

Exercises¹

► Exercise 1.

Write use case descriptions for the following activities:

- (a) Paying a bill at an automatic teller machine.
- (b) Creating a table in a word processor.
- (c) Programming a microwave oven to turn on in five hours and heat some food.
- (d) Read messages in a voice mail system.

► Exercise 2.

Write the following use cases, which are related to the 'Exit car park, paying cash' use case of Figure 2.

- (a) Exit car park by paying using a debit card.
- (b) Attempt to exit car park without initially entering enough money.
- (c) Exit car park.

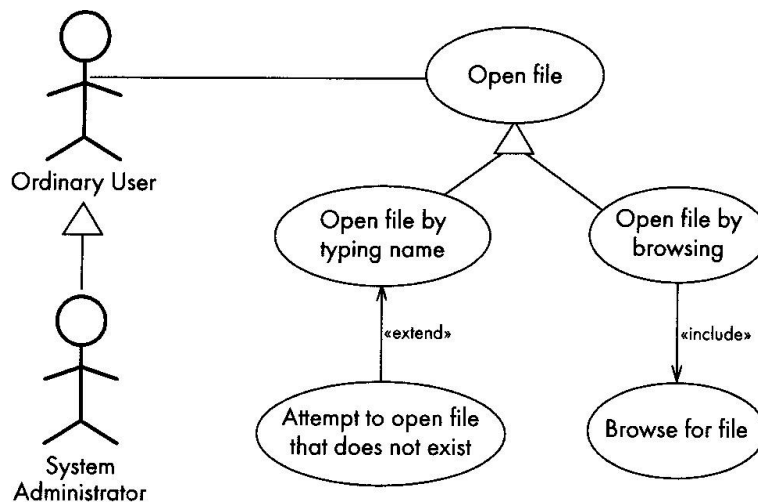


Figure 2: Extension, specialization and inclusion in use cases.

► Exercise 3.

¹for Chapter 7 from the 'Iloseng' book.

Draw a use case diagram showing the relationships among the use cases of the last exercise.

► **Exercise 4.**

For the following systems, create a list of use cases, and then identify the central ones, and other ones that should be given high priority.

- (a) A telephone answering machine.
- (b) The GANA navigation system.
- (c) A video-cassette recorder.

► **Exercise 5.**

Describe the error messages that will need to be displayed in each of the panels shown in Figure 3.

► **Exercise 6.**

Web sites often contain pages that take too long to download. Using a modem-based Internet connection, visit the web site of your university or company and check whether the response time is adequate.

► **Exercise 7.**

Use the twelve principles discussed in this section to guide you in the following. Prepare a paper prototype of a small user interface of your choice. Describe how you have adhered to each of the twelve usability principles.

► **Exercise 8.**

Draw a paper prototype of the user interface for the GANA system, whose requirements were presented in Chapter 4. Base your prototype on the textual description of the UI given in the requirements document, and the use cases you developed in Exercise 4. You will have to fill in some details not explicit in the textual description, such as the size and shape of buttons etc.

► **Exercise 9.**

Find an application that performs each of the following tasks. Perform a heuristic evaluation based on each task in order to find the situations where it violates the twelve usability principles described in the last section. Describe each of the defects you find, and suggest how it could be fixed.

(a) A facility for drawing a graph, in a spreadsheet or statistical application. Evaluate changing the graph format (e.g. scatter, bar or line), changing x- and y-axis labels, changing the scale of the axes and adding extra data points.

(b) Facilities for creating a table in a word processor. Evaluate converting text into a table, balancing the widths of columns and making the format of a table look like those in this book.

► **Exercise 10.**

Perform a heuristic evaluation of a web site or some freeware that you download. Choose a system that has some interesting complexity, e.g. one that has some forms to fill in, and at least five different pages or dialogues. You could choose a small part of a larger system if you prefer.

