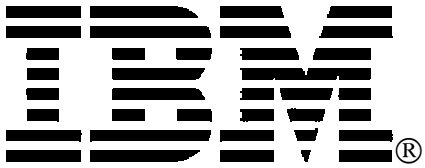


Rational® software

REQ480 Mastering Requirements Management with Use Cases

Student Workbook

Version 2003.06.00

Part# 800-026338-000

IBM Software Group
Rational University
REQ480 Mastering Requirements Management with Use Cases
Student Workbook
Version 2003.06.00

May, 2003

Printed in the United States of America.

Contents

WP: White Papers

- WP1: The Five Levels of Requirements Management Maturity
- WP2: Features, Use cases, Requirements, Oh My!
- WP3: Using the RUP to Evolve a Legacy System
- WP4: Generating Test Cases From Use Cases
- WP5: Structuring the Use-Case Model
- WP6: ACRE: Selecting Methods For Requirements Acquisition

▸ ▸ ▸ Contents

EX: Exercises

Exercise 3.1
Identify Actors and Use Cases

The purpose of this exercise is to identify actors and use cases for a simulated project. You have been introduced to the online Course Registration System that is the case study throughout this module. For this exercise, use information and artifacts from the Course Registration System case study.

First identify the actors in the example system. “Student” is identified as an actor. Who or what else interacts with this system? Refer to the questions for identifying actors.

For each of these actors, identify the types of interactions each might have with our system. Refer to the questions for identifying use cases and the guidelines for naming use cases.

Objectives

In this exercise, complete the following tasks:

- Identify the actors who interact with the system.
- Identify the use cases.
- Sketch a use-case diagram for the system.

Scenario

You have just been assigned the job of lead system analyst for a new system. You have been given a problem description ("Initial System Requests" below). Your first task is to understand the requirements for the new system. From the Initial Requests, you develop a use-case model of the requirements. In this first part of the modeling process, you identify actors and use cases and develop a use-case diagram.

As you look at the initial requests for the system, note that the requirements are far from complete. Note what assumptions you make, as well as what other information you want to ask your customer.

Directions

1. Read the Initial System Requests document.

As a whole class with the instructor leading the activity:

1. Run a short use-case workshop to decide on actors and use cases.
2. Draw the use-case diagram on easel paper or the board.
   - Show actors, use cases, and communicates-associations.
3. Compare the diagram with the sample solution.
4. Discuss the "Discussion Topics" at the end of this exercise.

Initial System Requests

Wylie College is planning to develop a new online Course Registration System. The new Web-enabled system replaces its much older system developed around mainframe technology. The new system allows students to register for courses from any Internet browser. Professors use the system to register to teach courses and to record grades.

Because of a decrease in federal funding, the college cannot afford to replace the entire system at once. The college will keep the existing course catalog database where all course information is maintained. This database is an Ingres relational database running on a DEC VAX. The legacy system performance is poor, so the new system accesses course information from the legacy database but does not update it. The registrar's office continues to maintain course information through another system.

Students can request a printed course catalog containing a list of course offerings for the semester. Students can also obtain the course information online at any time. Information about each course, such as professor, department, credit hours, and prerequisites assists students in making informed decisions.

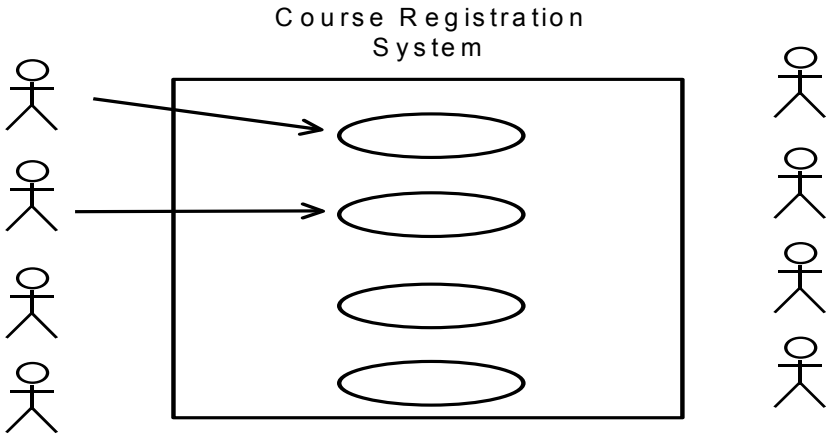
The new system allows students to select four course offerings for the coming semester. In addition, each student indicates two alternate choices in case the student cannot be assigned to a primary selection. Courses have a maximum of ten and a minimum of three students.

The registration process closes on the first or second day of classes for the semester. Any course with fewer than three students enrolled on the day registration closes is cancelled. All courses without an instructor on the day registration closes are cancelled. Students enrolled in cancelled classes are notified that the course has been cancelled, and the course is removed from their schedules. The registration system sends information about all student enrollments to the Billing System so that the students can be billed for the semester.

For the first two weeks of the semester, students are allowed to alter their course schedules. Students may access the online system during this time to add or drop courses. Changes in schedules are immediately sent to the Billing System so that an updated bill can be sent to the student.

At the end of the semester, the student can access the system to view an electronic report card. Since student grades are sensitive information, the system must employ security measures to prevent unauthorized access. All students, professors, and administrators have their own identification codes and passwords.

Professors must be able to access the online system to indicate which courses they want to teach. They also need to see which students signed up for their course offerings. In addition, professors can record the grades for the students in each class.



Identify Actors

Who uses the system?

Who gets information from the system?

Who provides information to the system?

Where in the organization is the system used?

Who supports and maintains the system?

What other systems use this system?

Identify Use Cases

What are the goals of each actor?

- What will the actor use the system for?
- Will the actor create, store, change, remove, or read data in the system?
- Will the actor need to inform the system about external events or changes?
- Will the actor need to be informed about certain occurrences in the system?

Does the system supply the business with all of the correct behavior?

Exercise 4.1
The Problem Behind the Problem

In this exercise, familiarize yourself with the RU Financial Services case study. This case study is used throughout the course as an example of applying requirements management concepts discussed in class. The class may decide to develop a different class project. The directions for the exercises are written generically so that they should be applicable for any class project you choose.

The purpose of this exercise is to apply problem analysis techniques. Identify any hidden problems that may exist in the problem domain of your class project. The "real" problem(s) may not be those most obvious at the beginning.

Objectives

In this exercise, complete the following tasks:

- Review the details for the RU e-st class project.
- Demonstrate how to use problem analysis techniques to find the root causes for a problem.

Scenario

You have been given the role of project lead for the RU e-st project. The first task is to validate that a stock trading system is, indeed, the best solution to solve the business' problems and help the business achieve its goals. To do this you must lead your team through the activities of problem analysis.

Directions

Gain an understanding of the business by reading Parts 1, 2, and 3 of the business problem and business background.

Once you have read these, work as a class to perform the problem analysis. The directions for this are located on page 10.

Part 1: Business Problem

RU Financial Services is an investment management organization with 500 offices located across every state. The head office is located in the nation's capital. RU Financial Services was the first to market with online stock trading in 1992. The personal stock trading business exceeded all expectations and became a significant part of RU Financial Services' revenue for the next eight years. Since then the company has had difficulty attracting new customers to their service, and existing customers have been switching to rival companies at an alarming rate.

