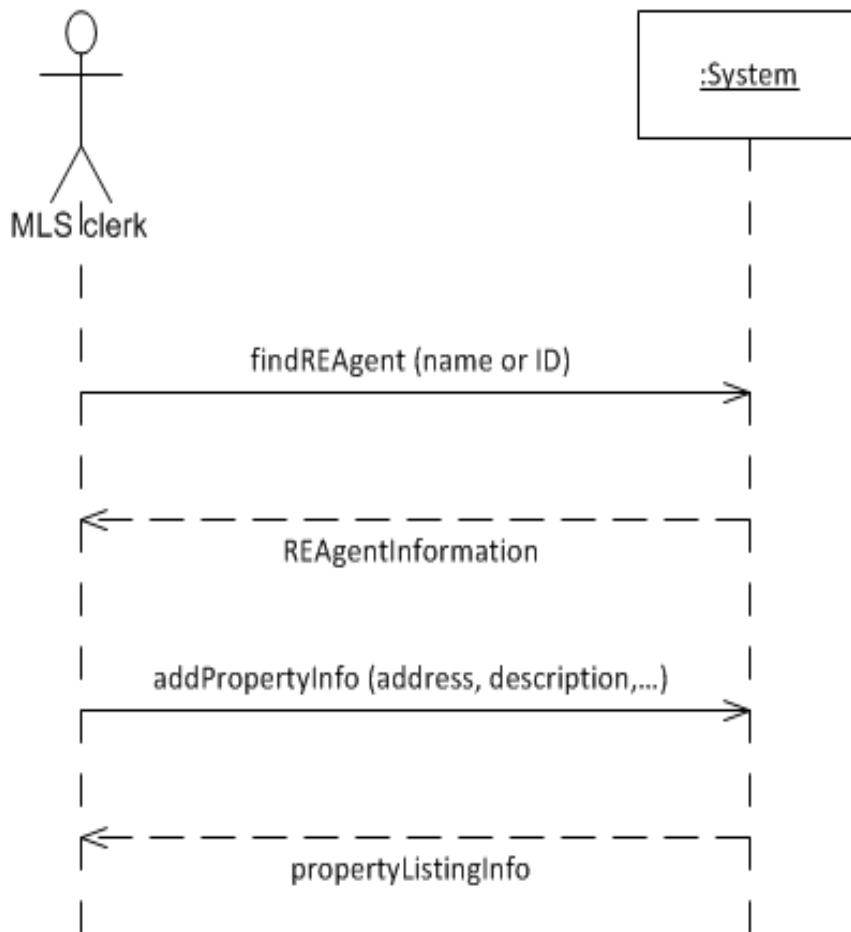
1. For the use case *Add agent to real estate office*, write a fully developed use case description, and draw an SSD. Review the case materials in previous chapters and recall that the system will need to know which real estate office the agent works for before prompting for agent information.

Use case name:	Add agent to real estate office	
Scenario:	MLS clerk adding agent	
Triggering event:	New agent hired in a real estate office	
Brief description:	The correct real estate office is identified, and the new real estate agent information is entered into the system.	
Actors:	MLS clerk	
Related use cases:	Real estate office adds new agent (Web based version scenario)	
Stakeholders:	Real estate office, real estate agent	
Preconditions:	The real estate office must exist	
Postconditions:	Real estate agent is created and associated with real estate office	
Flow of activities	Actor	System
	1. Find correct real estate office 2. Enter new agent information	1.1 Display real estate office information 2.1 Create new agent record, including with relationship to real estate office
Exception conditions:	1.1 No real estate office found for requested id/name. Display not found message.	



2. For the use case *Create new listing*, write a fully developed use case description, and draw an SSD. Recall that the system needs to know which agent made the listing before the system prompts for listing information.

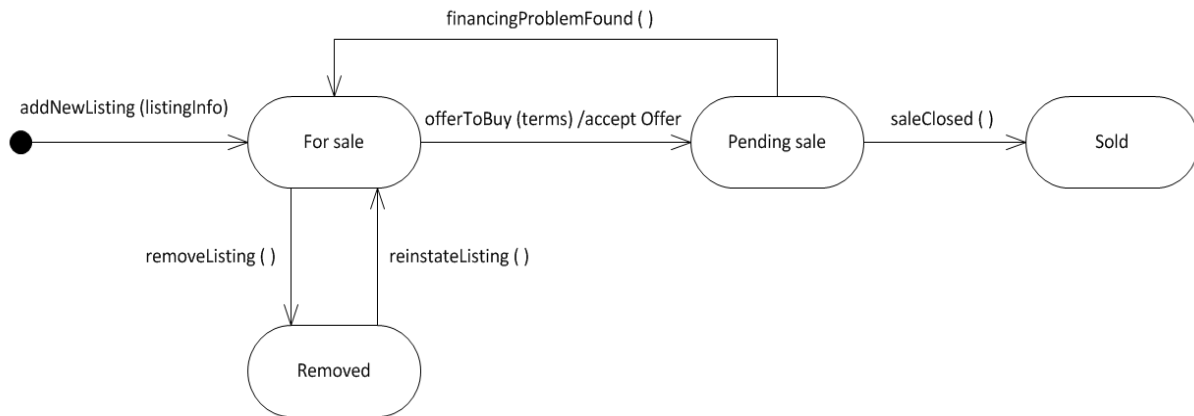
Use case name:	Create a new listing	
Scenario:	MLS clerk creates a new listing	
Triggering event:	New property is put up for sale	
Brief description:	The listing agent is identified and verified. The new property information is entered into the system, along with images etc.	
Actors:	MLS clerk	
Related use cases:	Real estate office/agent creates a new listing (Web version scenario)	
Stakeholders:	Real estate office, Real estate agent, Property owner	
Preconditions:	Real estate office must exist Real estate agent must exist	
Postconditions:	New listing must be created and associated with RE office and RE agent	
Flow of activities	Actor	System
	1. Find real estate agent 2. Enter new listing information	1.1 Display agent and office information 2. Create new property listing record, associated with agent. Display results.
Exception conditions:	1.1 Agent information not found. Display not found message.	



3. Draw a state machine diagram showing the states and transitions for a Listing object.

Note: Answers will vary because students will have to brainstorm valid states for a Listing object. In our solution we will use the following states and exit transitions. (The term “Sale” is used to mean either “Sale” or “Lease”)

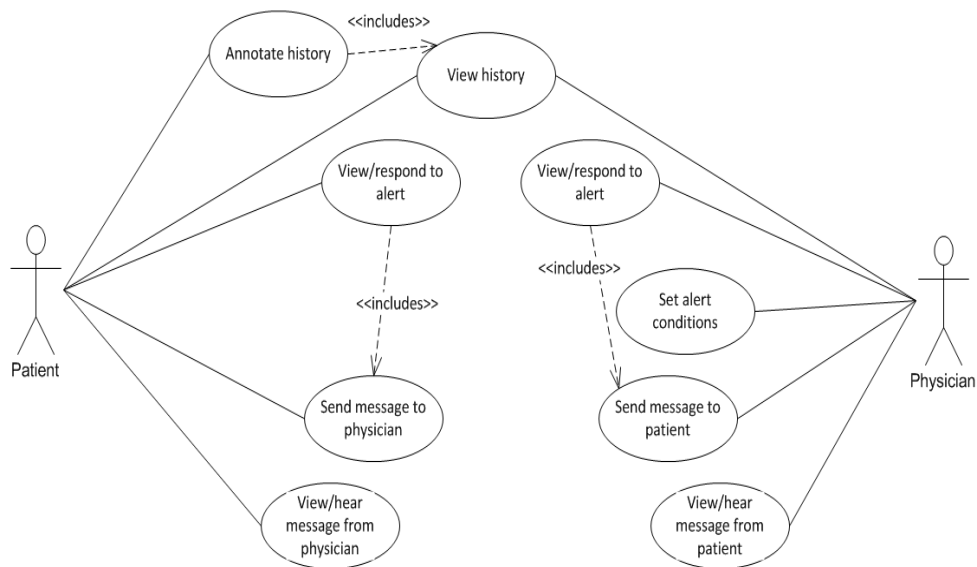
State	Exit transition
For sale	Offer to buy
Sale pending	Close the sale
Sold	
Removed	Reinstate listing



Running Case Studies: Sandia Medical Real-time Glucose Monitoring (Chapter 5)

Figure 5-22 shows a set of use cases for the patient and physician actors. Answer the following questions and/or complete the following exercises:

1. Which use cases include which other use cases? Modify the diagram to incorporate included relationships.

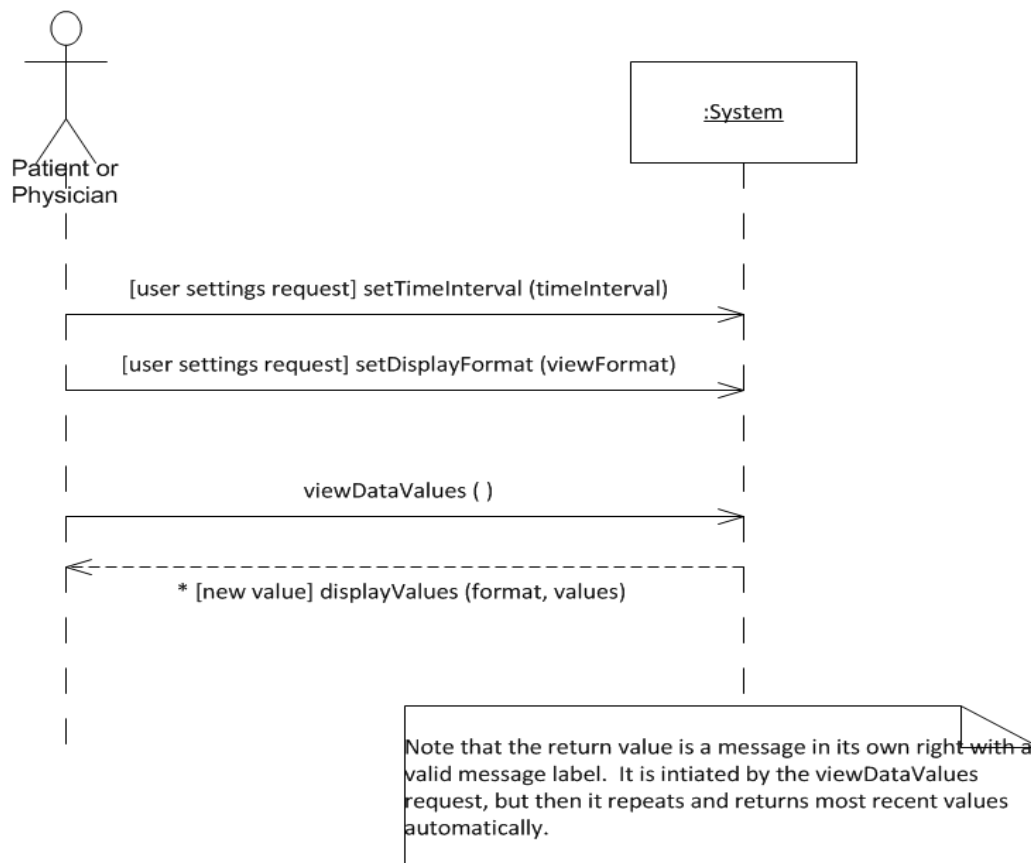


2. Consider the use cases *View/respond to alert* and *View history*. Both actors share the latter, but each has a different version of the former. Why do the actors have different versions of the *View/respond to alert* use case? Would the diagram be incorrect if each actor had his own version of the *View history* use case? Why or why not?

View/respond to alert: Even though the names are the same, the detail steps might be very different. It would be possible to make them different scenarios of the same use case. But making them separate use cases also works since the actors are different, the steps are different, and the system responses are different.

View history: View history does the same activities no matter who the actor is. In this case it would not make sense to have separate use cases since the processing steps, the data, and the activities are both the same.

3. Develop an SSD for the *View history* use case. Assume that the system will automatically display the most recent glucose level, which is updated at 5-minute intervals by default. Assume further that the user can ask the system to view glucose levels during a user-specified time period and that the levels can be displayed in tabular form or as a graph.



Unit 4: Essentials of Systems Design

The previous unit described the activities and decisions associated with discovering and understanding the major elements of the user's requirements—in other words, the *analysis activities*. This unit 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).

This unit gives an introduction in system design activities. It briefly discusses all design activities and goes into more detail with respect to user and system interfaces.

Readings/Activities



Please read the following sections in Chapter 6 of the textbook:

- Technology Decisions at Wysotronics, Inc. (page 154)
- Overview (page 155)

Section 1: Essentials of Design and the Design Activities

In this section, the concept of system design is introduced, and a general overview of all system design activities is provided. Furthermore, this section goes into more detail for the first design activity, namely designing the environment, including topics such as design for internal deployment, design for external deployment, and design for remote, distributed environments. Next, the section takes a look at the RMO environment.

Learning Objectives

When you have completed this section, you should be able to do the following:



1. Describe the difference between systems analysis and systems design.
2. Explain each major design activity.
3. Describe the major hardware and network environment options, and the various hosting services available.

Learning Objective 1

Describe the difference between *systems analysis* and *systems design*.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 6 of the textbook:

- The Elements of Design (pages 155–159)
- Inputs and Outputs for Systems Design (pages 159–160)



Review: Please check your learning with Review Questions 1–8 on page 181 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 2

Explain each major design activity.

Objective Readings/Activities



Readings: Please read the following section in Chapter 6 of the textbook:

- Design Activities (pages 160–167)



Review: Please check your learning with Review Questions 9 and 10 on page 181 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 3

Describe the major hardware and network environment options, and the various hosting services available.

Objective Readings/Activities



Readings: Please read the following section in Chapter 6 of the textbook:

- The Elements of Design (pages 167–180)



Review: Please check your learning with Review Questions 11–22 on page 181 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Section 2: Designing the User and System Interfaces

This section discusses user and system interfaces. The interface between the system and its environment is the zone that permits the exchange of data. Interfaces may be placed into two large categories: *system* and *user*. The chief objective of user-centered design is to create a usable system. In pursuit of this goal, we explore the interaction between the user and the computer, often referred to as *human–computer interaction*, or HCI. HCIs revolve around use cases with inputs and outputs. They form the kernel of user-interface (or *dialog*) design.