► **Exercise 11.**

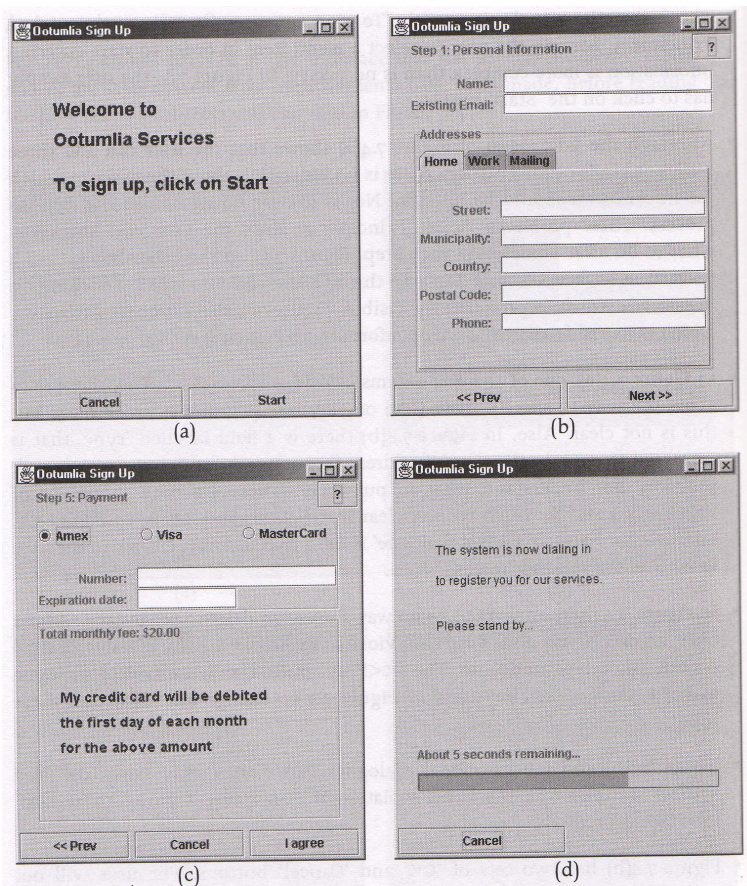


Figure 3: Views of the ISP sign-up application. It uses a 'wizard' approach as well as tabbed dialogues.

Work in groups of two or three to do the following. First, each member of the group independently performs a heuristic evaluation of the paper prototype you developed in Exercise 7. Then the group members get together and study each other's lists of defects. Determine how many were found by only one person, how many were found by two people and how many were found by all three (if you have three members). This exercise should demonstrate that having more than one evaluator is important.

► **Exercise 12.**

Perform a heuristic evaluation of the GANA UI that you drew in Exercise 8. Update your GUI if necessary; also, indicate any requirements that should be changed based on your LTI review.

1 Projects

► **Project 1.**

Develop a set of use cases for the SimpleChat system. These use cases should be based on the requirements you specified in the first exercise of Chapter 4.

► **Project 2.**

Working in teams of three or four, develop a graphical user interface for the client side of Phase 4 of the SimpleChat system (the version developed in the project exercises of the last chapter). The result of your work will be Phase 5. Use the guidelines presented in this chapter. In particular, undertake the following activities:

(a) Have each person in the team independently create a paper prototype that contains his or her best ideas about how the user interface should appear. Doing this should take no more than about half an hour.

(b) Have each person present his or her paper prototype to the other members of the group. The group members first should point out the good ideas of the prototype, and then should give constructive criticism. The creator of the prototype should write down the ideas.

(c) Working together, the group should then take the best ideas from the paper prototypes and develop a new unified paper prototype. Be careful, however, not to add too many features.

(d) Now, using AWT or a GUI building tool, develop Java code to implement your interface, replacing the ConsoleChat class.

(e) Evaluate your user interface by arranging for several members of some other group to use it. Take note of any problems they encounter, and update your interface to reduce the problems.

► **Project 3.**

On the book's web site, you will find a class called DrawPad. Associated with this there is a class called StartDraw that allows you to create an instance of DrawPad and draw some simple lines and curves using the mouse. Without changing DrawPad at all, add a feature to SimpleChat that will allow you to communicate with other users of SimpleChat by drawing pictures. This would work by having two clients simply open DrawPads and start drawing. The

drawing actions should be transmitted as commands via the server. The result of your work will be Phase 6 of SimpleChat.

► **Project 4.**

The SimpleChat system is not internationalized. Its most severe problem is that all the displayed messages are hard-coded in English. Design and implement an internationalized version of SimpleChat that obtains all the displayed messages from a message file. Arrange for the system to determine the locale from the operating system, and then select an appropriate message file. Hint: you can number each message (but do not display numbers to end-users). You will need to develop a way to embed parameters in messages.

► **Project 5.**

Develop a complete set of use cases for the Small Hotel Reservation System.

► **Project 6.**

Develop a prototype user interface for the Small Hotel Reservation System, as follows. If possible, involve users from the hotel industry.

(a) Do some parallel design of paper prototypes. Then pick the best features of each paper prototype to create a final version.

(b) Review your prototype, following the guidelines discussed in this chapter. Change it as necessary.

(c) Use a rapid prototyping tool to create a partially functional prototype based on your paper prototype. This prototype will not actually manipulate any data, so it will not make use of the object model you have developed.

(d) Have people use your prototype. Observe the difficulties they have. Change the prototype as necessary.

► **Project 7.**

If your experiences with the last exercises have taught you anything about the requirements, update your requirements for the Small Hotel Reservation System, and also update your use cases.