The goal is to see how RU Financial Services can regain its market share and continue to grow the personal stock trading side of the business. The board members have done some analysis and decided that the current stock trading system needs replacing. Harold Smedley, the Vice-President of RU Financial Services, has asked you to determine if this truly is the best course of action.

- Based on a recent customer satisfaction survey, 60% of customers were not satisfied with how long it took for updates to the software to become available. This dissatisfaction figure needs to be reduced to less than 5%.
- The current business model for RU Financial Services mandates that infrastructure costs should not be more than 10% of the revenue generated by the business. This is currently running at 40%.
- Maintenance costs of the software are currently running at 25% above the average maintenance costs of other software in the company. This figure must be reduced to no more than 15%.
- Based on a recent survey, it was found that 70% of customers require additional services that the current system does not provide. The main requests were: real-time stock quotes; ability to research stock, ability to fund trading account directly from a savings account held in another financial institution, and Web access instead of having to install client software.

- About 2% of customers have experienced errors in their trades in the last six months. This figure must be reduced to 0%.

Part 2: The Business Background

RU Financial Services introduced their online trading system in 1992 before the birth of e-commerce. At that time, the company had a corporate standard architecture of *VAX 6300* mainframe computers and *DECstation 3100*s for desktop machines. For this project, the company chose to use a *VAXft 3000*, DIGITAL's first fault tolerant machine. The new corporate standard for desktops is a standard PC running Microsoft Windows 2000. The server standard is a high-end PC running Windows 2000 Server or an *IBM eServer iSeries 890*.

The current stock trading system employs a client-server architecture. Updates to the software are mailed to the customer for installation bi-annually. Critical bug fixes are sent out on demand. The customer is required to install the client software on a machine running Microsoft Windows 98 or higher or an Apple running Mac OS X. The client software is written in QuickBasic and runs in a console window. The client software communicates with a server application via a 56K dial-up connection.

The server application runs on a VAXft 3000. The server software is written in COBOL and comprises 500,000 lines of code. The server application is the last application that runs on a VAXft 3000. All other corporate applications have been migrated to the new corporate standard platform. The VAXft 3000 is under a 24x7 maintenance contract. The cost of the maintenance contract is $200,000 per annum.

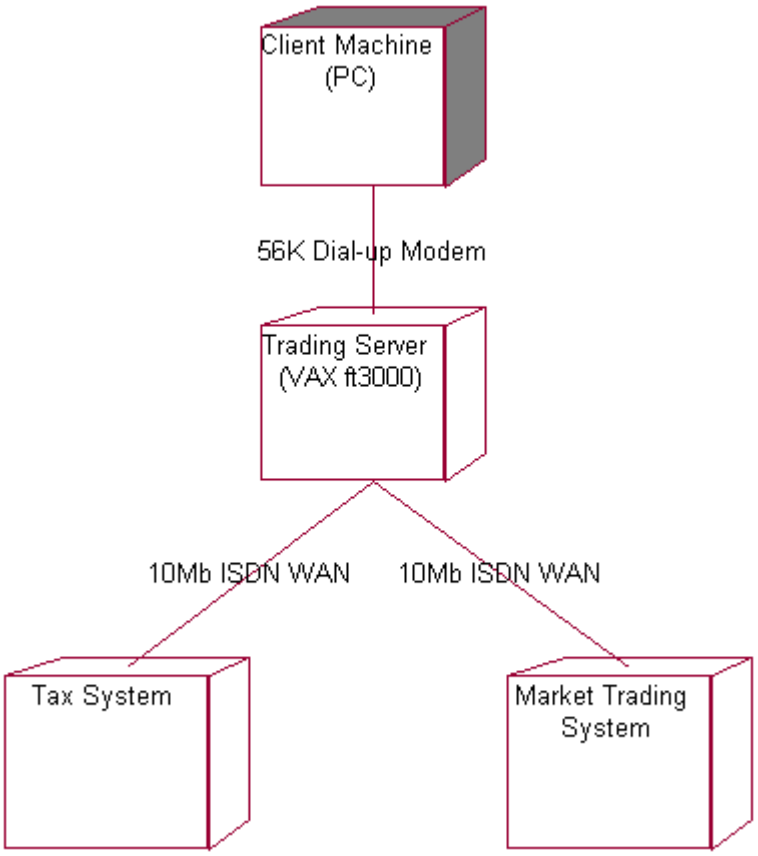
The company phased out the systems written in COBOL and QuickBasic some years ago. All software in the company is written one of the following: C++, Java, or Visual Basic. Due to the success of the system, the company did not want to rewrite the stock trading system and risk the introduction of bugs and loss of revenue. The current stock trading system is the only system that uses COBOL and QuickBasic.

The company has adopted the Rational Unified Process® for developing and maintaining all systems in the organization. All projects are required to use this process. The company mentor for the RUP® is Kelly Richardson.

There is no existing documentation for the client or server applications. None of the original development team remains employed at the company; therefore, contract programmers maintain the software. As a result, the code for both the client and the server has become very unstructured and fragile. Recently, this has caused some major bugs that have caused errors in trades and required multiple updates to the client and server software.

A dedicated testing group performs testing of the system. Because the system runs in a console window on the client PC, the testers are unable to use their automated testing software. The test scripts are performed manually. New tests are rarely designed because they are document based.

Part 3 – Current System Architecture



Directions

Working together as a class:

1. Identify root causes of the problem using a fishbone diagram.
   - Perform problem analysis, using the fishbone diagram technique. (Draw the fishbone diagram on easel paper or board.)
   - Identify the largest contributors to the problem.
   - Discuss whether the proposed solution really is the best solution.
   - Compare the class solution to the sample solution. (Only do this if you use the RU e-st project.)

Exercise 4.2
Describe the Problem

The purpose of this exercise is to describe the problem and start defining the boundaries of the solution in the Vision document.

Understand all the stakeholders in your project in addition to the customer who is paying the bill. Make sure you consider all those affected by the outcome of the system, including stock shareholders, system maintainers, and developers.

Objectives

In this exercise, complete the following tasks:

- Identify the stakeholders for the project
- Identify the actors who interact with our system.
- Use the standardized problem statement format provided to formulate a problem statement for the class project.

Scenario

An analysis of the root causes during the problem analysis found that replacing the current system is the best solution to solving the problems the company is experiencing. You now start work creating a Vision document for the project.

Directions

In this exercise, complete the following tasks:

1. Work in your groups to start the Vision document.
    a. Identify key stakeholders.
    b. Define the system boundaries by identifying actors who interact with the system. (Refer to exercise 4.1.)
    c. Identify constraints on your project.
2. As a whole class:
    a. Create a consolidated list of stakeholders from each group.
    b. Create a consolidated list of actors from each group.
    c. Together, write a problem statement to summarize the problem.

Note: If you are not using RU e-st as your class project, this scenario should be adjusted accordingly.

Exercise completion

At the end of this exercise you should have the beginnings of your system Vision document.

