Com S 362
Spring 2003

Name: _____

Object-Oriented Analysis and Design

Exam 2 on Requirements Analysis

This test has 6 questions and pages numbered 1 through 6.

Reminders

This test is open book and notes. However, it is to be done individually and you are not to exchange or share materials with other students during the test.

If you need more space, use the back of a page. Note when you do that on the front.

This test is timed. We will not grade your test if you try to take more than the time allowed. Therefore, before you begin, please take a moment to look over the entire test so that you can budget your time.

For diagrams and programs, clarity is important; if your diagrams or programs are sloppy and hard to read, you will lose points. Correct syntax also makes some difference.

1. (5 points) You are working in a company that is doing time boxed, iterative development. The day before the end of the current iteration, you realize that your part of the project will not be completed on time. Should you:

   (a) Tell your manager that you need an extension of the deadline for your part of the project, to take into account the time needed to complete the functionality you were supposed to implement.

   (b) Tell your manager that you need to reduce the scope of your part of the project by eliminating some of the functionality you were supposed to implement.

   (c) Tell your manager that you need to go to New Orleans for a vacation.

   Which of these alternatives best fits with time boxed iterative development? Pick one (and only one) of these alternatives, and give a brief explanation of why this is the best alternative.

2. (12 points) Consider an e-mail client system (such as Pine, Eudora, or Outlook); that is, a system that can read and send e-mail for users, and manage their e-mail files and addresses. For each of the following descriptions, you are to decide whether it satisfies the elementary business process (EBP) guidelines. Write "yes" or "no" (but not both) and give a brief description of why this does or does not follow the EBP guidelines.

   (a) Exchange several e-mail messages in a conversation between the User and one or more other people.

   (b) In the list of the e-mail messages sent to the User, change the selected e-mail message to the previous message.

   (c) Read an e-mail message (i.e., present it to the User).

3. (10 points) Briefly describe one *non-functional* requirement for an e-mail client system.

4. (13 points) Consider a simple e-mail client system (such as Pine, Eudora, or Outlook); that is, a system that can read and send e-mail for users, and manage their e-mail files and addresses. Using the brief format, write a use case titled "Send Single Recipient E-mail," which would involve a User composing an e-mail message, with a subject and body, and sending it to one person. (That is, no one else is to receive the message.)

   You should write your use case for a straightforward e-mail client, without lots of embellishments or fancy features. Keep it simple.

   Be sure your use case is written in an essential, UI-free style. You only need to consider the "happy path."

5. (25 points)

   Consider again a simple e-mail client system. Using the casual format, write a use case titled "Reply to E-mail Message," in which the User answers an e-mail message. In the main success scenario, assume that the e-mail message has just one recipient, namely the User, and that the User wishes to reply only to the sender of the e-mail message (i.e., to no one else). Also, in the main success scenario, assume that the sender has a legitimate network address.

   To save time, in addition to the main success scenario, just write two alternate scenarios: (a) what happens if the User decides that, after composing the reply, they do not wish to send it after all, and (b) what happens if the e-mail address of the sender of the original message is not legitimate (i.e., it does not exist on the network).

   Be sure your use case is written in an essential, UI-free style.

6. (35 points) Consider again a simple e-mail client system. Using the fully-dressed format, write a use case titled "Save Attachments," in which the User saves one or more, but not necessarily all, of the attachments in an e-mail message to files (on their computer). (An *attachment* is the contents of a file that has been transmitted inside an e-mail message in a format that allows an e-mail client system to extract its contents.) In the main success scenario, assume that the e-mail message has several attachments all of which can be successfully saved to files.

   To save time, in addition to the main success scenario, just write two alternate scenarios. These should describe (a) what happens if the file in which the User wishes to save an attachment is not writable, and (b) what happens if an input/output error occurs during the writing of an attachment to disk (e.g., the disk is full). Within each of these alternative scenarios you need only consider the "happy path" (i.e., you don't have to consider alternatives to the alternatives).

   Also, you can leave the "Technology and Data Variations List" section empty. You can consider failures and recovery to be an open issue.

   Be sure your use case is written in an essential, UI-free style.

   There is more space for your answer on the next page.