

# **SENG2130/6350 Software Development**

## **User Interface**

### **Workshop week 11**

#### **Exercise 1**

Consider a file system with a graphical user interface, such as Macintosh's Finder, Microsoft's Windows Explorer, or Linux's KDE. The following objects were identified from a use case describing how to copy a file from an USB disk to a hard disk: File, Icon, TrashCan, Folder, Disk, Pointer. Specify which are entity objects, which are boundary objects, and which are control objects.

## Exercise 2.

The following diagram is a screen dump of a (first iteration) prototype customer information input screen for a hotel registration system. Previously the guests were required to complete a registration card every time they booked into the hotel. This was annoying for regular customers.

Reservation Number:	<input type="text"/>	Date Arriving:	<input type="text"/>
Name:	<input type="text"/>	Number of Nights:	<input type="text"/>
Address:	<input type="text"/>	Special Request:	<div style="border: 1px solid black; height: 60px; width: 150px;"></div>
Telephone:	<input type="text"/>		
Date Booked:	<input type="text"/>	Taken by:	<input type="text"/>
Account:	<input type="text"/>	Package:	<input type="text"/>
Confirm:	<input type="checkbox"/>	Deposit Required:	<input type="checkbox"/>
Package:	<input type="text"/>	Deposit Due:	<input type="text"/>
Travel Agent:	<input type="text"/>	Deposit Received:	<input type="checkbox"/>

Along with a snapshot of the screen is a list of comments from staff involved in evaluating the new system.

Address the staff comments on prototype 1.

1. *"Name" box should be separate first name and last name for easier searching:* Probably a reasonable suggestion.
2. *Is 'taken by' where we are supposed to enter our own names? Stupid label:* If "taken by" means "enter the name of the staff member who took the booking", then it *is* a stupid name.
3. *"Confirm" should not be a check box. It should be a date:* Good suggestion. Confirmation would probably suggest a date. But does it mean that we agree to confirm, or that the customer should confirm the booking?
4. *The background looks stupid:* Irrelevant.
5. *Could you include a spell checker please:* Given that names of people and places are often not common words, this might be a nuisance.
6. *Where are supposed to enter the room number?:* Good point. Who forgot this?
7. *The special request box looks a bit small:* Is this reasonable? Extra text might simply result in a scroll bar appearing. Or is this not a good idea – information should appear on the screen.

8. *Package should be a drop box. We only have a few packages and it would not be hard to list them:* Probably a reasonable suggestion. Staff would have to know how to add extra packages.
9. *What if the customer wants more than one room? Are we supposed to fill out one screen for each:* Obviously impractical to fill out 1 per room. But what is the procedure.
10. *This will be slower than having customers fill out their own cards. Will we have extra staff to help?* Is this a fair comment? Is it likely to be faster?
11. *What happens if the system crashes? Or computer malfunctions?* Good questions, what is the answer?

**Consider the relevance of the staff responses and what action you would take (if any) in view of these responses. Then design the next iteration prototype of the registration page.**

### Approaches

1. Discuss how to address the staff comments. Some approaches are listed as following:

#### Poor Interface Design

#### Psychological Responses of Users

- Confusion: too much detail on screens
- Panic: long delays/response times, unless messages inform users (Loading ... Please Wait) users are unsure of what is happening or if the system is OK?
- Boredom: too much help detail for experienced users, slow progress through screen
- Frustration: Users cannot 'undo' a change or 'quit' the system, no system acknowledgement for a user's change

#### Physical Responses of Users

- Abandonment: use of the old system or other sources by managers
- Incomplete Use of System
- Indirect Use of System
- Misuse of System
- Task Modification
- Compensatory Activity
- Direct Reprogramming

#### Good vs. Bad Design

- Iterate – your attempt may work right, look great, but users may not be able to use it. That's why we iterate !
- Try it with a few users

2. Think of ways to improve the screen that may not have occurred to the users (eg. "Deposit Required" could be an amount box instead of a check box, so that instead of just indicating that a deposit is required, the screen could indicate how much deposit is required. This would also help with checking that the deposit, when paid, is sufficient.)

3. Think about designing a new prototype input screen keeping in mind the staffs' responses to prototype 1 and your own common sense. For instance, adding more text boxes could make the window seem cluttered, or the text too small, but scrolling or flipping between screens is also clumsy. What is the best way to incorporate more data?