System interfaces involve inputs and outputs that require minimal human intervention. In other words, information exchange occurs between automated system components. User interfaces involve inputs and outputs that more directly involve a system user. A user interface enables a user to interact with the computer. On the other hand, system interfaces process standard outputs such as billing notices, reports, printed forms, and electronic outputs to other automated systems. Inputs that are automated or come from nonuser interface devices are also included, for example, inputs from automated scanners, bar-code readers, optical character recognition devices, and other computer systems. Identifying system interfaces can become challenging, as they are not always obvious.

Learning Objectives

When you have completed this section, you should be able to do the following:

1. Describe the difference between user interfaces and system interfaces.
2. Describe the historical development of the field of human–computer interaction (HCI).
3. Discuss how visibility and affordance affect usability.
4. Create storyboards to show the sequence of forms used in a dialog.



5. Describe user interface guidelines that apply to all types of user-interface types and additional guidelines specific to web pages and mobile applications.
6. Discuss examples of system interfaces found in information systems.
7. Discuss the identification and design of system inputs.
8. Discuss and design system outputs.

Learning Objective 1

Describe the difference between user interfaces and system interfaces.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 7 of the textbook:

- Interface Design at Aviation Electronics (page 188)
- Overview (pages 188–189)
- User and System Interfaces (page 189)



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 2

Describe the historical development of the field of human–computer interaction (HCI).

Objective Readings/Activities



Readings: Please read the following section in Chapter 7 of the textbook:

- Understanding the User Interface (pages 189–193)



Review: Please check your learning with Review Questions 1 - 6 on page 219 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 3

Discuss how visibility and affordance affect usability.

Objective Readings/Activities



Readings: Please read the following section in Chapter 7 of the textbook:

- User-Interface Design Concepts (page 193–196)



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 4

Create storyboards to show the sequence of forms used in a dialog.

Objective Readings/Activities



Readings: Please read the following section in Chapter 7 of the textbook:

- The Transition from Analysis to User-Interface Design (pages 196–200)



Review: Please check your learning with Review Questions 8 and 9 on page 219 of the textbook.

Learning Objective 5

Describe user-interface guidelines that apply to all user-interface types and additional guidelines specific to web pages and mobile applications.

Objective Readings/Activities



Readings: Please read the following section in Chapter 7 of the textbook:

- User-Interface Design (pages 200–208)



Review: Please check your learning with Review Question 7 on page 219 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 6

Discuss examples of system interfaces found in information systems.

Objective Readings/Activities



Readings: Please read the following section in Chapter 7 of the textbook:

- Identifying System Interfaces (pages 208–210)



Review: Please check your learning with Review Question 13 on page 219 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 7

Discuss the identification and design of system inputs.

Objective Readings/Activities



Readings: Please read the following section in Chapter 7 of the textbook:

- Designing System Inputs (pages 210–211)



Review: Please check your learning with Review Questions 10–12 on page 219 of the textbook.

Learning Objective 8

Discuss and design system outputs.

Objective Readings/Activities



Readings: Please read the following section in Chapter 7 of the textbook:

- Understanding the User Interface (pages 211–218)



Review: Please check your learning with Review Questions 14–17 on page 219 of the textbook.

Unit 4 – Summary and Unit Activities



Readings:

- Read the Chapter 6 Summary on pages 180–181 of the textbook.
- Read the Chapter 7 Summary on page 218 of the textbook.



Review:

- Review the Key Terms on page 181 and 219, and check if you know what they mean
- Answer questions 1 and 3 from Problems and Exercises in Chapter 6 on page 182 of the textbook.
([Suggested solutions are available here.](#))
- Read “Community Board of Realtors” in Chapter 6, and answer the questions asked on page 183 of the textbook.
([Suggested solutions are available here.](#))
- Read “Sandia Medical Devices” in Chapter 6 and answer questions 1–5 on pages 184–185 of the textbook.
([Suggested solutions are available here.](#))
- Answer questions 2 and 3 from Problems and Exercises in Chapter 7 on pages 219 of the textbook.
([Suggested solutions are available here.](#))
- Read “Community Board of Realtors” in Chapter 7, and answer the questions asked on page 221 of the textbook.
([Suggested solutions are available here.](#))

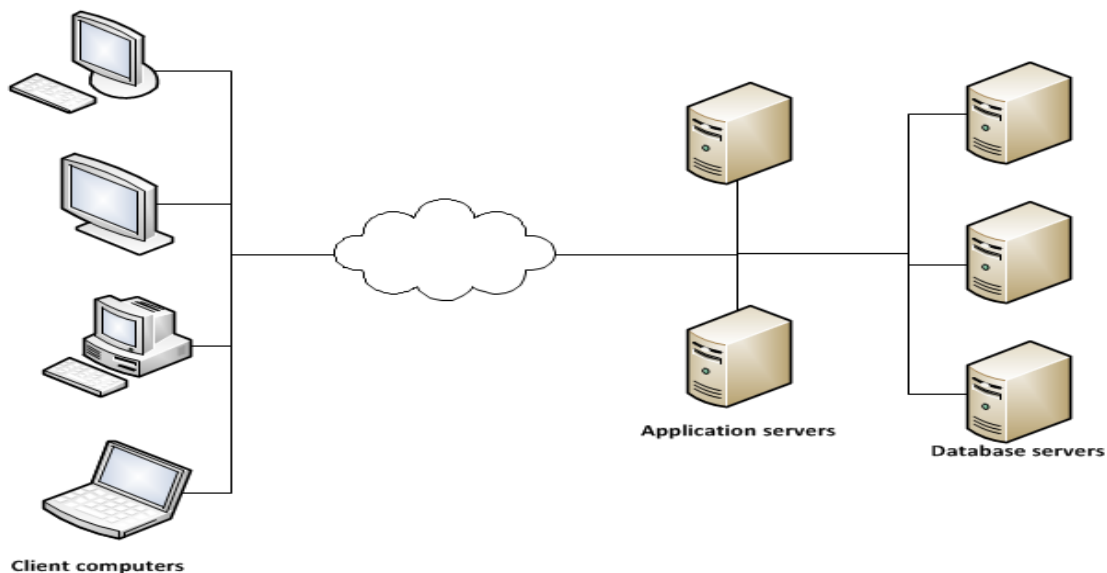
Unit 4: Essentials of Systems Design

Answers for Unit Activities

Problems and Exercises (Chapter 6)

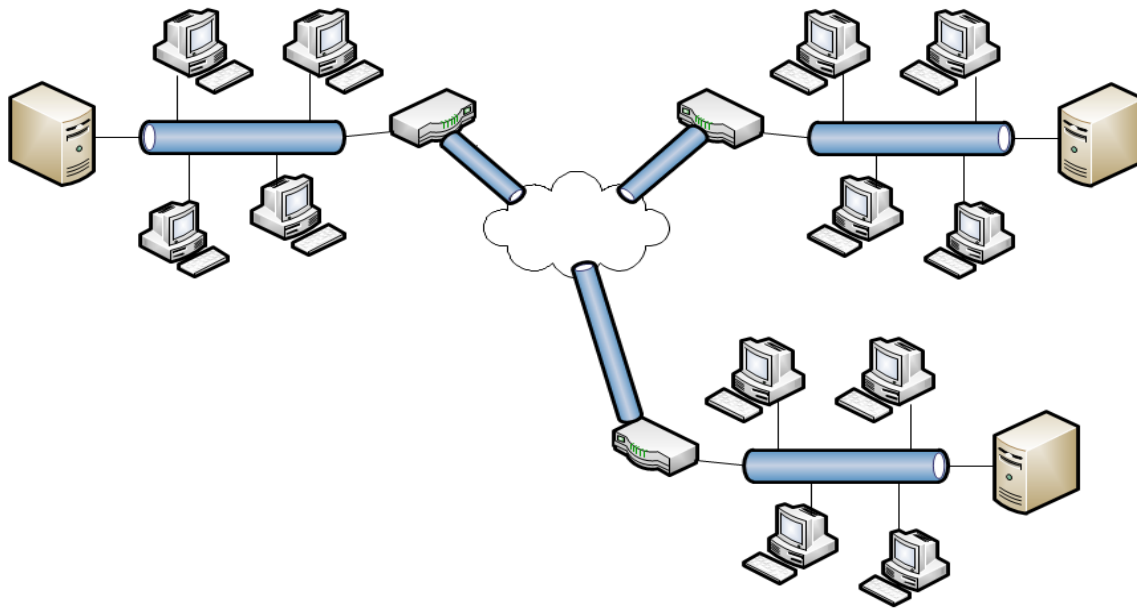
1. A financial corporation has desktop applications running in several different offices that are all supported by a centralized application bank of two computers. In addition, there is a centralized database, which requires three servers. Draw a network diagram representing this requirement.

Note: The problem definition of “several different offices” implies a WAN requirement or an Internet requirement. We will represent the WAN or Internet by the cloud. Various types of client devices are illustrated to indicate that there are multiple types of devices that must be supported.



3. A medium-sized engineering firm has three separate engineering offices. In each office, a local LAN supports all the engineers in that office. Due to the requirement for collaboration among the offices, all the computers should be able to view and update the data from any of the three offices. In other words, the data storage server within each LAN should be accessible to all computers, no matter where they are located. Draw a network diagram that will support this configuration.

Note: We show the LANs connected together with a router, or device that supports the LANs. Then the router devices are connected to the Internet to provide peer-to-peer connectivity, possibly with a VPN.



Running Case Studies: Community Board of Realtors (Chapter 6)

The Community Board of Realtors' Multiple Listing Service (MLS) will be a web-based application with extensions to allow wireless smart phone interaction between the agents and their customers. Review the functional and nonfunctional requirements you have developed for previous chapters. Then, for each of the six design activities discussed in this chapter, list some specific tasks to design the environment, application architecture and software, user interfaces, system interfaces, database, and system controls and security. You may want to refer back to the Tradeshow System discussed in Chapter 1 for some design specifics.

Design the environment: The environment is basically a client-server architecture. The application software will run under a web server, and the database will be directly supported. Design tasks will consist of defining the server and hosting requirements.

Design application architecture and software: As mentioned, the application is a three-layer design consisting of the database, the application logic, and the user interface. Design will follow the normal design steps of specifying the code structure and the methods. Since the

client will consist of browser-based display on many different types of devices, the code will have to discern what to send, depending on the type of client device. Tasks will include creating use case descriptions, sequence diagrams, and other application models.

Design the user interface: The user interface will be one of the more difficult portions of this system. Listing information can include text, images, and even possible videos. To be able to have meaningful display of all three types on desktops, laptops, and mobile devices, some care will need to be given in designing the user interface. Tasks will include design sessions with some users for each type of device. Perhaps even some trial prototypes will need to be built to test the effectiveness of different screen layouts.

Design system interfaces: It does not appear that there are extensive system interfaces for this application other than the printed reports. Report design should also involve user input on the best way to print multiple listing books and brochures.

Design the database: Design of the database will require defining the various indexes and searching options. Other tasks include defining the attribute characteristics and foreign keys.

Design system controls and security: The primary concern with security in this system is to protect the listing data as it resides on the server. The server, the database, and the applications must all be protected against hacking and defacing. Output data is created for the general public so it is not private or confidential at that point. Design tasks will be to integrate secure data input, update, and protection.

Running Case Studies: Sandia Medical Devices (Chapter 6)

Answer these questions in light of HIPPA requirements:

1. Does HIPAA apply to the RTGM system? Why or why not?

It would appear that HIPAA regulations do apply to the RTGM system. The system maintains personal health-related information about the patient; hence it must comply. It also transmits detailed monitoring results over public links and therefore should encrypt the data that is being sent.

2. How should the system ensure data security during transmission between a patient's mobile device(s) and servers?

Since the data is being sent over telephone links, it is only as secure as cellular telephone signals. Someone could possibly receive those transmissions by finding the correct

frequency. Therefore, the data itself should be encrypted before being sent. Hence Sandia will need to provide, and patients will need to install, encrypted SMS to receive their text message alerts. The glucose monitoring results should also be sent securely and encrypted.

- 3. Consider the data storage issues related to a patient's mobile device and the possible ramifications if the device is lost or stolen. What measures should be taken to protect the data against unauthorized access?**

This is an interesting problem. Most patients will not want to have to enter a password or a pin every time they need to access their alert messages or to activate the transmission of monitoring data. It is not clear how much responsibility rests on the patient (to keep his/her phone secure) and how much responsibility rests on Sandia (to require a password or pin to activate). The data itself could be encrypted on the smart phone, but of course, the application is able to decrypt it to access it. Perhaps a middle ground might be to require a pin when the smartphone is turned on, or once each day.

- 4. Consider the issues related to health care professionals accessing server data by using workstations and mobile devices within a health care facility. How will the system meet its duty to record and examine access to ePHI? If a health care professional uses a mobile device outside a health care facility, what protections must be applied to the device and/or any data stored within it or transmitted to it?**

Requirements for health care professionals can be more stringent to access secure ePHI data. Each health care person can be required to enter ID and password to access the data within the facility on workstations or laptop computing devices. Mobile devices can also require a login process in order for the professional to access the data. And the data should be encrypted when it is transmitted. Something as simple as a four-digit pin for mobile devices is not too burdensome, yet still provides a level of security.

All access to the data should also be logged so that Sandia knows who has access and who has accessed the data. The system will maintain login information for authorized users. (See Chapter 12 for more details.)

- 5. Consider the issues related to wired and wireless data transmission between servers and workstations within a health care facility. What security duties, if any, apply to transmissions containing ePHI? Does your answer change if the servers are hosted by a third-party provider?**

Security of wired and wireless data transmission and hosted servers requires first that the servers are hosted in a secure environment and second that all data stored and transmitted be encrypted. Encryption of local data is easier because the access program maintains the encryption key. (Hence the program itself must be kept physically secure.) Transmitted

data is more complex because the encryption must be dynamic due to the remote nature of mobile devices and distributed computers.

The solution will need to be the same whether the servers are hosted in house or with a third-party provider. The physical facilities and access to the servers themselves must be in a HIPAA secure facility.

Problems and Exercises (Chapter 7)

- 2. Visit some websites, and then identify all the controls used for navigation and input. Are they all obvious? Discuss some differences in visibility and affordance among the controls.**

Answers will vary. Most sites have good affordance and visibility, but sometimes there will be a file download or an update to a field where it is not obvious that it has completed.

Websites include many clickable objects. It is hard to tell which objects can be clicked and which cannot. Many objects do not indicate that they have been clicked. In addition, it is not always easy to tell what the control does—does it go to another part of the current page, to another page at the site, or to another site altogether?