Remember, the Vision document is developed iteratively, and you should not expect to complete it in one sitting.

Part 1: Identify Stakeholders (Section 3.2 of the Vision document)

Using the Vision document included in your student pack, complete **Section 3.2: Stakeholder Summary**.

Who are the **key** stakeholders for the class problem?

Who are the stakeholders that will actually be using the system (potential actors)?

Which stakeholders will you seek to obtain requirements from? For the stakeholders that are not sources of requirements, what should you do with them?

All users of the system are stakeholders because they are materially affected by the system. But there are some stakeholders (for example, stockholders) who are probably not actors because they will never use the system.

Part 2: Identify Actors and Boundaries

An actor is a user of the system. A user can either be an individual or an external system. By definition, an actor is outside of the system, therefore; identifying the actors helps to define the boundaries of the system.

Label some of the actors in the diagram below. Use the following questions to help you determine if you have found them all.

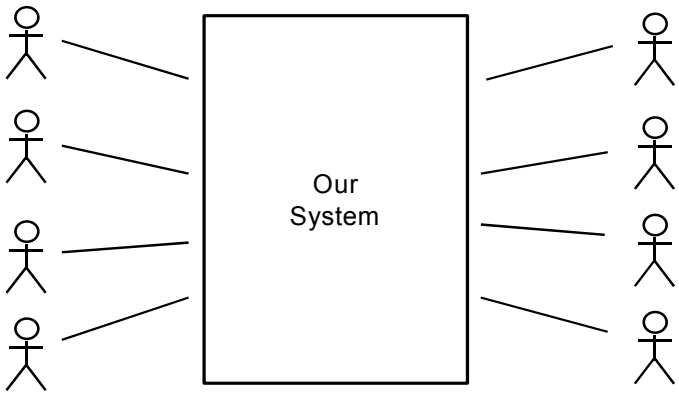
Refer back to the business background in Exercise 4.1 to help you identify your initial list of actors.

Actors

- Who uses the system?
- Who gets information from the system?
- Who provides information to the system?
- Where in the organization is the system used?
- Who supports and maintains the system?
- What other systems use this system?

Boundaries

- What are the interfaces to outside systems for our project?
- How can use cases help us figure out these boundaries?
- What is part of the system?
- What is *not* part of the system?



Part 3: Identify Constraints (Section 6 of the Vision document)

Complete **Section 6: Constraints** of the Vision document.

What types of constraints might you encounter in your class project?

Part 4: Problem Statement (Section 2.2 of the Vision)

Complete **Section 2.2: Problem Statement** of the Vision document.

The problem statement table is a tool for gaining agreement on the problem being solved. This is a simple technique to help articulate the problem and to ensure that everyone agrees on the problem. Fill in the problem statement table for your class project.

Below is a copy of the template and an example to help you get started.

The problem of	(describe the problem)
Affects	(the stakeholders affected by the problem)
The impact of which is	(what is the impact of the problem)
A successful solution would	(list some key business benefits of a successful solution)

Example Problem Statement (for a customer support system)

The problem of	untimely and improper resolution of customer service issues
Affects	our customers, customer support representatives, and service technicians.
The impact of which is	customer dissatisfaction, perceived lack of quality, unhappy employees, and loss of revenue.
A successful solution would	provide real-time access to a trouble-shooting database by support representatives and facilitate dispatch of service technicians, in a timely manner, only to those locations that genuinely need their assistance.

Exercise 5.1
Part 1: Review Customer Requirements Specification

In this exercise, identify and itemize requirements from system specifications. This is a system specification for the online stock trading system.

The goals of this exercise are for students to become familiar with requirements expressed in traditional format and for them to begin to identify declarative requirements.

As you review the system specification

- Refine your current list of actors from Exercise 4.2
- Identify new actors

Objectives

In this exercise, you do the following tasks:

- Review the Online Stock Trading System Specifications.
- Look for possible requirements.
- Mark and number each requirement.
- Refine the solution boundary by revisiting the list of actors.
- Review a sample list of stakeholder requests.
- Update the Vision with the stakeholder needs you have identified. (optional)

Scenario

The board of directors has decided that replacing the current online trading system is the best way forward. In light of the decision to develop a new online stock trading system, the following initial requests have been produced.

Directions

Part 1

Working individually:

1. Turn to the initial requests for the online stock trading system.
   - Note that the specification is not complete.
2. Read through the specification.
3. Mark and number anything you think is a software requirement.

As a whole class:

1. Present the number of requirements you found.
2. Compare the number of requirements from different individuals in the class.
3. Discuss the "Discussion Topics" at the end of this exercise.

Discussion Topics

Why are the numbers so different? How should the differences be resolved?

What are the benefits of standard documents? What are the identification methods for requirements? For example, "shall," delimiters, and so on.

What do you do with a requirement spec that you received from your customer? (Especially when you have no control of the content or the format.)

Were there any changes to your actor list?

RU E-ST SYSTEM SPECIFICATION

Note: The sample in this handout contains only pages 1 through 4 of the specification.

Disclaimer: This is a fictitious system and does not claim to be a model for a good Software Requirement Specification. It is used for exercise purposes only.

INTRODUCTION

RU Financial Services is beginning an e-commerce project to develop software for a Web-based stock and securities trading system. The new system has been tentatively named the **RU e-st** system.

Purpose

The purpose of this SRS is to serve as a statement of understanding between the Online Trading Business and Engineering Divisions of RU Financial Services. Any desired departures from the specifications described herein must be negotiated by the affected parties.

The audience for this SRS includes the engineers who are involved in the development of the system, the marketers who are involved in its advertisement and sale, and the testers who validate the system.

Scope

The overall objective of the **RU e-st** Stock Trading System is to provide our trading customers with a convenient way to manage their stock portfolios. Managing a stock portfolio includes such capabilities as buying and selling and researching stock in a secure environment.

REQUIREMENTS

In this section, all of the functions that the **RU e-st** Stock Trading System Software is to perform are specified. These specifications are first given from total system perspective.

The **RU e-st** system shall provide all of the current functionality of the current stock trading system, plus any additions specified in this document.

Trading Customer Requirements

The **RU e-st** system allows its users to trade securities online, using a Web interface (an existing browser, such as Netscape or Internet Explorer). A trading customer can log on to the Internet anywhere (hotel room, airport, and so on.) Users are able to perform all the basic operations for securities trading: open accounts, trade stocks and other securities, and transfer assets among RU e-st accounts. Users can also obtain current and historical information about their accounts, such as the number of shares held, the current price, and the account total value.

Customers are able to execute many types of trade orders: market trading (buy and sell at current market prices), transfers from one mutual fund to another within one

account, and limit trading (buy and sell at a specific price). The **RU e-st** system uses the existing Market Trading System to perform the securities trades. **RU e-st** also allows users to transfer cash in an account to or from financial institutions such as banks or credit card vendors, using the existing Financial Network System. The usual restrictions apply:

- For online market sales, the securities to sell must be in the customer's account.
- For market purchases, cash for the purchase must be in the account by the settlement date.
- Transfers and purchases from an account are allowed only if, at the time the transaction occurs, they have enough cash to fund the transaction in the account.