- 3. A common maxim for designing a man–machine interface is that it is better to change the machine than to try to change the human to accommodate the machine. Are there machines (or systems) that you use in your daily life that have room for improvement? Are the current generations of Windows PC and Apple Mac as usable as they might be? If not, what improvements can you suggest? Is the World Wide Web as usable as it might be? If not, what improvements can you suggest? Are we just beginning to see some breakthroughs in usability, or have most of the big improvements already been made?**

Student solutions will vary (probably wildly).

Some students might want to compare Windows and Mac interfaces with other user interfaces on physical devices. Some devices for comparison might be as follows:

- automobile user interfaces (with GPS, mobile phone, etc.)
- home appliances (microwave and other programmable devices)
- home entertainment systems (TV, DVRs, DVD players, etc.)

Running Case Studies: Community Board of Realtors (Chapter 7)

One of the key use cases for the multiple listing service system is *Create new listing*, where the Realtor enters all the important information about a new listing he or she has obtained. Consider the information that must be entered when creating a new listing, and list the dialog steps that are necessary. Keep in mind that when designing for a smartphone, less information can be entered in each step compared to a full-screen web application. Also keep in mind that typing is error prone and awkward for many users, so think about opportunities to use check boxes, radio buttons, and list boxes to aid selection. Create a storyboard of this use case for a mobile device, showing each step of the dialog that maximizes the use of check boxes, radio buttons, and list boxes.

Answers will vary: Designing smartphone apps is a whole new endeavor and quite different from normal browser-based apps. The following design is based on one screen to act as a menu screen (what to do). The second screen allows entry of all the data. It is on one screen which is a scrolling screen with data entry on entire screen. Again for smartphone apps, there are combinations of screen data entry, keyboard, Save hot link – sometime on the screen, sometimes on the keyboard, and settings pop-up menu option.

Unit 4: Essentials of Systems Design
Answers for Unit Activities

Multiple Listing Service	
Listing Information	Find Listings
	New Listings
Agent Information	Find Agents
	New Agents
Office Information	Find Offices

Multiple Listing Service	
Property Location	
Street	<input type="text"/>
City	<input type="text"/>
State	<input type="text"/>
Postal	<input type="text"/>
Selling Information	
Price	<input type="text"/>
Property Description	
Square Feet	<input type="text"/>
Number of bedrooms	<input type="text"/> ▼
Number of baths	<input type="text"/> ▼
Year built	<input type="text"/> ▼
Seller Information	
Name	<input type="text"/> First <input type="text"/> Last
Street	<input type="text"/>
City	<input type="text"/>
State	<input type="text"/>
Postal	<input type="text"/>
<input type="button" value="Save"/> <input type="button" value="Cancel"/>	

Unit 5: Advanced Design Concepts

This unit focuses on two levels of design—*architectural design*, often referred to as *high-level design*, and *detailed design*, where the design of each use case is specified. The unit first discusses general *object-oriented* design issues and principles. Then it goes into more detail, explaining the use of modelling techniques for use case realization including *design class diagrams*, *CRC cards*, *interaction diagrams*, and *package diagrams*. The unit focuses especially on the design of *multilayer systems*. Furthermore, the unit discusses the use of *design patterns* for detailed object-oriented design. In addition, the design of databases, controls, and security is discussed in detail.

Section 1: Object-Oriented Design: Principles

So what is object-oriented design? It is a process by which a set of detailed object-oriented design models are built and then used by programmers to write and test the new system. System design is the bridge between user requirements and programming the new system.

One strength of the object-oriented approach is that the design models are often just extensions of the requirements models. Obviously, it is much easier to extend an existing model than to create an entirely new one. However, we emphasize that it is good practice to create design models and not just jump into coding. Just as a builder does not build something larger than a doghouse or a shed without a set of blueprints, a system developer would never try to develop a large system without a set of design models.

Students who are building personal web pages or small systems for course assignments sometimes think that design models are unnecessary. Remember, however, that while blueprints may not be necessary for a doghouse, they are certainly necessary for a home. The doghouse-scale design model that you learn to do here will prepare you for the house-scale model later.

In this section, an overview of object-oriented design is given, and the fundamental principles of object-oriented detailed design are introduced. The creation of object-oriented architectural design, design classes, and design class diagrams is discussed. Detailed design with CRC cards is also introduced. The section concludes with a discussion on fundamental detailed design principles.

Learning Objectives

When you have completed this section, you should be able to do the following:



1. Explain the purpose and objectives of object-oriented design.
2. Develop UML component diagrams.
3. Develop design class diagrams.
4. Use CRC cards to define class responsibilities and collaborations.
5. Explain some of the fundamental principles of object-oriented design.

Learning Objective 1

Explain the purpose and objectives of object-oriented design.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 10 of the textbook:

- New Capital Bank: Part 1 (page 294)
- Overview (pages 294–295)
- Object-Oriented Design: Bridging from Analysis to Implementation (pages 294–298)



Review: Please check your learning with Review Questions 1–3 on page 321 of the textbook.

Learning Objective 2

Develop UML component diagrams.

Objective Readings/Activities



Readings: Please read the following section in Chapter 10 of the textbook:

- Object-Oriented Architectural Design (pages 298–304)



Review: Please check your learning with Review Questions 4 and 6–9 on page 321 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 3

Develop design class diagrams.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 10 of the textbook:

- Fundamental Principles of Object-Oriented Detailed Design (pages 304–308)
- Design Classes and the Design Class Diagram (pages 308–314)



Review: Please check your learning with Review Questions 5 and 10–19 on page 321 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 4

Use CRC cards to define class responsibilities and collaborations.

Objective Readings/Activities



Readings: Please read the following section in Chapter 10 of the textbook:

- Detailed Design with CRC Cards (pages 314–317)



Review: Please check your learning with Review Questions 20 and 21 on page 321 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 5

Explain some of the fundamental principles of object-oriented design.

Objective Readings/Activities



Readings: Please read the following section in Chapter 10 of the textbook:

- Fundamental Detailed Design Principles (page 317–320)



Review: Please check your learning with Review Questions 22–24 on page 321 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Section 2: Object-Oriented Design: Use Case Realizations

In the following sections, we pursue object-oriented detailed design in much more depth and formality. Detailed design is a subject that can be addressed at multiple levels. For the beginner, a fairly straightforward, yet complete, process can be defined. It can also be pursued in more depth.

This section provides an introduction into the foundational principles of use case realization, especially with respect to multilayer systems. Detailed design of multilayer systems and the use case realization with sequence diagrams is discussed. The design of communication diagrams and package diagrams is introduced, and the need for updating the design class diagram at this stage in the design phase is highlighted. In addition, the concept of design patterns is introduced.

Learning Objectives

When you have completed this section, you should be able to do the following:



1. Explain the different types of objects and layers in a design.
2. Develop sequence diagrams for use case realization.
3. Develop communication diagrams for detailed design.
4. Develop updated design class diagrams and multilayer

subsystem packages.

5. Explain design patterns, and recognize specific patterns.

Learning Objective 1

Explain the different types of objects and layers in a design.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 11 of the textbook:

- New Capital Bank: Part 2 (page 328)
- Overview (pages 328–329)
- Detailed Design of Multilayer Systems (pages 329–332)



Review: Please check your learning with Review Questions 1–4 on page 361 of the textbook.

Learning Objective 2

Develop sequence diagrams for use case realization.

Objective Readings/Activities



Readings: Please read the following section in Chapter 11 of the textbook:

- Use Case Realization with Sequence Diagrams (pages 332–349)



Review: Please check your learning with Review Questions 5–16 on pages 361–362 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 3

Develop communication diagrams for detailed design.

Objective Readings/Activities



Readings: Please read the following section in Chapter 11 of the textbook:

- Designing with Communication Diagrams (pages 349–350)



Review: Please check your learning with Review Questions 17 and 18 on page 362 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 4

Develop updated design class diagrams and multilayer subsystem packages.

Objective Readings/Activities



Readings: Please read the following section in Chapter 11 of the textbook:

- Updating and Packaging the Design Classes (pages 351–356)



Review: Please check your learning with Review Questions 19–21 on page 362 of the textbook.



Supplementary Readings: Please check the [digital reading room](#) for additional reading materials on the topic of this section.

Learning Objective 5

Explain design patterns, and recognize various specific patterns.

Objective Readings/Activities



Readings: Please read the following section in Chapter 11 of the textbook:

- Design Patterns (pages 356–360)



Review: Please check your learning with Review Questions 22 and 23 on page 362 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Section 3: Databases, Controls, and Security

In this section, the design of databases, controls, and security are discussed.

Databases and database management systems are important components of a modern information system. Databases provide a common repository for data so that it can be shared by the entire organization. Database management systems provide designers, programmers, and end users with sophisticated capabilities to store, retrieve, and manage data. Sharing and managing the vast amounts of data needed by a modern organization simply would not be possible without a database management system.

Information system controls are mechanisms and procedures that are built into a system to safeguard both the system and the information within it. Generally, controls that are integrated into the application and database are called *integrity controls*. The controls in the operating system and network are often referred to as *security controls*.

Learning Objectives

When you have completed this section, you should be able to do the following:

1. Design a relational database schema based on a class diagram.
2. Evaluate and improve the quality of a database schema.
3. Describe the different architectural models for distributed databases.
4. Determine when and how to design the database.
5. Explain the importance of integrity controls for inputs, outputs, data, and processing.
6. Discuss issues related to security that affect the design and operation of information systems.



Learning Objective 1

Design a relational database schema based on a class diagram.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 12 of the textbook:

- Downslope Ski Company: Designing a Secure Supplier System Interface (page 372)
- Overview (page 373)
- Databases and Database Management Systems (pages 373–374)
- Relational Databases (pages 374–383)



Review: Please check your learning with Review Questions 1–10 on page 404 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 2

Evaluate and improve the quality of a database schema.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 12 of the textbook:

- Relational Databases (pages 383–387)
- Data Access Classes (page 387)



Review: Please check your learning with Review Questions 11–13 on page 404 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 3

Describe the different architectural models for distributed databases

Objective Readings/Activities



Readings: Please read the following sections in Chapter 12 of the textbook:

- Distributed Database Architectures (pages 387–391)



Review: Please check your learning with Review Questions 14 and 15 on page 404 of the textbook.

Learning Objective 4

Determine when and how to design the database.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 12 of the textbook:

- Database Design Timing and Risks (pages 391–392)



Review: Please check your learning with Review Questions 16 and 17 on page 404 of the textbook.

Learning Objective 5

Explain the importance of integrity controls for inputs, outputs, data, and processing.

Objective Readings/Activities



Readings: Please read the following section in Chapter 12 of the textbook:

- Designing Integrity Controls (page 392–396)



Review: Please check your learning with Review Questions 18–21 on page 404 of the textbook.

Learning Objective 6

Discuss issues related to security that affect the design and operation of information systems.

Objective Readings/Activities



Readings: Please read the following sections in Chapter 12 of the textbook:

- Designing Security Controls (pages 396–402)



Review: Please check Please your learning with Review Questions 22–27 on page 404 of the textbook.



Supplementary Readings: check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Unit 5 – Summary and Unit Activities



Readings: Please read the following sections of the textbook:

- Chapter 10 Summary (page 320)
- Chapter 11 Summary (page 361)
- Chapter 12 Summary (page 403)



Review:

- Review the Key Terms on pages 321, 361, 403 and 404, and check if you know what they mean.
- Answer questions 1 and 3 from Problems and Exercises in Chapter 10 on page 322 of the textbook.
([Suggested solutions are available here.](#))
- Read “Community Board of Realtors” in Chapter 10, and answer questions 1–3 on page 323 of the textbook.
([Suggested solutions are available here.](#))
- Read “Sandia Medical Devices” in Chapter 10, and answer questions 1 and 2 on page 324 of the textbook.
([Suggested solutions are available here.](#))
- Read “Community Board of Realtors” in Chapter 11, and answer question 1–6

on page 367 of the textbook.

([Suggested solutions are available here.](#))

- Read “Sandia Medical Devices” in Chapter 11, and answer questions 1–4 on page 368 of the textbook.
([Suggested solutions are available here.](#))
- Answer questions 2, 3, and 5 from Problems and Exercises in Chapter 12 on pages 405 of the textbook.
([Suggested solutions are available here.](#))
- Read “Community Board of Realtors” in Chapter 12, and answer the question there on page 406 of the textbook.
([Suggested solutions are available here.](#))
- Read “Sandia Medical Systems Real-time Glucose Monitoring” in Chapter 12, and answer the questions in part 1 and 2 on pages 407–408 of the textbook.
([Suggested solutions are available here.](#))

Unit 5: Advanced Design Concepts

Answers for Unit Activities

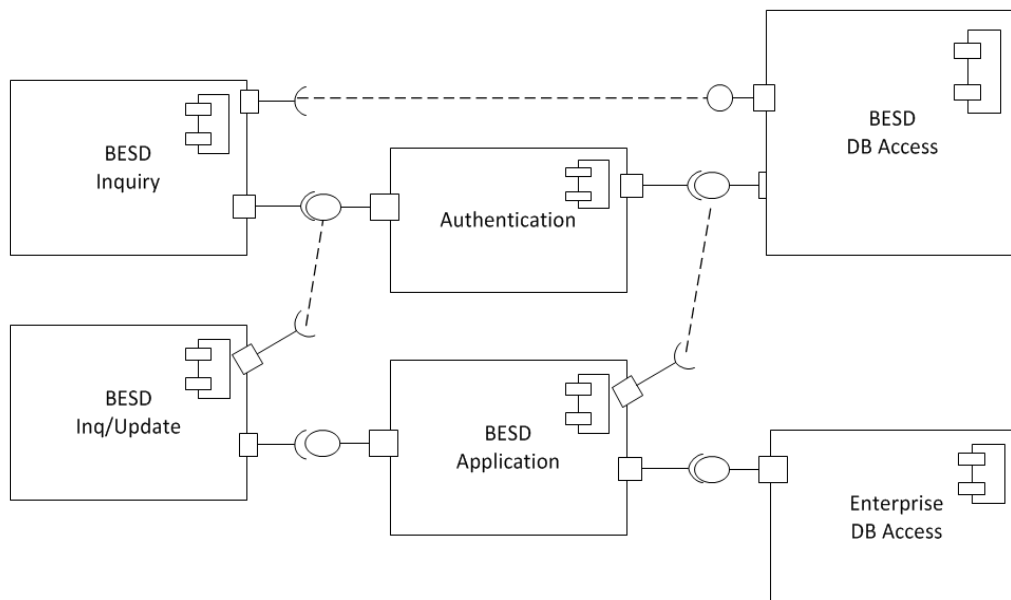
Problems and Exercises (Chapter 10)

1. Given the following system description, develop a component diagram for a desktop-operated internal network system (i.e., Internet access is not required).

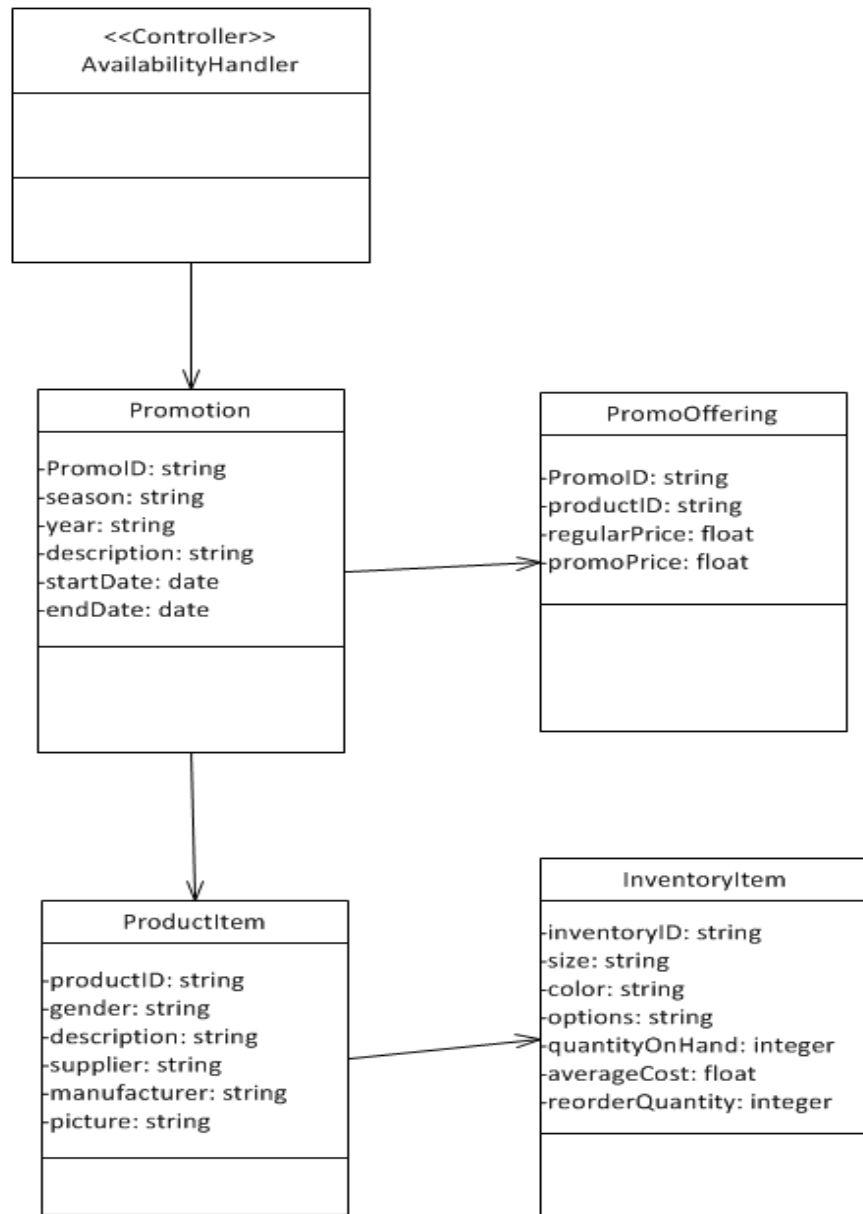
The new Benefits for Employees, Spouses, and Dependents (BESD) system will be used primarily by the human resources department and will contain confidential information. Consequently, it will be built as a totally in-house system, without any Internet elements. The database for the system is the human resource employee database (HRED), which is shared by several other systems within the company.

There are two types of screens from a systems design viewpoint: simple inquiry screens and complex inquiry/update screens. The simple inquiry screens just access the database, with no business logic required. The complex screens usually do fairly complicated calculations based on sophisticated business rules. These programs often have to access other data tables from other databases in the company.

The database will always remain on a central database server. The application program itself will be installed on each desktop that is allowed access. However, authentication is a centralized process, and it will control which screens and program functions can be accessed by which users.



3. In this chapter, we developed a first-cut DCD, a set of CRC cards, and a final DCD for the *Create phone sale* use case for RMO. Create the same three drawings for the *Look up item availability* use case.



Unit 5: Advanced Design Concepts
Answers for Unit Activities

<i>InquireOnItemWindow</i>	
<i>accept item data</i> <i>display items</i>	<i>AvailabilityHandler</i>

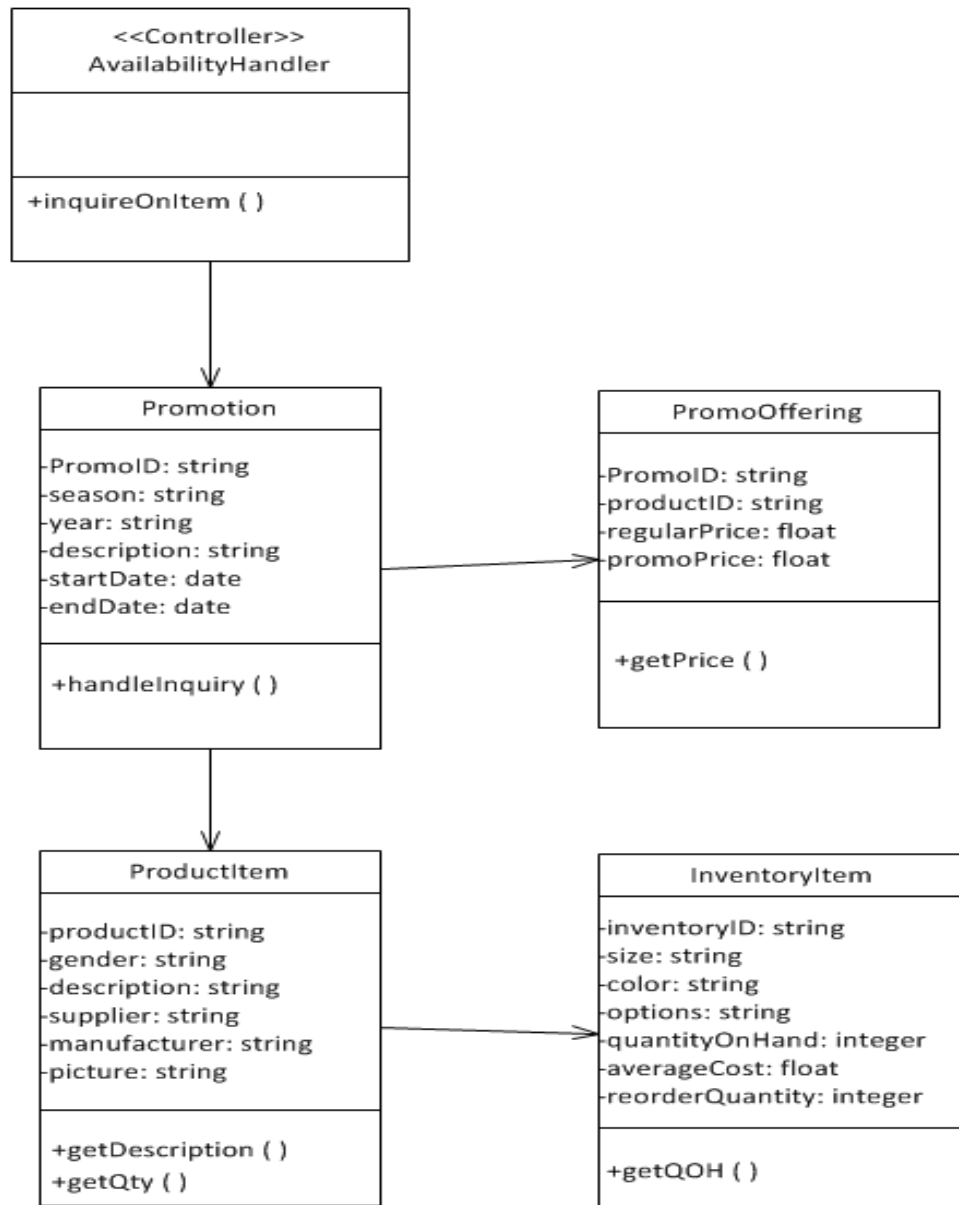
<i>AvailabilityHandler</i>	
<i>inquireOnItem</i>	<i>Promotion</i>

<i>Pomotion</i>	
<i>handle inquiry</i>	<i>ProductItem</i> <i>PromoOffering</i>

<i>PromoOffering</i>	
<i>get price</i>	

<i>Product Item</i>	
<i>get description</i> <i>get quantity</i>	<i>InventoryItem</i>

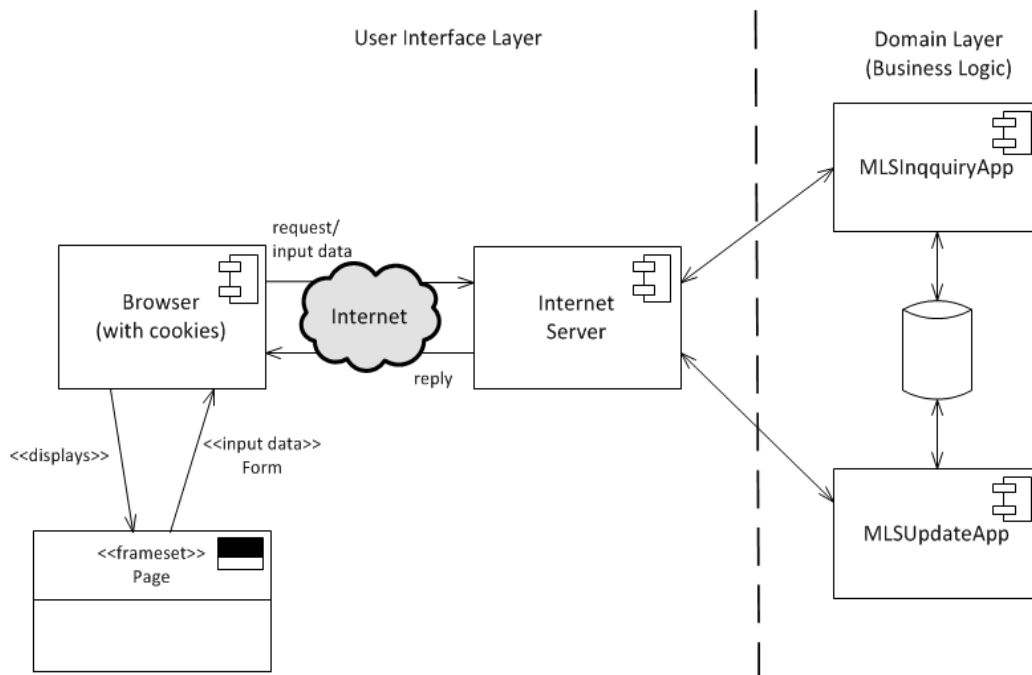
<i>InventoryItem</i>	
<i>get QOH</i>	



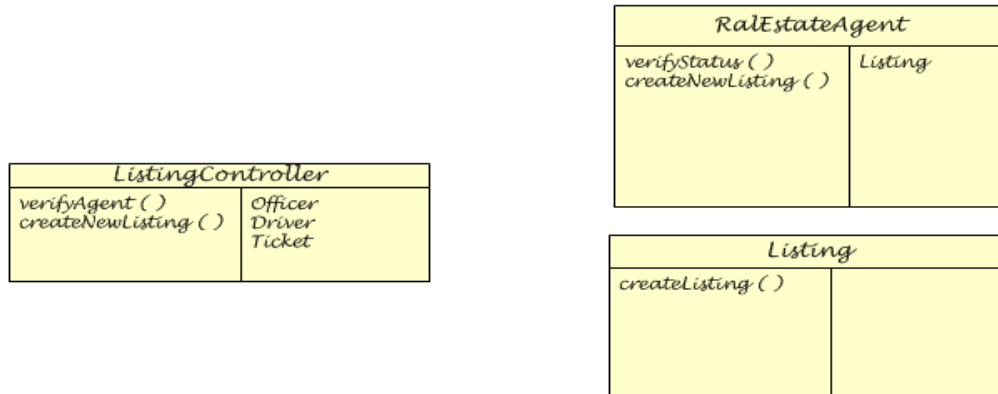
Running Case Studies: Community Board of Realtors (Chapter 10)

In Chapter 3 and Chapter 5, you identified and then modeled use cases for the Multiple Listing Service (MLS) application. You also identified and modeled domain classes. Use your solutions from these chapters to do the following:

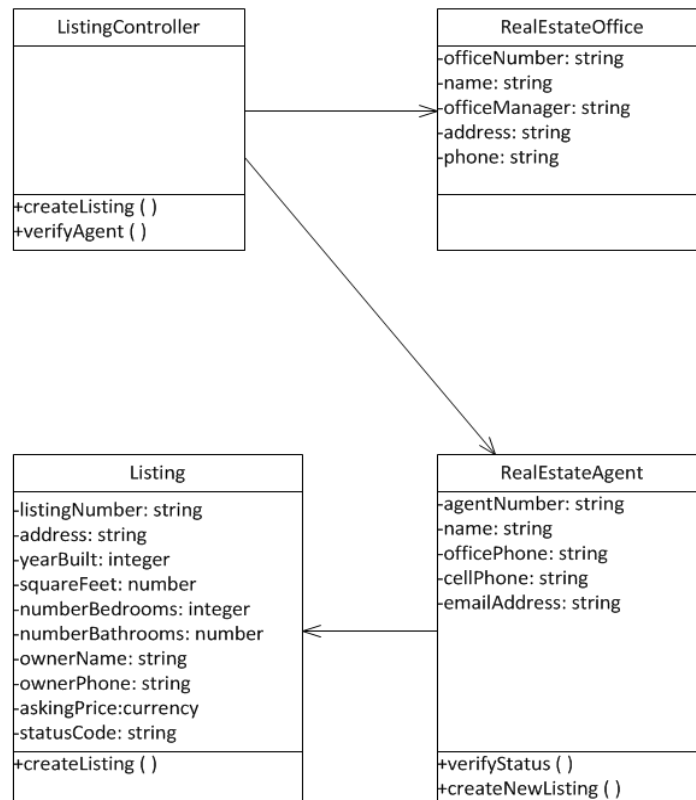
1. Draw a basic component diagram showing the architectural design for the system, assuming that it is a two-layer Internet architecture.