Trading customers are also able to obtain information about what is happening in the securities markets. A trading customer can obtain price quotes and news headlines by entering the trading symbol (for example, IBM) for a particular security. The **RU e-st** system obtains current and historical quotes from the existing Quote System service and news items from the existing News System. **RU e-st** also has a feature to broadcast news headlines periodically on the trading customer's screen without being asked.

Regulatory Requirements

The system must report yearly tax information to the IRS and state tax boards. Tax forms must be communicated to the IRS and copies mailed to the customer. The information is also available online for customer viewing.

System Administration Requirements

Updates to the information on the Web pages must be possible without making the system unavailable.

Development Requirements

The system must be written in one of the company's approved programming languages. Refer to document Enterprise Architecture and Development Standards document EA-1234[1] for details.

Hardware Requirements

The hardware platform must be one supported by the enterprise hardware maintenance contract. Supported platforms are specified in the Enterprise Architecture and Development Standards document EA-1234.

---

[1] This document is not supplied with the course.

ü ü ü Exercise 5.1
Part 2: Review Sample Stakeholder Requests

In this exercise, familiarize yourself with the list of sample stakeholder requests. This list is used to illustrate traceability in subsequent exercises.

Note: The following is a sample list of stakeholder requests that were produced during the workflow detail of Understand Stakeholder Needs. This list is included so that the traceability from stakeholder requests to features and use cases can be illustrated. The list is not intended to be exhaustive and there any many other stakeholder requests that you may have elicited.

Directions

Part 2

1. Review the list of sample stakeholder requests. (This is used as input into the next exercise.)
2. List the needs in section 3.7 of your Vision document (Optional).

Requirements	Origin
STRQ1: Want to be able to transfer funds from other accounts (not necessarily held with this firm) to a trading account.	Trading Customer
STRQ2: State and federal regulations require monthly reports of account activity. Refer to specification RUFS-1234 for details of the information required.	Regulatory
STRQ3: The system should allow the use of any browser.	Trading Customer
STRQ4: Customers want to manage their retirement funds.	Trading Customer
STRQ5: Must be able to upgrade the system without taking it offline.	On-line Trading Business
STRQ6: The system should allow traders to trade in multiple markets across the world.	Trading Customer
STRQ7: Must be able to provide convenient answers to customer's most common questions.	Customer Support
STRQ8: The system must provide a secure environment that prohibits fraudulent access.	Trading Customer
STRQ9: Need a way to train customers in the use of the system quickly and conveniently.	Customer Support
STRQ10: The system must operate on hardware that falls under the company's current maintenance contracts.	Finance Department
STRQ11: Need to be able to maintain the system with our current IT hardware and skills. Refer to enterprise architecture document EA-1234 for details.	System Administration
STRQ12: Need account activity statements for tax reporting.	Trading Customer
STRQ13: The system must provide all the basic capabilities of a normal stock broking firm.	Trading Customer
STRQ14: Need to be able to perform research on any given stock.	Trading Customer
STRQ15: The system must allow traders to obtain up-to-date news and alerts on nominated stock.	Trading Customer
STRQ16: The system must provide current and historical information on Trading Acounts. Such as number of shares held, current price, total Trading Account value	Trading Customer
STRQ17: The system shall provide the following types of trades: Market Trades (buy and sell), Limit Trades (buy and sell), and transfers between mutual funds.	Trading Customer
STRQ18: The system shall use the existing Market Trading System to perform securities trades.	Trading Customer
STRQ19: The system must report yearly tax information to the IRS and state tax boards.	Regulatory
STRQ20: Tax forms will be sent electronically to the IRS and state tax authorities.	Regulatory
STRQ21: Tax information can be viewed on-line by the Trading customer and printed if requested.	Regulatory
STRQ22: The system performance must be aceptable to customers that do not have high-speed data access for their computers.	Trading Customer

Exercise 5.2 Brainstorming

In this exercise, come up with some high-level needs of the stakeholders for your system.

The purpose is to:

- Try brainstorming as a requirements elicitation technique.
- Update the Vision document with the needs you identify.

Objectives

In this exercise, complete the following tasks:

- Gather a list of stakeholder needs using brainstorming techniques.
- Clarify and organize the ideas.
- Condense ideas.
- Prioritize remaining ideas.

Scenario

As project lead, you need to identify the needs of the stakeholders for your new system. You have called your stakeholders together for a brainstorming session.

Remember: The needs you identify must help solve the business problem. If they do not, why have them?

To do this exercise, put yourself in the shoes of the stakeholders you identified in Exercise 4.2. Consider what they need from the system.

All stakeholders have all been trained on the rules of brainstorming:

- Clearly state the objective of the session.
- Generate as many ideas as possible.
- Criticism and debate are not allowed.
- Change and combine ideas.

Directions

1. Prepare.
   - Stack of sticky notes for each participant.
   - Large markers for all.
2. Gather ideas.
   - Shout it out.
   - Write it down (include the stakeholder name.)
   - Facilitator posts each idea on the board.
3. Clarify and organize ideas.
   - Describe each idea.
   - Move the notes around.
   - Organize by FURPS or by the stakeholder name.
4. Prune ideas.
   - Discard redundant or outrageous ideas.
   - Store “needs more development” ideas.
   - Combine like or similar ideas.
5. Prioritize remaining ideas.
   - Vote or apply evaluation criteria.

Exercise 6.1
Identify the System Features

In this exercise, refine the Vision document. The content that has been covered in previous modules is formalized in key portions of the Vision document.

Objectives

In this exercise, complete the following tasks:

- Brainstorm the system features based on the stakeholder requests elicited in Exercise 5.1.
- Trace the features to stakeholder requests.
- Refine the Vision.
  - Write a product position statement.
  - List the features for the class project.

Scenario

As project lead, you are responsible for the Vision document for your project. Your team has been working to describe the problem and how to address it, identify stakeholders and users, elicit stakeholder/user needs, identify key product features, and recognize constraints. Now, formalize this information in your Vision document.

Directions

Work in a small group.

1. Review exercise results.
   - Problem statement (Module 4).
   - Stakeholders and actors (Module 4).
   - Fishbone diagram (Module 4).
   - Key stakeholder and user needs (Module 5).
   - Constraints (Module 4).
2. Brainstorm some product features.
   - Look at the list of stakeholder requests.
   - For each stakeholder request, identify some features that you would include in the system that would satisfy these stakeholder requests.
   - Give each feature a requirement tag (For example, FEAT1, FEAT2).
   - For each feature, identify which stakeholder request it satisfies. (This is your traceability matrix.)
3. Write a Product Position Statement.
4. Refine your Vision. (optional)
   - Brief summary.
   - Problem statement.
   - Product position statement.
   - Stakeholders and users.
   - Stakeholder and user needs.
   - Capture the features in Section 5 of the Vision document.
   - Constraints.

As a whole class:

1. Compare your product position statements.

Things To Consider

- Are there any stakeholder requests that do not trace to one or more features? What should you do in this situation?
- How does the Product Position Statement differ from the Problem Statement? Why is it useful to have both of them in your Vision document?
- How does viewing the results of putting together a Vision document influence your perception of your project?