2. Use the CRC cards technique to identify the classes that are involved in the *Create new listing* use case. Recall that creating a new listing involves an agent, a real estate office, and a listing. Decide which class should have the primary responsibility for collaborating with the other classes, and then complete the CRC cards for the use case.



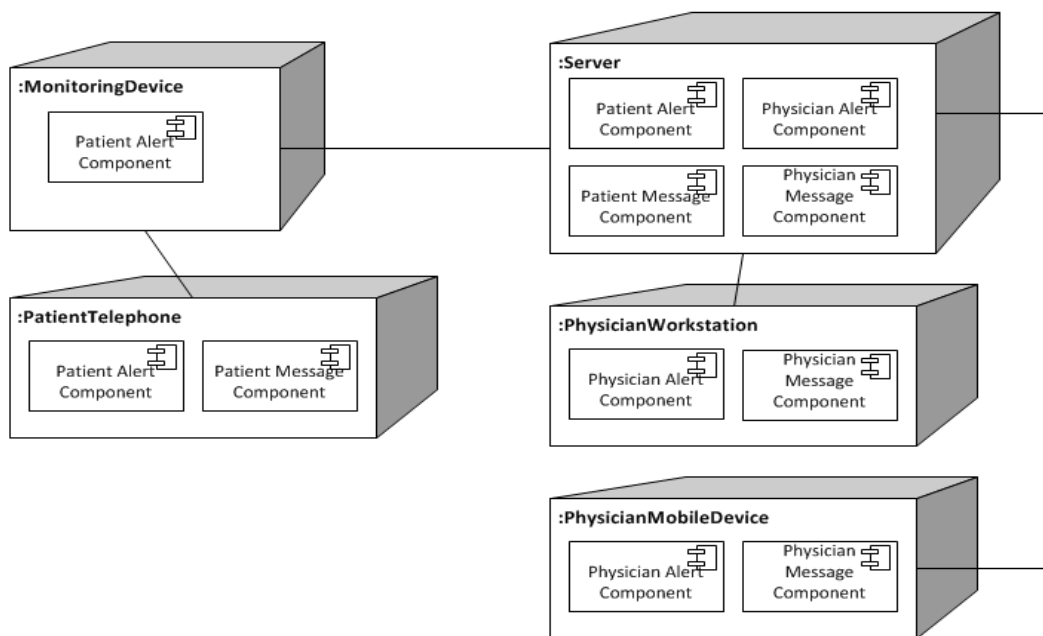
3. Draw a first-cut design class diagram (DCD) based on the CRC cards for this use case.



Running Case Studies: Sandia Medical Devices (Chapter 10)

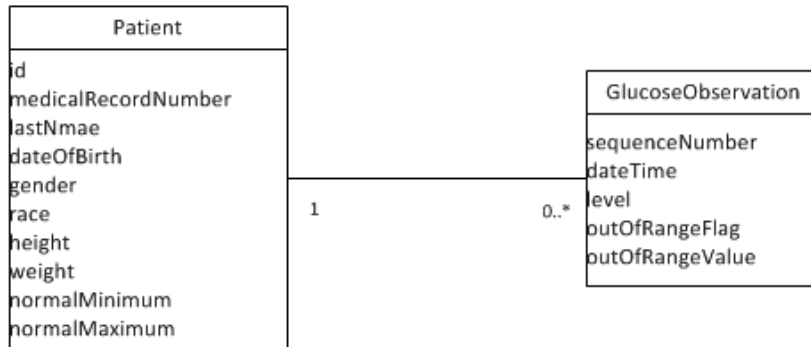
Review the original system description in Chapter 2, additional project information in Chapters 3, 4, 6, 8, and 9, and the use case diagram shown in [Figure 10-20](#) to refamiliarize yourself with the proposed system. Complete these tasks:

1. Develop a deployment diagram that shows the equipment specified in Chapter 6 and the list of software components you developed while answering question 1 in Chapter 9.

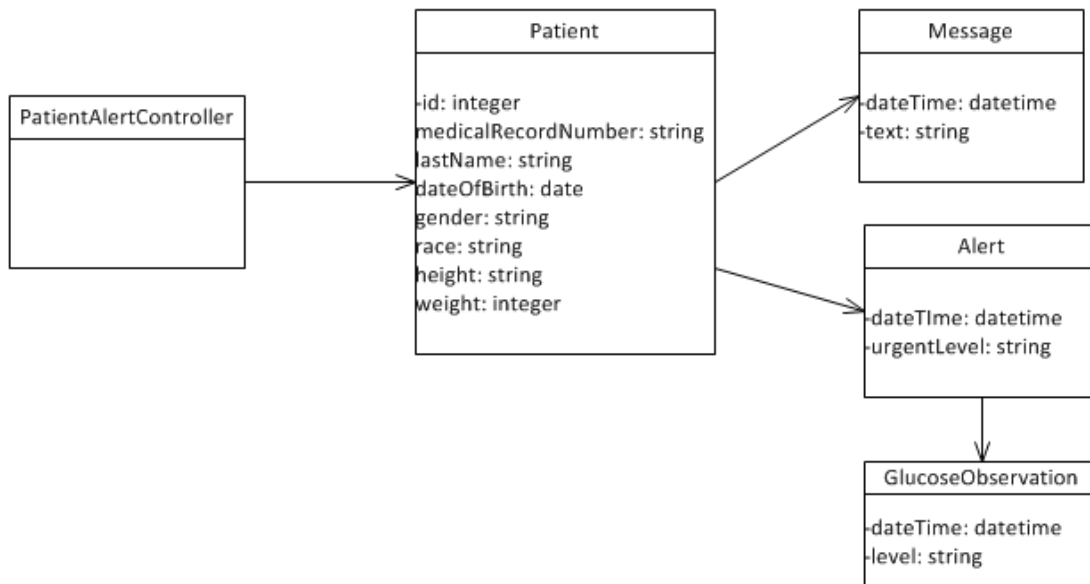


2. For the moment, assume that the database will store two glucose levels for each patient—normal minimum and normal maximum—and that an alert will be generated if three or more consecutive glucose readings are above or below those levels. Expand the domain class diagram in Chapter 4 to include this information, and then develop a first-cut design class diagram to support the patient use case *View/respond to alert*.

Additional fields required in Domain model



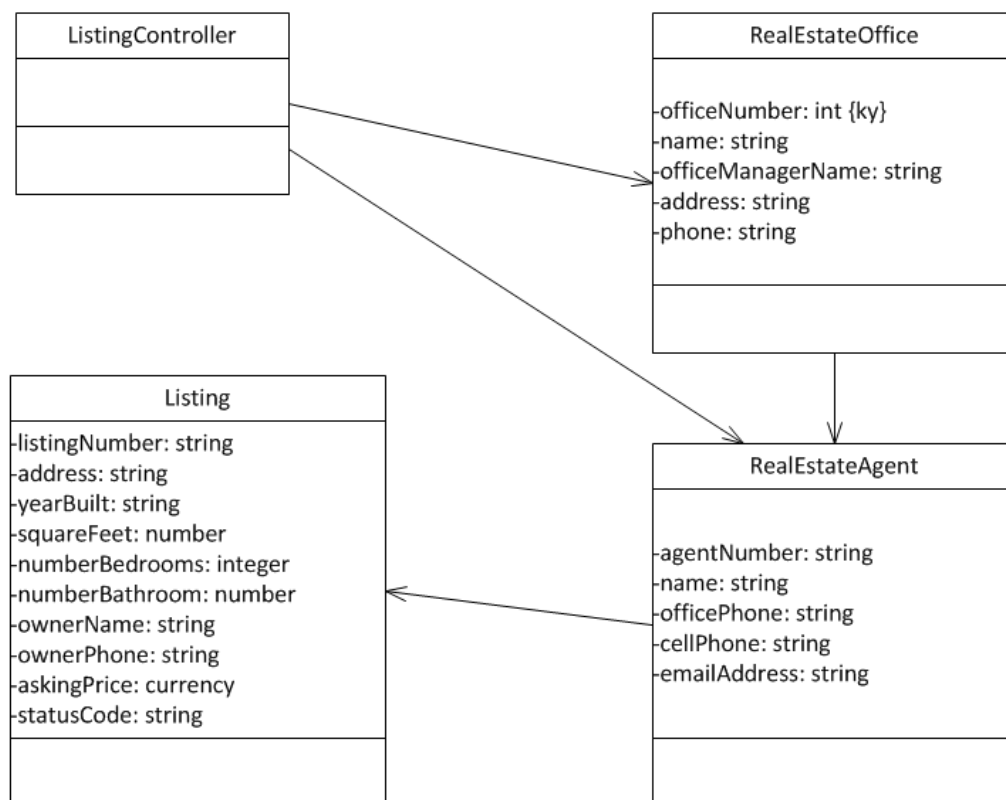
DCD for *View/Respond to alert*



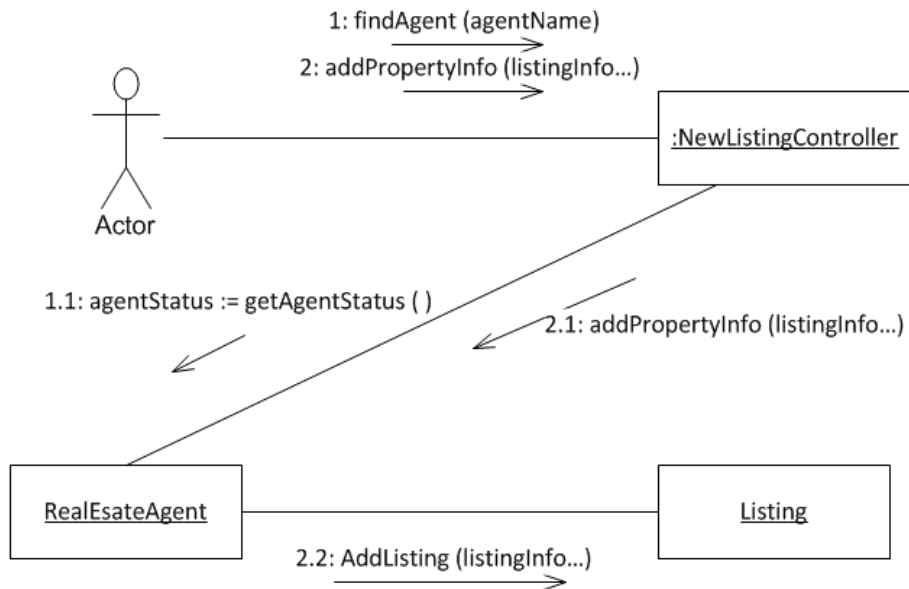
Running Case Studies: Community Board of Realtors (Chapter 11)

In Chapter 3, you identified use cases for the business events for the Community Board of Realtors. In Chapter 5, you elaborated on those use cases. In Chapter 4, you identified the classes associated with the business events. Using your solutions from those chapters, develop:

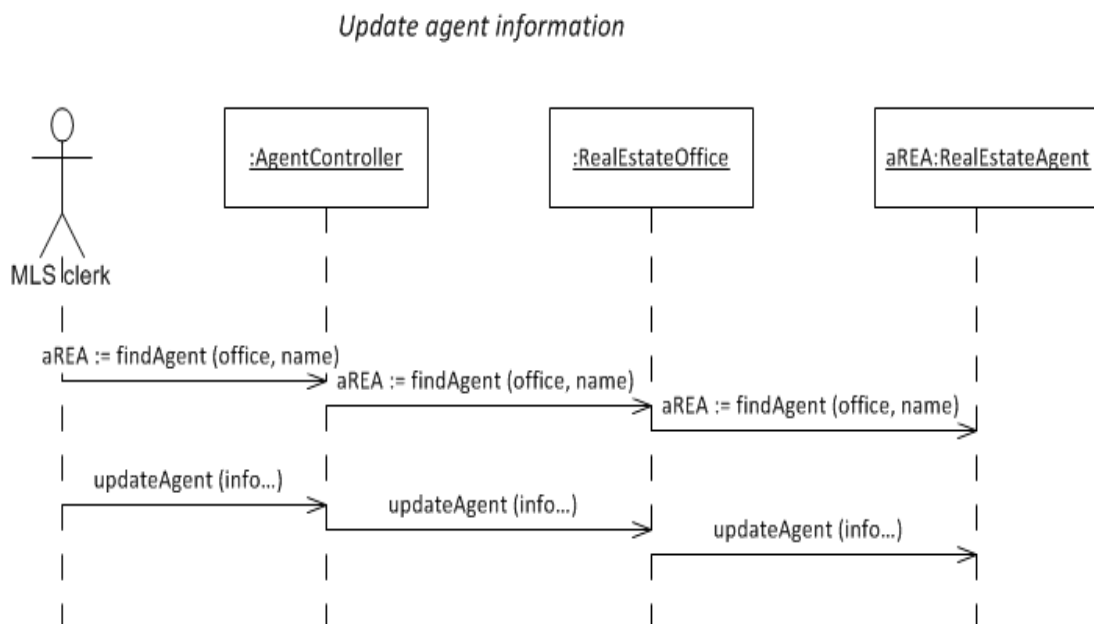
1. A first-cut DCD by using the problem domain classes that you identified in Chapter 4.



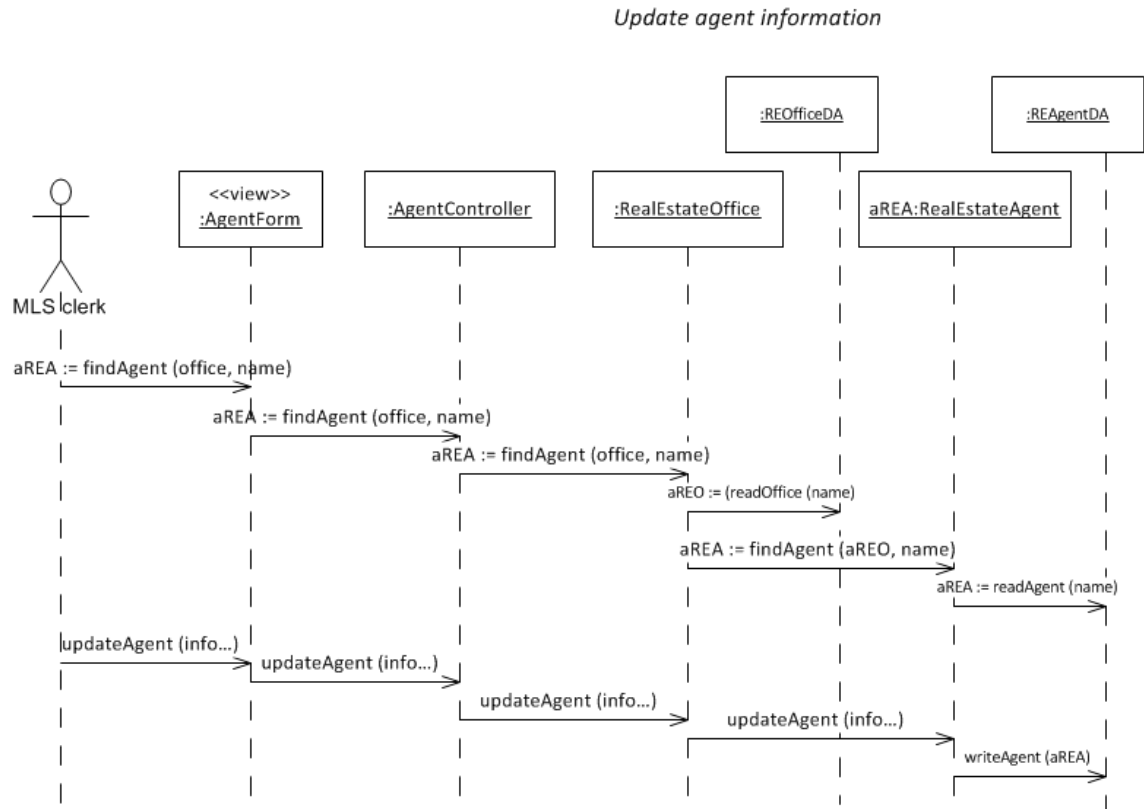
2. A first-cut communication diagram for the *Create new listing* use case (domain classes and controller class only).



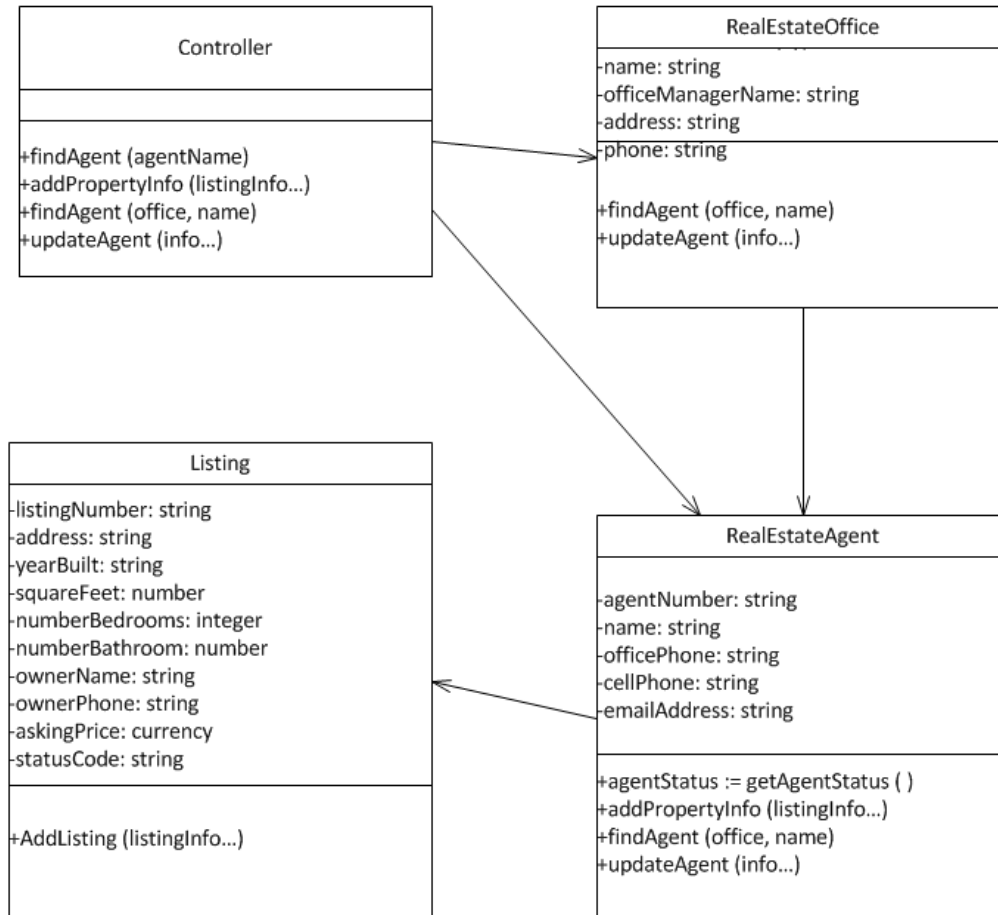
3. A first-cut sequence diagram for the *Update agent information* use case (domain classes and controller class only).



4. A multilayer sequence diagram for the *Update agent information* use case that includes domain classes and data access classes (see item 5 below).
5. A separate multilayer sequence diagram for the *Update agent information* use case that includes the domain classes and the view layer classes.



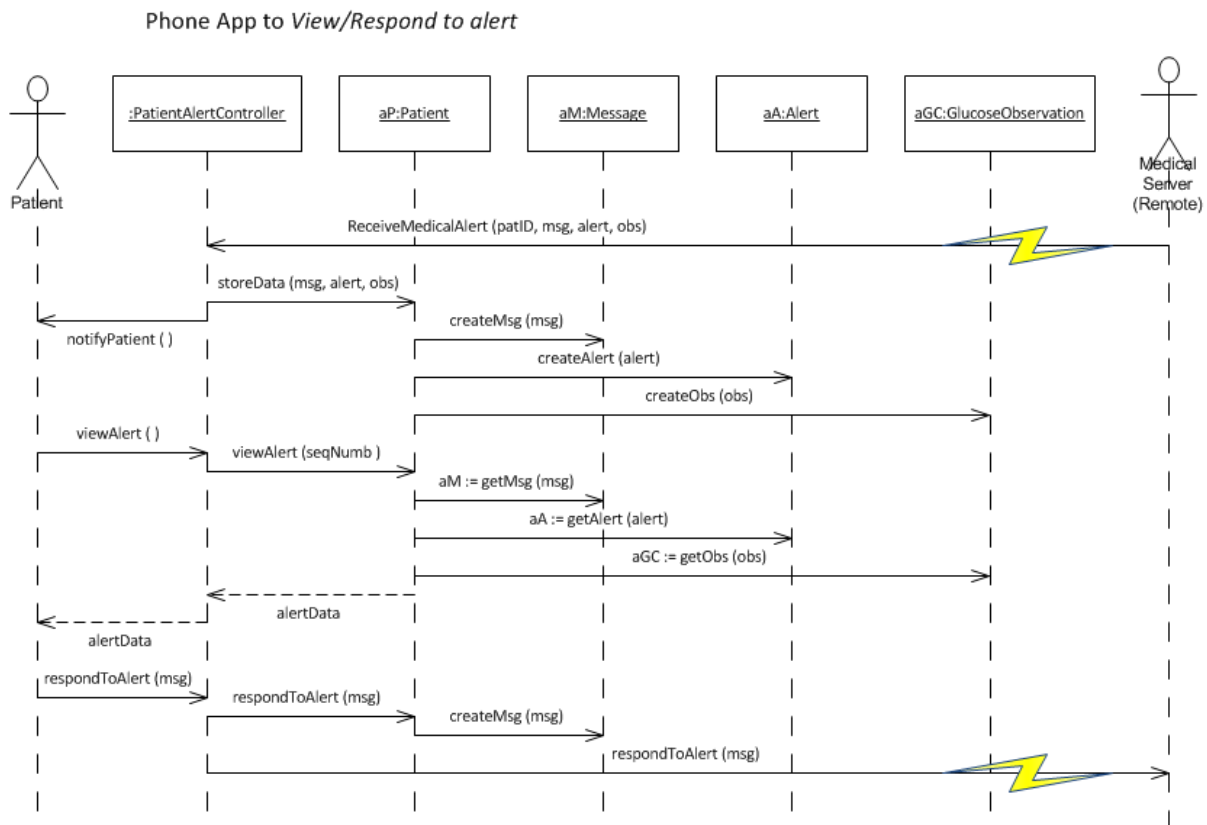
6. A final design class diagram that includes the classes from both use cases. Include elaborated attributes, navigation arrows, and all the method signatures from both use cases.



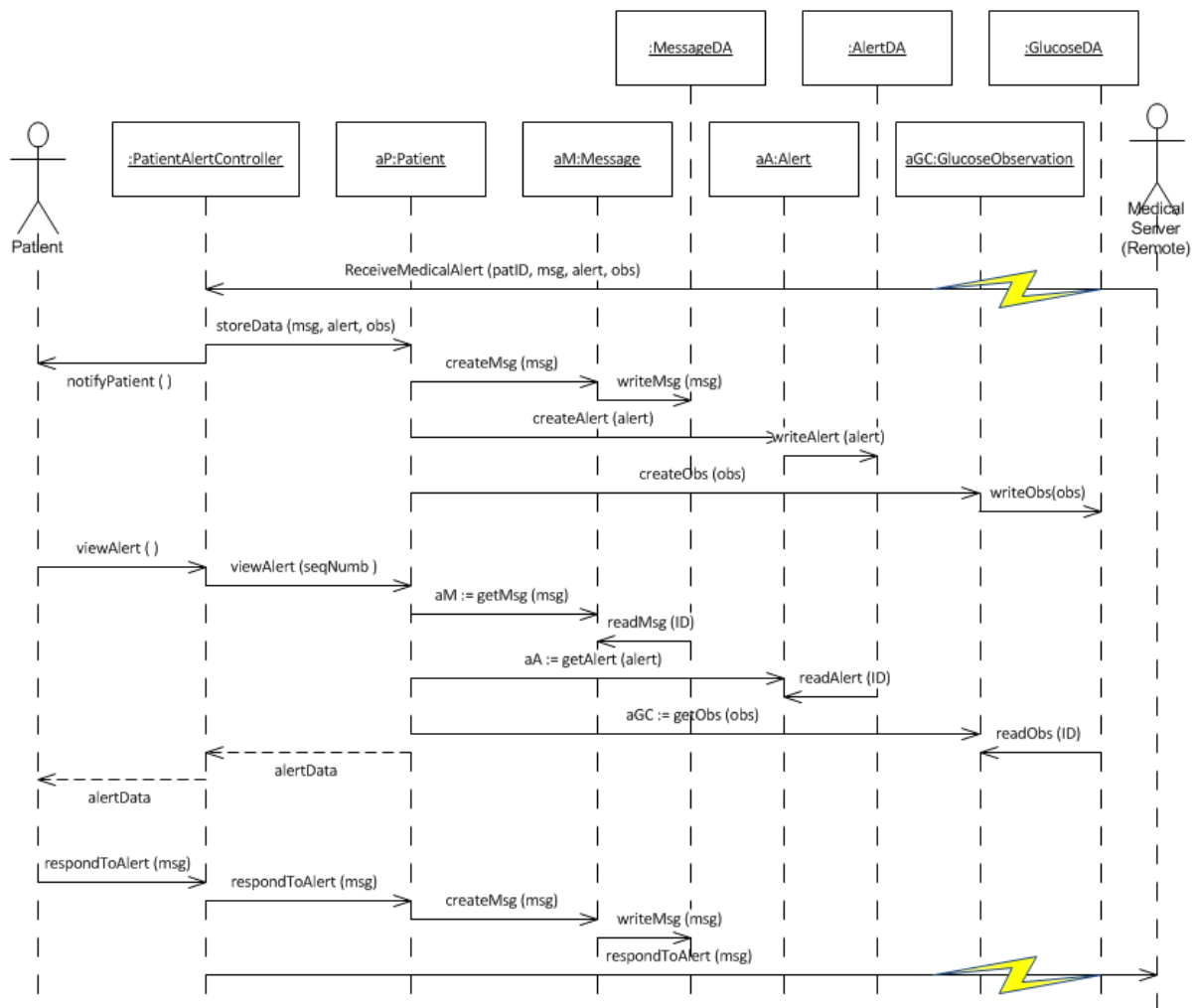
Running Case Studies: Sandia Medical Devices (Chapter 11)

Review your answers to the case-related questions in Chapter 10, and then do the following:

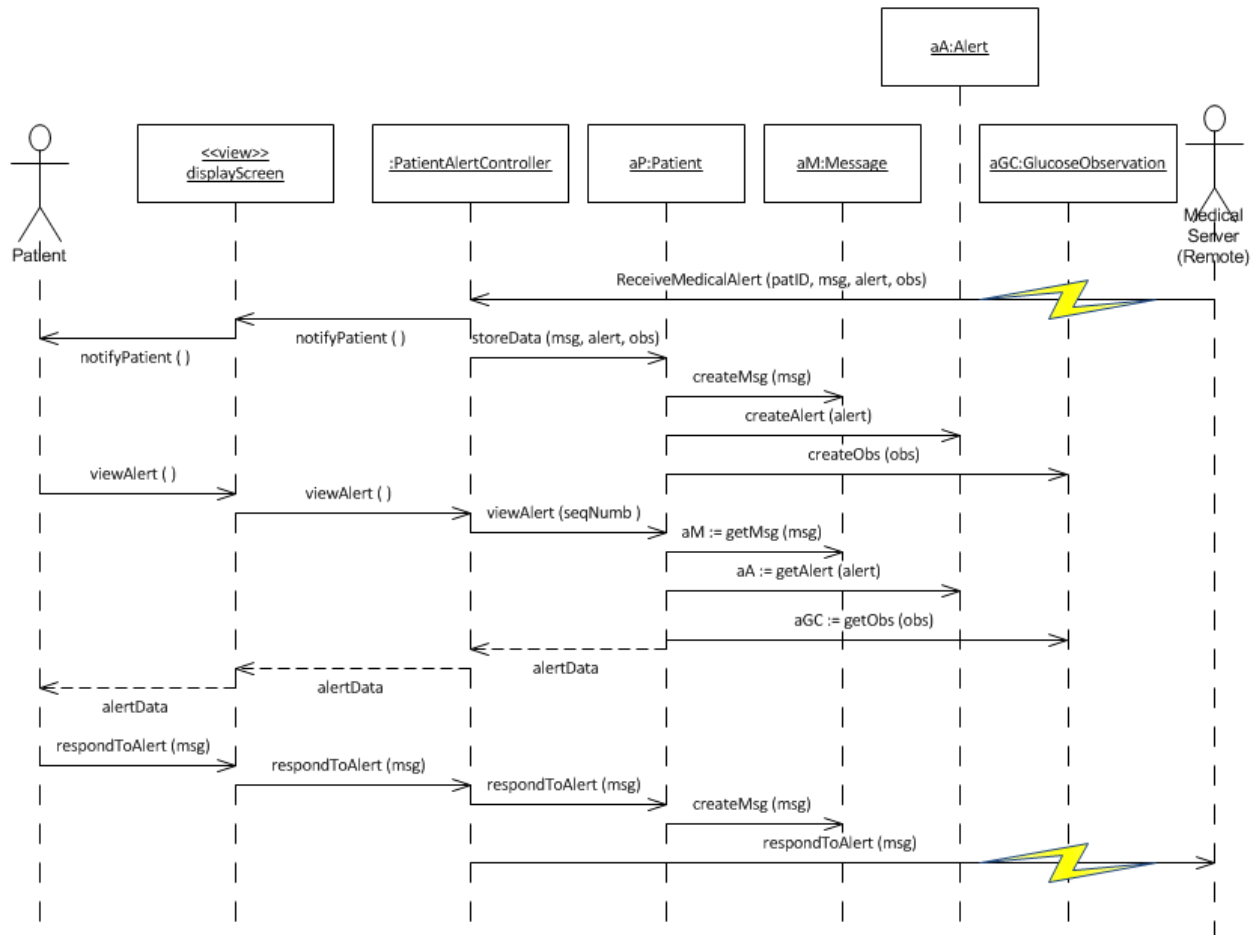
1. Develop a first-cut sequence diagram for the patient use case *View/respond to alert*.