1. Product Position Statement (Section 2.3 of the Vision)

Complete Section 2.3: Product Position Statement in your Vision document.

The product position statement is a tool for communicating the intent of the application and the importance of the project to all concerned personnel.

Complete the product position statement table for your class project. Refer to the Vision document for further instructions.

Below is an example to help you get started.

Product Position Statement	
For	Rational Software's existing and prospective customers
Who	need to get timely solutions to their technical problems while using our products
The TSXWEB	is a Technical Support External Web Site
That	brings the latest technologies to the Web and allows for 24X7 technical support, from finding solutions to participating in case activity, in a self-help Web environment, thus passing the control to the customer.
Unlike	the traditional non-Web alternative,
our product	will provide easy-to-use, accessible, online product support that is not dependent on human support personnel.

This Page Is Intentionally Left Blank

Exercise 6.2
Identify the Use Cases

The purpose of this exercise is to apply what you have learned about identifying actors and use cases to the class project.

Objectives

In this exercise, complete the following tasks:

- Identify the use cases for the system.
- Create a brief description of the selected use cases.
- Create a use-case diagram for the system.

Scenario

As the project lead, you want to understand the system from the user's perspective; what the users want to be able to do using the system. Review the information you have gathered thus far. From the information gathered, develop a use-case model of the requirements. In this part of the use-case modeling process, identify use cases, write a brief description for each use case, and develop a use-case diagram. In subsequent exercises, you will create a detailed description for the use cases.

As you look at the Initial Requests and other information about the system, note that the requirements are far from complete. Document your assumptions and any other information you want to ask your customer.

Directions

Work in small groups.

As you begin this exercise, use the information gathered earlier to help complete Step 1.

- Problem description (Exercise 4.1).
- Stakeholder requests (refer to page 46).
- Vision document (Exercise 6.1).
- Actors (Exercise 4.2).

1. Discuss the requirements, revise the actors, and identify use cases.
2. Write a brief description for one to two use cases. (Pick use cases that you think are the most important.)
3. Draw the use-case diagram on easel paper or the board.
   - Show actors, use cases, and communicates-associations.
4. Post the use-case diagram.

As a whole class:

1. Present each solution.
2. Compare the use-case diagrams and brief descriptions from the different groups.
3. Compare with the sample solution if using the RU e-st project.
   - See RUCS4: Use-Case Model Survey.
4. Cover the "Discussion Topics" at the end of this exercise.

Review Initial System Requests for Online Stock Trading System

See page 46

Identify Use Cases

What are the goals of each actor?

- What will the actor use the system for? Will the actor create, store, change, remove, or read data in the system? **If so, why?**
- Will the actor need to inform the system about external events or changes?
- Will the actor need to be informed about certain occurrences in the system?

Does the system supply the business with all of the correct behavior?

Draw the Use-Case Diagram

Draw a use-case diagram for the class project.

Discussion Topics

How does your solution compare with the sample solution? What is different? What is the same?

Are the use cases too small? Should some use cases be combined?

Does the brief description provide a good overview of the use cases purpose? Is there too much detail, not enough detail, or just enough detail?

Does the diagram cover all activities? Can you think of any activities not covered?

How do you know what is done in each use case?

Does your use-case model satisfy the goals of all stakeholders?

Did you identify any new actors?

RU e-st System Specification

Note: The sample in this handout contains only pages 1 through 4 of the specification.

Disclaimer: This is a fictitious system and does not claim to be a model for a good Software Requirement Specification. It is used for exercise purposes only.

INTRODUCTION

RU Financial Services is beginning an e-commerce project to develop software for a Web-based stock and securities trading system. The new system has been tentatively named the **RU e-st** system.

Purpose

The purpose of this SRS is to serve as a statement of understanding between the Online Trading Business and Engineering Divisions of RU Financial Services. Any desired departures from the specifications described herein must be negotiated by the affected parties.

The audience for this SRS includes the engineers who are involved in the development of the system, the marketers who are involved in its advertisement and sale, and the testers who validate the system.

Scope

The overall objective of the **RU e-st** Stock Trading System is to provide our trading customers with a convenient way to manage their stock portfolios. Managing a stock portfolio includes such capabilities as buying and selling and researching stock in a secure environment.

REQUIREMENTS

In this section, all of the functions that the **RU e-st** Stock Trading System Software is to perform are specified. These specifications are first given from total system perspective.

The **RU e-st** system shall provide all of the current functionality of the current stock trading system, plus any additions specified in this document.

Trading Customer Requirements

The **RU e-st** system allows its users to trade securities online, using a Web interface (an existing browser, such as Netscape or Internet Explorer). A trading customer can log on to the Internet anywhere (hotel room, airport, and so on.) Users are able to perform all the basic operations for securities trading: open accounts, trade stocks and other securities, and transfer assets among RU e-st accounts. Users can also obtain current and historical information about their accounts, such as the number of shares held, the current price, and the account total value.

Customers are able to execute many types of trade orders: market trading (buy and sell at current market prices), transfers from one mutual fund to another within one

account, and limit trading (buy and sell at a specific price). The **RU e-st** system uses the existing Market Trading System to perform the securities trades. **RU e-st** also allows users to transfer cash in an account to or from financial institutions such as banks or credit card vendors, using the existing Financial Network System. The usual restrictions apply:

- For online market sales, the securities to sell must be in the customer's account.
- For market purchases, cash for the purchase must be in the account by the settlement date.
- Transfers and purchases from an account are allowed only if, at the time the transaction occurs, they have enough cash to fund the transaction in the account.

Trading customers are also able to obtain information about what is happening in the securities markets. A trading customer can obtain price quotes and news headlines by entering the trading symbol (for example, IBM) for a particular security. The **RU e-st** system obtains current and historical quotes from the existing Quote System service and news items from the existing News System. **RU e-st** also has a feature to broadcast news headlines periodically on the trading customer's screen without being asked.

Regulatory Requirements

The system must report yearly tax information to the IRS and state tax boards. Tax forms must be communicated to the IRS and copies mailed to the customer. The information is also available online for customer viewing.

System Administration Requirements

Updates to the information on the Web pages must be possible without making the system unavailable.

Development Requirements

The system must be written in one of the company's approved programming languages. Refer to document Enterprise Architecture and Development Standards document EA-1234[1] for details.

Hardware Requirements

The hardware platform must be one supported by the enterprise hardware maintenance contract. Supported platforms are specified in the Enterprise Architecture and Development Standards document EA-1234.

[1] This document is not supplied with the course.

Exercise 6.3
Outline the Use Cases

Write a step-by-step outline for the flow of events in each selected project use case.

The purpose of this exercise is to practice outlining use cases to help understand what functions are contained in them.

Objectives

In this exercise, complete the following tasks:

- Create a step-by-step outline of the flow of events for the selected use cases.
- Create a list of scenarios for the use cases.

Scenario

Your job as system analyst for the new system is progressing nicely. The stakeholders have accepted your use-case diagram.

1. Create a step-by-step outline of the flow of events for use cases from the use-case diagram of your new system.
2. Create a scenario list for each use case.

Directions

Work in small groups.

1. Focus on the use cases selected for you to outline.
   - Review your use-case diagram (developed in Exercise 6.2.)
2. Create a step-by-step outline for the selected use cases.
   - First, outline the flow of events for the basic flow.
   - Second, identify three alternative flows.
   - Write the outline neatly on lined paper or a flip chart.
   - If possible, make copies of the outline(s) for the entire class.
3. Enumerate the scenarios for the use case. (Every flow should appear in at least one scenario.)
4. If time permits, outline another selected use case and enumerate the scenarios.

As a whole class:

1. Present each solution.
2. Compare the solutions between the different groups and the sample at the end of this exercise.

   If using the RU e-st project, see RUCS5: Step-by-Step Outlines.
3. Cover the “Discussion Topics” at the end of this exercise.

Write a Step-by-Step Outline

Use Case Name:

Brief Description:

Basic Flow

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

Alternative Flows

A1

A2

A3

Scenarios

S1

S2

S3

Write a Step-by-Step Outline

Use Case Name:

Brief Description:

Basic Flow

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

Alternative Flows

A1

A2

A3

Scenarios

S1

S2

S3

Discussion Topics

Do all the steps in your use-case outline reflect interactions with the system? Are there some steps that are done manually outside the system and do not belong in the use case?

Does your use-case outline show the basic flow from the beginning until the goal is achieved?

Does each flow appear in at least one scenario?

Now that you see the outline of the Get Quote use case, do you think it should be combined with the Execute Trade use case?

- Are they so related that they really belong together? If so, what would the combined use case be called? What would it look like?
- If you think they should not be combined, why might you want to have them separate?

Exercise 7.1
Prioritize Requirements Using Attributes

The purpose of this exercise is to explore how you might use attributes of feature requirements to make a decision about the relative importance of each and to determine what to cut when managing scope.

Each group should come up with a ranking that shows the order in which the tasks should be considered for inclusion in the release of the product, using the following input (attributes):

- The text of the requirement
- Priority (input from the customer)
- Difficulty (input from development)
- Risk (to the project)
- Stability (of the requirement)

Objectives

In this exercise, complete the following tasks:

- Review the feature matrix and attributes.
- Determine the relative importance of the attributes.
- Prioritize the list of features.
- Decide on baseline scope of features.

Scenario

You are the Vice President of Engineering for the product. The Development Manager (the instructor) has just told you that you only have the resources to accomplish two-thirds of the work shown on the feature list.

You have to catch a cab to the airport in 20 minutes, so you have no time to go back to talk with the customer. What requirements would you recommend for the baseline scope?

Directions

Work together in a small group.

1. Review the features matrix worksheet on the next page.
2. Discuss the relative importance of the attributes.
3. Rank the features based on their attributes.
   - The text of the requirement
   - Priority (input from the customer)
   - Difficulty (input from development)
   - Risk (to the project)
   - Stability (of the requirement)
4. List the feature numbers in the order of priority on chart paper or a transparency.

As a whole class:

1. Compare the feature rankings from the different groups.
2. Discuss reasoning.
3. Cover the “Discussion Topics” at the end of this exercise.

Using Attributes in Managing Scope

Requirements	Customer Priority	Difficulty	Risk	Stability	Rank
FEAT1: The system uses the Financial Services Network to enable transfer of funds between other financial institutions and the customers Trading Account.	Desired	Medium	Medium	High	
FEAT2: The system provides automatic reporting of tax-related information to the IRS and state tax boards.	Mandatory	Medium	Medium	High	
FEAT3: The system supports the most common browsers: Microsoft Internet Explorer, (version 4 and higher) and Netscape Navigator, (versions 4.72 through 4.75 and 6.2).	Mandatory	Low	Low	High	
FEAT4: The system allows funds to be transferred to and from Mutual Fund accounts.	Desired	Medium	Low	High	
FEAT5: The system allows a customer to operate multiple Trading Accounts with a single login.	Desired	Medium	Low	Medium	
FEAT6: The system can be upgraded without taking services off-line.	Desired	High	High	Medium	
FEAT7: During upgrades the system will preserve all "in-progress" transactions to ensure an error free experience for the customer.	Mandatory	High	Medium	High	
FEAT8: The system uses the Market Trading System to allow trading in any market worldwide.	Mandatory	Medium	High	Medium	
FEAT9: The system provides an FAQ page to answer customer's most frequent questions quickly and simply.	Desired	Low	Low	Low	
FEAT10: The system provides an "I need help now" capability that will open a "chat window" with the next available customer support representative. If there is no support personal available the system will inform the customer where they are in the queue for help.	Optional	High	Medium	Medium	
FEAT11: All Web pages shall be encrypted using 128 Bit SSL Security.	Mandatory	Low	Medium	High	
FEAT12: All personal and financial information shall be stored on a separate system with a secure network connection between the systems.	Mandatory	High	High	Low	
FEAT13: The system provides comprehensive "how-to" documentation that is structured like a story of using the system for a particular purpose. For example, "How to perform a Limit Buy."	Desired	Low	Low	Medium	
FEAT14: The system runs on a corporate standard platform.	Mandatory	Low	Low	High	
FEAT15: The system provides electronic and printed statements of account activity, including profit and loss information, for yearly income tax reporting.	Mandatory	Medium	Low	Medium	
FEAT16: The system provides statements of Trading Account activity including transfers between Trading Accounts, Trade summaries including: ticket code, number of shares traded and trade price.	Mandatory	Medium	Low	Medium	
FEAT17: Transfer cash from one customer trading account to another.	Optional	Medium	Low	High	
FEAT18: The system executes transactions in an Internet comfortable speed (less than 3 seconds, not counting transmission times on a 56K modem).	Desired	High	High	Medium	
FEAT19: Charge to a credit card and place funds in a customer's trading account.	Desired	Medium	High	High	
FEAT20: Rollover from a retirement account in another institution in to a retirement account managed by the system.	Desired	High	High	Low	
FEAT21: The system allows Market Buy and Sell transaction types.	Mandatory	High	Medium	High	
FEAT22: The system allows Limit Buy and Sell transaction types.	Mandatory	Medium	Medium	High	
FEAT23: The system allows mutual fund management.	Desired	Medium	Medium	High	
FEAT24: The system allows stock research and alerts - online quotes, news flashes, and so on.	Desired	High	Medium	Low	

Discussion Topics

What method did you use to get the results?

What are the advantages of using a formula?

What different assumptions or methods led to the different results?

What other information would you like to help make your decisions?

How do you set scope now in your own organization?

Exercise 7.2
Prioritize Scenarios

The purpose of this exercise is to identify which scenarios you want to detail in this iteration. You should use your prioritized list of feature requirements to make a decision.

As a whole class, consider the ranking of each feature and describe the scenario it traces to.

Objectives

In this exercise, complete the following tasks:

- Review the prioritized list of features.
- Trace the features into each scenario and use that to list the scenarios that you detail in the next module.

Scenario

Based on the prioritized list of features from the previous exercise, list the scenarios that should be detailed and implemented in this iteration.