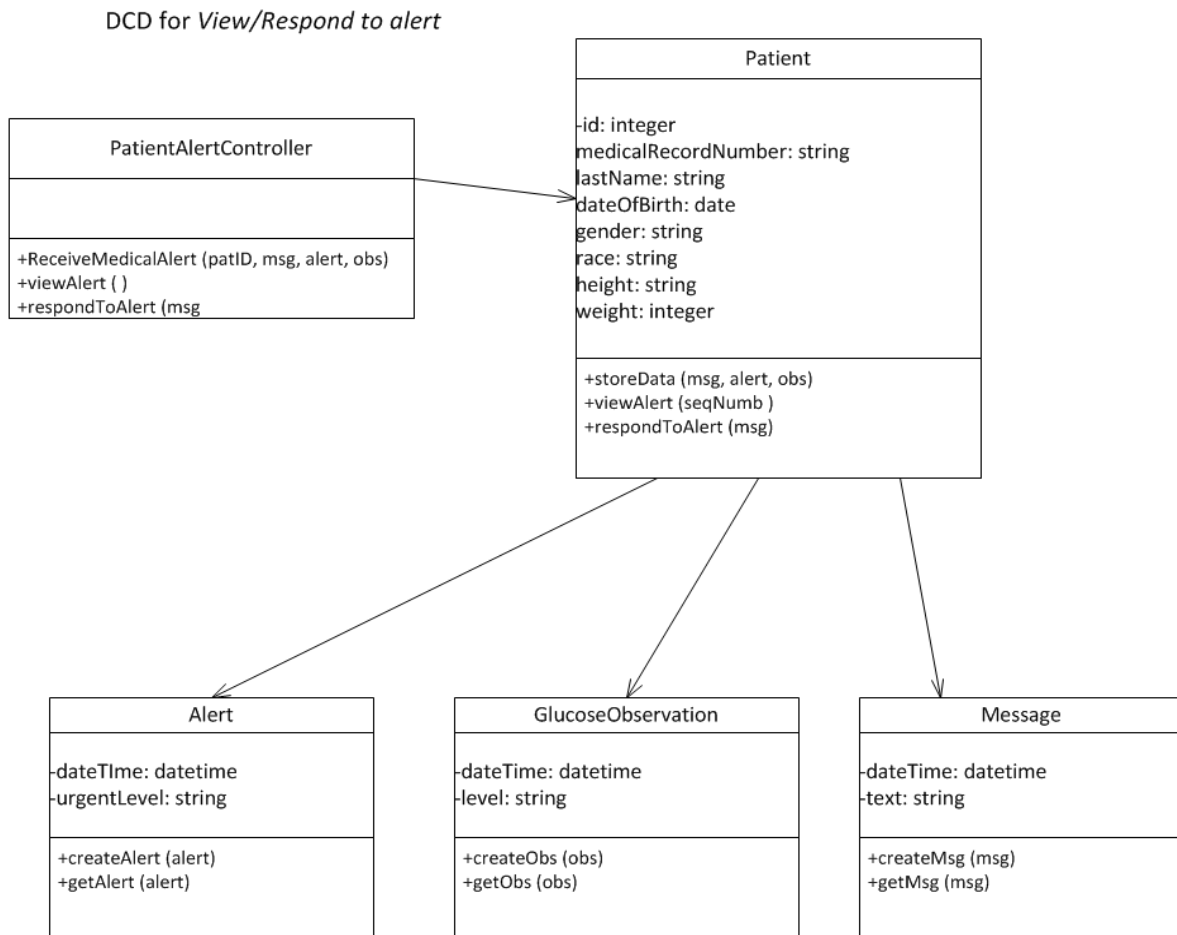
2. Develop a multilayer sequence diagram that includes domain classes and data access classes.



3. Develop a separate multilayer sequence diagram that includes the domain classes and the view layer classes. (We won't combine view and data access layers on the same drawing. It makes the drawing too complex.)



4. Update your DCD from Chapter 10 to include the methods you have identified. Also, include any changes you may have made to navigation visibility and attribute details.



Problems and Exercises (Chapter 12)

2. Assume that RMO will begin asking a random sample of customers who order by telephone about purchases made from competitors. RMO will give customers a 15 per cent discount on their current order in exchange for answering a few questions. To store and use this information, RMO will add two new classes and three new associations to the class diagram. The new classes are Competitor and ProductCategory. Competitor has a one-to-many association with ProductCategory, and the existing Customer class also has a one-to-many association with ProductCategory. Competitor has a single attribute called Name. ProductCategory has four attributes: Description, DollarAmountPurchased, MonthPurchased, and YearPurchased. Revise the relational database schema shown in Figure 12-10 to include the new classes and associations. All tables must be in 3NF.

The following tables must be added to the relational database schema:

Competitor = **Name**

ProductCategory = *CompetitorName*, *CustomerAccountNo*, *MonthPurchased*,
YearPurchased, **Description**, **DollarAmountPurchased**

Primary keys are shown in bold, and foreign keys are shown in italics. Note that the primary key of ProductCategory is guaranteed to be unique only if multiple customer purchases from a competitor in the same month and for the same product category (description) are combined in a single row.

3. Assume that RMO will use a relational database, as shown in Figure 12-10. Assume further that a new catalog group located in Milan, Italy, will now create and maintain the product catalog. To minimize networking costs, the catalog group will have a dedicated database server attached to its LAN. Develop a plan to partition the RMO database. Which tables should be replicated on the catalog group's local database server? Update Figure 12-18 to show the new distributed database architecture.

Assumptions:

Milan will have responsibility for describing and maintaining product items and accessory packages.

Milan will also have responsibility for supporting and maintaining Promotions.

The following tables will need to be replicated on the local LAN. Access requirements (C,R,U, and D) are shown in parentheses.

Promotion (CRUD)

PromoOffering (CRUD)

ProductItem (CRUD)

AccessoryPackage (CRUD)

InventoryItem (R)

Updates to all of these tables are assumed to be relatively infrequent; thus, the performance cost of complete replication with immediate or frequent updates is minimal. Milan can be represented in Figure 12-18 in the same manner as the warehouse LAN or retail store LAN.

- 5. This chapter described various situations that emphasized the need for controls. In the first scenario presented, a furniture store sells merchandise on credit. Based on the descriptions of controls given in this chapter, identify the various controls that should be implemented in the system to ensure that corrections to customer balances are made only by someone with the correct authorization.**

Answers will vary but should include at least the following:

- Transaction logging to note all changes (especially financial) made to the database. Log records should include the login ID of the person making the transaction.
- Financial transaction screens should be available (and visible) only via authorization of the correct level of registered user.
- Possibly a notification report of any changes (other than standard payments) made to correct account balances.

In the second scenario illustrating the need for controls, an accounts payable clerk uses the system to write checks to suppliers. Based on the information in this chapter, what kinds of controls would you implement to ensure that checks are written only to valid suppliers, that checks are written for the correct amount, and that all payouts have the required authorization? How would you design the controls if different payment amounts required different levels of authorization?

Answers will vary but should include at least the following:

- Both manual and automated controls might be needed for this process. The manual control will require authorization by a supervisor on paper documents for payment. Also, a paper audit trail (numbered invoice) might be required.
- Payments made only to valid suppliers can be controlled by having pre-defined PayTo fields that come from a supplier file. The supplier file should be maintained by different people to ensure separation of duties.
- Ensuring that checks are written for the correct amount can be accomplished by making sure a payment amount corresponds with the invoice amount in the system.
- A supervisor can also verify payments for correct amounts and viable suppliers. This can be done either with paper documents or with electronic forms. Before a check is written, a payment transaction can be approved by an electronic signature of a different person.
- Output reports detailing payments should be provided and reviewed.
- Internal edits can be developed to note whether payments are customary and normal. Out-of-range payments can be flagged as exceptions and verified by a manager.
- Different levels of payments will require the same types of controls; however, they may require different electronic signatures by higher-level registered users.

Running Case Studies: Community Board of Realtors (Chapter 12)

In Chapter 4, you developed a domain model class diagram. Using your previous solution or one provided to you by your instructor, update your domain model class diagram with any additional problem domain classes, new associations, or additional attributes that you have discovered as you worked through the previous chapters. Finalize this comprehensive domain model, and then turn it in as part of your solution.

Using this comprehensive domain model class diagram, develop a relational database schema. In the schema, identify the foreign keys that are required. Also, identify a key attribute for each table. You may need to add a key field if there isn't an attribute that could logically serve as the key. Remember that a candidate key for an association class is the combination of the keys of the connected classes. However, it may make sense to define a shorter, more concise key field.

Verify that each table is in first, second, and third normal form. Discuss any discrepancies you had to fix from your first solution. Discuss any tables that may not be in third normal form and why you are leaving it as not-normalized.

Note: We will use the following class diagram from Chapter 4 problem 3 for this problem. The following changes/additions were made:

1. Commissioner was dropped. It is not a logical piece of the MLS system.
2. A few new attributes were added.

Unit 5: Advanced Design Concepts
 Answers for Unit Activities

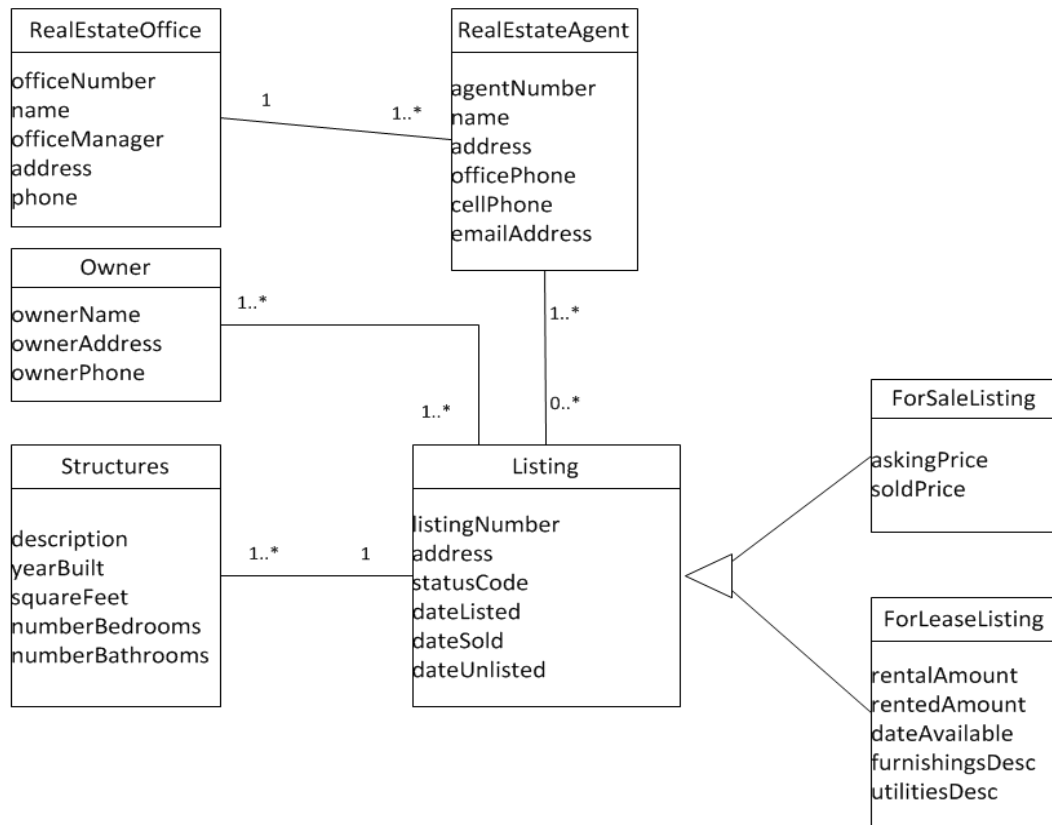


Table	Fields (columns)
REOffice	office_number , office_name, manager, street, city, state_province, postal_code, telephone
REAgent	agent_number , <i>office_number</i> , agent_lastname, agent_firstname, street, city, state_province, postal_code, office_phone, mobile_phone, email_address
Listing	listing_number , listing_type, property_street, property_city, property_state, property_postal, status_code, date_listed, date_sold, date_unlisted, asking_price, sold_price
ForLeaseListing	listing_number , rental_amount, rented_amount, date_available, furnishing_desc, utilities_desc
Owner	owner_number , owner_lastname, owner_firstname, street, city, state_province, postal_code
Structures	structure_number , <i>listing_number</i> , description, year_built, square_feet, number_bedrooms, number_bathrooms
AgentOnListing	<i>agent_number</i> , <i>listing_number</i>
OwnerOnListing	<i>owner_number</i> , <i>listing_number</i>

Note:

Primary key is bold. Foreign key is italicized

The ForSaleListing was combined with the listing table, since most listings are For Sale Listings. ForRentListing is a separate table because the information is unique and there are only a few of those types of records. A new field, listing_type, was added to denote rental listings.