For the purposes of this exercise, imagine you are in the first iteration of the *Construction Phase* of the Rational Unified Process®. This means that all architectural risks have been mitigated and that you are now focusing on removing the risks related to not delivering system functionality that is important to the customer, as well as getting the bulk of the work done.

Directions

Work together as a class.

1. Review the prioritized feature requirements form the previous exercise.
2. Use the use-case outlines provided at the end of this exercise (page 61) to trace the features to flows.
3. For each flow that is traced to, identify a scenario that the flow is part of. When identifying the scenario, consider the Analyst's criteria for selecting scenarios.
4. List the scenarios to detail in the next exercise.

Discussion Topics

Were there any features that you could not trace to a flow? If so, what should you do about that?

Were there any flows that were not traced from a feature? If so, what does this mean?

Were there any use cases that had no flows traced from a feature? So, what does this mean?

Exercise 8.1
Choose a Style

Look at the styles of writing a flow of events and compare them. Each style has different benefits.

The way to describe the flow of events depends on factors, such as:

- **Who is the reader?**

  This may differ from project to project. Sometimes you may have a "customer" of the system, or at other times you need to try to understand what the market wants without having a “customer.” Depending on who will be reading the flow of events, there are many different ways to write it. If the readers are unfamiliar with the system and the Rational Unified Process®, you should describe it in one way. On the other hand, if the systems are well known to all readers, you may choose to describe it in another way. Regardless of the way you choose to describe it, be consistent for all use cases in your project.

- **Who is the author?**
  It is not easy to make everyone in a project team write the flow of events in the same way. The style depends on many factors, including the skills and background of the authors. For example, if some people have problems expressing themselves, it may be easier to make a more formal template for the flow of events. Your use-case modeling guidelines can help ensure that there is a consistent style.

Objectives

In this exercise, complete the following tasks:

- Review different flow of events descriptions.
- Compare and contrast the different styles.
- Recommend a style for your project team.

Scenario

After you have finished detailing your flow of events, your co-worker Bob mentions that he has seen use cases written in other styles. He thinks that the team should decide what style they want to use for their use cases. He proposes some sample styles to choose from.

Directions

Working by yourself:

Review each of the following samples of a flow of events that are written in different styles. As you read each style, consider the following:

Consider the following questions when comparing the use cases:

- Is it easy to read and understand?
- Do you think it would be easy to create (write) with a word processor?
- Do you think it would be easy to maintain if you were inserting and reorganizing the flows?
- In what ways is it better than the others?
- In what ways is it worse than the others?
- Which style do you prefer?

As a whole class:

- Compare the styles of flow of events, using the questions above.
- Recommend a style for your project team.

Style I–Table

1. Use Case Name: Get Quote

1.1 Brief Description

The Trading Customer can get current and historical information about the trading price of securities.

2. Flow of Events

2.1 Basic Flow

Step	*Trading Customer*	*System*	*Quote System*
1	The use case starts when the Trading Customer logs on.		
2		The system validates the customer id and password. The system presents a list of available functions.	
3	The Trading Customer chooses to "Get Quote."		
4		The system displays the list of trading symbols and names of securities on which it has quotes.	
5	The Trading Customer selects from the list of securities or enters the trading symbol for a security.		
6		The system sends the trading symbol to the Quote System	
7			The Quote System returns the most recent quote for the requested trading symbol.
8		The system presents the corresponding Quote Display (see the fields to display in Supplementary Specifications) to the Trading Customer.	
9	The Trading Customer logs off the system. The use case ends.		

2.2 Alternative Flows

2.2.1 Get Additional Quotes

Step	*Trading Customer*	*System*	*Quote System*
1		At the end of step 8 of the Basic Flow, if the Trading Customer wishes to get additional quotes, the use case resumes at Step 5 in the Basic Flow.	

2.2.2 Unidentified Trading Customer

Step	Trading Customer	System	Quote System
1		In Step 1 in the Basic Flow, if the system determines that the customer id or password are not valid, an error message is displayed. The use case ends.	

2.2.3 Quote System Unavailable

Step	Trading Customer	System	Quote System
1		In Step 3 in the Basic Flow, if the system is unable to communicate with the Quote System, the system informs the Trading Customer. The use case ends.	

2.2.4 Quit

The RU e-st System allows the Trading Customer to quit at any time during the use case. The use case ends.

Step	Trading Customer	System	Quote System
1	The RU e-st System allows the Trading Customer to quit at any time during the use case. The use case ends.		

2.2.5 Unknown Trading Symbol

Step	Trading Customer	System	Quote System
1		In Step 6 in the Basic Flow, if the system cannot recognize the trading symbol, the system notifies the Trading Customer that the trading symbol is not recognizable. The use case continues at the beginning of Step 5 in the Basic Flow.	

2.2.6 Quote System Cannot Locate Information

Step	Trading Customer	System	Quote System
1		In Step 8 in the Basic Flow, if the Quote System responds that it does not have the requested information, the system notifies the Trading Customer that the Quote System does not have information on the requested security. The use case continues at the beginning of Step 5 in the Basic Flow.	

Style II-Bulleted

1. Use Case Name: Get Quote

1.1 Brief Description

The Trading Customer can get current and historical information about the trading price of securities.

2. Flow of Events

2.1 Basic Flow

- Customer Logs On.
  - The use case starts when the Trading Customer logs on. The system validates the customer id and password. The system presents a list of available functions.
- Customer Selects "Get Quote" Function.
  - The Trading Customer chooses to "Get Quote." The system displays the list of trading symbols and names of securities on which it has quotes.
- Customer Gets Quote.
  - The Trading Customer selects from the list of securities or enters the trading symbol for a security. The system sends the trading symbol to the Quote System and receives the Quote System Response. The system presents the corresponding Quote Display (see fields to display in Supplementary Specifications) to the Trading Customer.
- Customer Logs Off.
  - The Trading Customer logs off the system. The use case ends.

2.2 Alternative Flows

2.2.1 Get Additional Quotes

At the end of Customer Gets Quote in the Basic Flow, if the Trading Customer wishes to get additional quotes, the use case resumes at *Customer Gets Quote* in the Basic Flow.

2.2.2 Unidentified Trading Customer

In Customer Logs On in the Basic Flow, if the system determines that the customer id or password are not valid, an error message is displayed. The use case ends.

2.2.3 Quote System Unavailable

In Customer Gets Quote in the Basic Flow, if the system is unable to communicate with the Quote System, the system informs the Trading Customer. The use case ends.

2.2.4 Quit

The RU e-st System allows the Trading Customer to quit at any time during the use case. The use case ends.

2.2.5 Unknown Trading Symbol

In Customer Gets Quote in the Basic Flow, if the system cannot recognize the trading symbol, the system notifies the Trading Customer that the trading symbol is not recognizable. The use case continues at the beginning of Customer Gets Quote in the Basic Flow.

2.2.6 Quote System Cannot Locate Information

In Customer Gets Quote, in the Basic Flow, if the Quote System responds that it does not have the requested information, the system notifies the Trading Customer that the Quote System does not have information on the requested security. The use case continues at the beginning of Customer Gets Quote in the Basic Flow.

Style III-RUP

1. Use Case Name: Get Quote

1.1 Brief Description

The Trading Customer can get current and historical information about the trading price of securities.

2. Flow of Events

2.1 Basic Flow

1. Customer Logs On.

The use case starts when the Trading Customer logs on. The system validates the customer id and password. The system presents a list of available functions.

2. Customer Selects "Get Quote" Function.

The Trading Customer chooses to "Get Quote. " The system displays the list of trading symbols and names of securities on which it has quotes.

3. Customer Gets Quote.

The Trading Customer selects from the list of securities or enters the trading symbol for a security. The system sends the trading symbol to the Quote System and receives the Quote System Response. The system presents the corresponding Quote Display (see fields to display in Supplementary Specifications) to the Trading Customer.

5. Customer Logs Off .

The Trading Customer logs off the system. The use case ends.

2.2 Alternative Flows

2.2.1 Get Additional Quotes

At the end of step 3, Customer Gets Quote, if the Trading Customer wishes to get additional quotes, the use case resumes at step 3 in the Basic Flow.

2.2.2 Unidentified Trading Customer

In Step 1, Customer Logs On, in the Basic Flow, if the system determines that the customer id or password are not valid, an error message is displayed. The use case ends.

2.2.3 Quote System Unavailable

In Step 3, Customer Gets Quote, in the Basic Flow, if the system is unable to communicate with the Quote System, the system informs the Trading Customer. The use case ends.

2.2.4 Quit

The RU e-st System allows the Trading Customer to quit at any time during the use case. The use case ends.

2.2.5 Unknown Trading Symbol

In Step 3, Customer Gets Quote, in the Basic Flow, if the system cannot recognize the trading symbol, the system notifies the Trading Customer that the trading symbol is not recognizable. The use case continues at the beginning of Step 3 in the Basic Flow.

2.2.6 Quote System Cannot Locate Information

In Step 3, Customer Gets Quote, in the Basic Flow, if the Quote System responds that it does not have the requested information, the system notifies the Trading Customer that the Quote System does not have information on the requested security. The use case continues at the beginning of Step 3 in the Basic Flow.

Style IV-Tag

1. Use Case Name: Get Quote

1.1 Brief Description

The Trading Customer can get current and historical information about the trading price of securities.

2. Flow of Events

2.1 Basic Flow

{Trading Customer logs on}

1. The use case starts when the Trading Customer logs on.
2. The system validates the customer id and password. The system presents a list of available functions.
3. The Trading Customer chooses to "Get Quote."
4. The system displays the list of trading symbols and names of securities on which it has quotes.

{Customer Gets Quote}

5. The Trading Customer selects from the list of securities or enters the trading symbol for a security.

{Request Quote from Quote System}

6. The system sends the trading symbol to the Quote System and receives the Quote System Response.
7. The system presents the corresponding Quote Display (see fields to display in Supplementary Specifications) to the Trading Customer.

{Log Off}

8. The Trading Customer logs off the system. The use case ends.

2.2 Alternative Flows

2.2.1 Get Additional Quotes

In **{Log Off}**, if the Trading Customer wishes to get additional quotes, the use case resumes at **{Customer Gets Quote}**.

2.2.2 Unidentified Trading Customer

In **{Trading Customer logs on}** if the system determines that the customer id or password are not valid, an error message is displayed. The use case ends.

2.2.3 Quote System Unavailable

In **{Request Quote from Quote System}** if the system is unable to communicate with the Quote System, the system informs the Trading Customer. The use case ends.

2.2.4 Quit

The RU e-st System allows the Trading Customer to quit at any time during the use case. The use case ends.

2.2.5 Unknown Trading Symbol

In **{Customer Gets Quote}** if the system cannot recognize the trading symbol, the system notifies the Trading Customer that the trading symbol is not recognizable. The use case continues at **{Customer Gets Quote}**.

2.2.6 Quote System Cannot Locate Information

In **{Request Quote from Quote System}** if the Quote System responds that it does not have the requested information, the system notifies the Trading Customer that the Quote System does not have information on the requested security. The use case continues at **{Customer Gets Quote}**.

Style V–Pseudo Code

1. Use Case Name: Get Quote

1.1 Brief Description

The Trading Customer can get current and historical information about the trading price of securities.

2. Flow of Events

2.1 Basic Flow

```
<= 'Show log in menu'
=> 'Log in' (User name, Password)
IF the user name or password are incorrect
<= 'Show log in info incorrect message'
   The use case ends
END IF

=> 'Get Quote'
DO
   <= 'Show list of tickers (trading symbols) and names of securities'
   => 'Select or enter ticker'
   IF the Trading Customer chooses to quit
    The use case ends
   ENDIF
   The system sends the quote request to the Quote System.
   IF the Quote System responds
      IF the Quote System found the Ticker
            <= 'Show the Quote Details'
      ELSE IF the Quote System responds that it does not recognize the Ticker
       <= 'Show Unknown Ticker message'
      ELSE IF the Quote System responds that it does not have the information
       <= 'Show Information Currently Unavailable message'
      ENDIF
   ELSE
    <= 'Show Quote System Unavailable message'
    The use case ends
   ENDIF
WHILE => 'Get Additional Quotes'
The use case ends.
```

Exercise 8.2

Detail the Flows

The purpose of this exercise is to apply what you have learned about detailing use cases. In this exercise, you further detail the use cases outlined in Exercise 6.3 and prioritized in Exercise 7.2.

Use the Get Quote Use Case Report below as an example of a fully detailed use case. Unlike your use cases for this exercise, every flow in this use case has been detailed. Notice the level of detail in this example. Remember, you need to describe every detail that the customer wants the developers to know.

Objectives

In this exercise, complete the following tasks:

- Identify the scenarios to be detailed in this iteration (from module 7).
- Identify the flows for each scenario to be detailed.
- Add detail to the step-by-step outline of the basic flow of events.
- Describe clearly what happens in each alternative flow.
- Revise the use-case scenario list
- Identify preconditions.
- Identify postconditions.

Scenario

Your requirements for the new system are becoming clear. The stakeholders have accepted your use-case outlines.

Now fill out the step-by-step outline with more detail to the basic and the alternative flows. As the flows are detailed it is likely that additional alternative flows are identified. As additional flows are identified the scenario list should be updated to include them.

As part of detailing use cases you also write a preconditions and postconditions.

Directions

Work together in a small group:

1. Focus on the scenarios you selected to detail in module 7.
   - Review your use case outline (developed in Exercise 6.3).
2. Detail the flow of events. (Use the sample on page 79 as a style guide.)
   - Give each step in the Basic Flow a title.
   - Write one to three sentences that detail each step.
   - Write the details of each Alternative Flow.
3. Revise the scenario list.
4. Add a precondition.
5. Add a postcondition.
6. Write the group's use case report neatly on lined paper.
   - If possible, make copies of the use case report(s) for the entire class.
7. If time permits, detail another selected use case.

As a whole class:

1. Present each solution.
2. Compare the solutions from the different groups.
3. Compare the solutions with the sample solution if using the RU e-st project.
   - See RUCS6: Use Case Reports.
4. Discuss the "Discussion Topics" at the end of this exercise.