Neither the type codes (string, integer, number, etc.) nor the length have been include. This information will have to be added before the tables can be entered into a database.

The tables are in 3NF with one exception. State is functionally dependent on Postal_code, i. e., a state can be determined by postal code. But due to common usage of always having state and postal code included, they are maintained together. (An alternative is to have a separate postal_code to state translation table.)

Running Case Studies: Sandia Medical Devices (Chapter 12)

Part 1.

Review the original system description in Chapter 2, the additional project information in Chapters 3, 4, and 8, and the domain class diagram shown in [Figure 12-26](#) to refamiliarize yourself with the proposed system. Assume that the type attribute of the AlertCondition class identifies one of three alert types:

1. Glucose levels that fall outside the specified range for 15 minutes (three consecutive readings)
2. Glucose levels that fall outside the specified range for 60 minutes (twelve consecutive readings)
3. An average of glucose levels over an 8-hour period that falls outside a specified range

The specified range for an AlertCondition object is the set of values between and including lowerBound and upperBound. AlertCondition objects also include an effective time period specified by the attributes startHour and endHour, which enables physicians to set different alert parameters for sleeping and waking hours.

When an alert is triggered, an object of type Alert is created and associated with an alertCondition object. The dateTime attribute records when the Alert object was created, and the value(s) attribute record(s) the glucose levels (alert types 1 and 2) or average level (alert type 3) that fell outside the specified range. Each Alert object is indirectly related to a Patient object via the association between Alert and AlertCondition and the association between AlertCondition and Patient. Develop a set of relational database tables based on the domain class diagram. Identify all primary and foreign keys, and ensure that the tables are in 3NF.

Table	Fields (Columns)
Patient	patientID , physicianID, deviceID, medical_rec_number, last_name, first_name, birthdate, gender, race, height, weight
Physician	physicianID , last_name, first_name
MonitorDevice	deviceID , serial_number, manufacturer, manufacture_date, firmware_version
AlertCondition	alert_condID , patientID, type, start_hour, end_hour, upper_bound, lower_bound
Alert	alert_number , alert_condID, date_time, value
CellPhone	phoneID , patientID, phone_number, operating_system, os_version, application_version
GlucoseObservation	observation_number , patientID, date_time, level

Primary keys are written in bold font.

Foreign keys are written in italic font.

All tables are in 3NF. (Correctly built class diagrams always result in tables in 3NF.)

Part 2.

Based on what you learned in this chapter about databases, controls, and system security, review your answers to the questions for this case in Chapter 6. Assume that the patient's cell phone and the centralized servers are different nodes in a replicated database architecture and are regularly synchronized. What changes, if any, should be made to your answers now that you have a deeper understanding of databases, controls, security, and related design issues?

Answers will vary by student.

Unit 6: Making the System Operational

This unit discusses the activities required for making a system operational. After the system has been carefully designed, the system development and testing takes place. Further issues for making the system operational deal with *data conversion*, the *configuration* and *installation* processes, *documentation*, and *training activities*.

Section 1: Implementation and Deployment Activities

This section focuses on the implementation and deployment activities of the systems development life cycle (see Figure 13-1). Implementation activities are activities that deal with building, testing, and integrating system components. Deployment activities include all activities that aim at completing system tests and deploying the system.

In this section, an overview of implementation and deployment activities is given. Subsequently, the various types software product testing are discussed. Then, deployment activities are addressed in more detail, including a discussion on how to convert and initialize data, how to train users, and how to configure the production environment. Furthermore, activities and concepts around planning and managing implementation, testing, and deployment are described in detail. The section is concluded by taking another look at the RMO case study.

Learning Objectives

When you have completed this section, you should be able to do the following:

1. Describe implementation and deployment activities.
2. Describe various types of software tests, and explain how and why each is used.
3. List various approaches to data conversion and system deployment, and describe the advantages and disadvantages of each.
4. Explain the importance of configuration management, change management, and source code control to the implementation, testing, and deployment of a system.



Learning Objective 1

Describe implementation and deployment activities.

Objective Readings/Activities



Readings: Please read the following section in Chapter 13 of the textbook:

- Tri State Heating Oil: Juggling Priorities to Begin Operation (page 410)
- Overview (pages 410–411)



Review: Please check your learning with Review Questions 1 on page 438 of the textbook.

Learning Objective 2

Describe various types of software tests and explain how and why each is used.

Objective Readings/Activities



Readings: Please read the following section in Chapter 13 of the textbook:

- Testing (pages 411–417)



Review: Please check your learning with Review Questions 2–4 on page 438 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 3

List various approaches to data conversion and system deployment, and describe the advantages and disadvantages of each.

Objective Readings/Activities



Readings: Please read the following section in Chapter 13 of the textbook:

- Deployment Activities (pages 417–423)



Review: Please check your learning with Review Questions 5 and 6 on page 438 of the textbook.

Learning Objective 4

Explain the importance of configuration management, change management, and source code control to the implementation, testing, and deployment of a system.

Objective Readings/Activities



Readings: Please read the following section in Chapter 13 of the textbook:

- Planning and Managing Implementation, Testing, and Deployment (pages 423–434)
- Putting It All Together – RMO Revisited (pages 434–437)



Review: Please check your learning with Review Questions 7–11 on page 438 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Unit 6 – Summary and Unit Activities



Readings: Please read the Chapter 13 Summary on page 437 of the textbook.



Review:

- Review the Key Terms on page 437, and check if you know what they mean.
- Answer question 1 from Problems and Exercises on page 438 of the textbook. ([Suggested solutions are available here](#))

Unit 6: Making the System Operational

Answers for Unit Activities

Problems and Exercises (Chapter 13)

- 1. Describe the process of testing software developed with the IPO (input, process, output), top-down, and bottom-up development orders. Which development order results in the fewest resources required for testing? What types of errors are likely to be discovered earliest under each development order? Which development order is best, as measured by the combination of required testing resources, and ability to capture important errors early in the testing process?**

Software testing under top-down development order starts with an implementation of the topmost module and stubs for each of the second-level modules. One by one, each stub is replaced with a fully implemented module, which is then tested. As each stub is replaced, new stubs must be created for any subordinate modules. The process continues until all of the modules have been fully implemented and tested.

Bottom-up development order starts with a driver for one related set of bottom-level modules and the initial implementation of one of the bottom-level modules. After that module is tested, another bottom-level module is created, and the driver is modified to call it (or a new driver is created). After all bottom-level modules in a calling group have been implemented, the driver is replaced with a new module, and a new driver is written to call the new module. Implementation and testing moves up the structure chart in this fashion until all program modules have been implemented and tested.

Under top-down development order, a stub is written once for every module. Under bottom-up development order, a smaller set of drivers is written, but each driver must be updated several times as subordinate modules below it are implemented. In addition, driver modules are generally more complex than stub modules (for example, compare the driver in Figure 13-4 to the stubs in Figure 13-5). Thus, bottom-up development order tends to consume more resources because more effort is expended writing and modifying modules that will be discarded after testing.

Top-down development order finds errors in upper-level modules first. These modules typically contain the business logic of the system and integrate the activities of the lower-level modules. Thus, logic and integration errors are found sooner with top-down development order.

Bottom-up development order finds errors in the lower-level modules first. These modules typically contain detailed procedural code including data access, I/O formatting, and computations. Thus, errors in those modules are found sooner with bottom-up development.

Neither implementation order completely dominates the other when required resources and early error detection are considered. If errors are assumed to be evenly distributed through the structure chart and if they are assumed to be equally “fixable” regardless of structure chart location, then top-down development order is better due to its lower resource consumption. But many systems have complex procedural, I/O, or data access modules, and discovering errors in those modules early in the development process can save time and testing resources.

Unit 7: Current Trends in Systems Development

This unit introduces some of the trends in system development. It first discusses trends in system development methodologies, including *the Unified Process*, *Extreme Programming*, and *Scrum*. Subsequently, trends in technology infrastructure are discussed, focusing on areas such as client computing devices, Internet and telephone communications, and back-end computing. In addition, this unit introduces trends in application software availability, including concepts such as software as a service and open source software. Furthermore, the Web as an application platform is discussed.

Readings/Activities



Please read the following sections in Chapter 14 of the textbook:

- Valley Regional Hospital: Measuring a Project's Progress (page 444)
 - Overview (pages 444–445)
-

Section 1: Trends in Systems Development Methodologies

In Unit 2, you were introduced to Agile development. The Agile philosophy has proven to be an effective way to approach software development in today's fast-paced, continually changing landscape of computer applications. However, the Agile philosophy only proposes principles; it isn't meant to be a complete methodology, with practices and action steps. In this section, three methodologies are presented that incorporate Agile principles, but are also complete methodologies, with specific techniques and practices. These methodologies are The Unified Process, Extreme Programming, and Scrum.

Learning Objectives

When you have completed this section, you should be able to do the following:



1. Describe the elements of the Unified Process (UP).
2. Compare and contrast the features of Extreme Programming and Scrum development.

Learning Objective 1

Describe the elements of the Unified Process (UP).

Objective Readings/Activities



Readings: Please read the following sections in Chapter 14 of the textbook:

- Trends in System Development Methodologies (pages 445–450)



Review: Please check your learning with Review Questions 1–3 on page 476 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Learning Objective 2

Compare and contrast the features of Extreme Programming and Scrum development.

Objective Readings/Activities



Readings: Please read the following section in Chapter 14 of the textbook:

- Extreme Programming (pages 450–454)
- Scrum (page 454)
- Scrum Philosophy (pages 454–456)



Review: Please check your learning with Review Questions 4–9 on page 476 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Section 2: Trends in Technology Infrastructure

The technology infrastructure refers to the computing devices along with the operating system and communication software that allow those devices to function. The rapid pace of change in

microprocessors for the past 30 years has allowed manufacturers to bring a continual stream of faster, more powerful computing devices to market.

At first, there was the tremendous growth in the speed and computing capabilities of personal computers. The same increased capacity also occurred in reasonably priced server computers. No longer were large, expensive mainframe computers necessary to support high-speed and high-capacity computers. More recently, as the increase in capacity of personal and server computers has become more moderate, the most rapid changes are occurring in such handheld mobile devices as tablets and smartphones.

In this section, some of the trends in technology infrastructure are discussed, including client computing devices, Internet and telephone communications, and back-end computing.

Learning Objectives

When you have completed this section, you should be able to do the following:



1. Describe the major trends in connectivity, Internet, and telephone technologies.

Learning Objective 1

Describe the major trends in connectivity, Internet, and telephone technologies.

Objective Readings/Activities



Readings: Please read the following section in Chapter 14 of the textbook:

- Trends in Technology Infrastructure (pages 457–461)



Review: Please check your learning with Review Questions 10 and 11 on page 476 of the textbook.

Section 3: Trends in Application Software Availability

Partly due to the changes in the technology infrastructure described earlier, people and organizations are finding new ways to deploy and provide applications. Historically, when a large or small organization needed a software application to support some organizational

procedure, it either developed that software itself or, if the problem was general enough, purchased software and modified it to fit the in-house procedure. One of the major trends in today's information systems environment is that new methods are available for obtaining software functionality. This section discusses several of the more prevalent ones.

Learning Objectives

When you have completed this section, you should be able to do the following:



1. List and describe the various methods of deploying application software.

Learning Objective 1

List and describe the various methods of deploying application software.

Objective Readings/Activities



Readings: Please read the following section in Chapter 14 of the textbook:

- Trends in Application Software Availability (pages 461–465)



Review: Please check your learning with Review Questions 12–14 on page 476 of the textbook.

Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Section 4: The Web as an Application Platform

As the use of the Internet in mobile and computing devices becomes more widespread, a fundamental shift in our society—the way we work and the way we interact—is occurring. For many people—either at work or at home—the web browser is the computer program they use the most; in fact, for some people, it is the only application they use. For many others, web-based applications are used in almost all aspects of their employment. This trend toward immediate connectivity has also caused a fundamental shift in software development and deployment. The Web has become the primary environment for the deployment of new software applications and systems. The majority of new applications are being written for the Internet—either as browser-based applications or as freestanding Internet applications.

Development of this type follows the same pattern that has been discussed throughout this course. In this section, we focus on browser-based applications as a new type of development platform for application software.

Learning Objectives

When you have completed this section, you should be able to do the following:



1. List and describe the various elements that enhance web applications and various approaches to developing Rich Internet Applications.

Learning Objective 1

List and describe the various elements that enhance Web applications and various approaches to developing Rich Internet Applications.

Objective Readings/Activities



Readings: Please read the following section in Chapter 14 of the textbook:

- The Web as an Application Platform (pages 465–475)



Review: Please check your learning with Review Questions 15–22 on page 476 of the textbook.



Supplementary Readings: Please check the [Digital Reading Room](#) for additional reading materials on the topic of this section.

Unit 7– Summary and Unit Activities



Readings:

- Please read the Chapter 14 Summary on page 475 of the textbook.



Review:

- Review the Key Terms on page 476, and check if you know what they mean.
- Answer question 2 from Problems and Exercises on pages 476–477 of the textbook.

([Suggested solutions are available here.](#))

Unit 7: Current Trends in System Development

Answers for Unit Activities

Problems and Exercises (Chapter 14)

- 1. Consider XP's team-based programming approach in general and its principle of allowing any programmer to modify any code at any time in particular. No other development approach or programming management technique follows this particular principle. Why not? In other words, what are the possible negative implications of this principle? How does XP minimize these negative implications?**

Answers will vary.

Some of the potential problems of allowing anybody to make changes to the code are as follows:

- Not communicating to the primary author when changes are made.
- Primary author (and tester) not being aware of changes and making conflicting tests or changes.
- Primary author becoming defensive about code quality.
- Primary author wanting to "own" the code.

XP helps with these problems by 1) establishing ground rules that code can be modified by multiple programmers, and 2) setting up an effective communication system to let others know when code is changed. Also by integrating continuously with small releases, the impacts of code changes are kept current for all to use.