

Software Design and Analysis

CS-3004

mistakes while writing use cases

Dr. Javaria Imtiaz

Assistant Professor

javaria.imtiaz@nu.edu.pk



quest lab

Good use cases are

- Text
- No GUI
- No data formats
- 3 – 9 steps in main scenario
- Easy to read
- At user's goal level
- A record of decisions made

Good use cases are aren't*

- describing the GUI
- describing data formats
- multiple-page main scenario
- complicated to read
- at program-feature level

Use cases add value to different project members at different times

1. The list of goal names provides executives:
 - Shortest summary of what system will contribute
 - Project planning skeleton (priorities & timing)
2. The main success scenario provides all:
 - Agreement as to the system's responsibilities
3. The extension conditions provide programmers:
 - List of things programmers have to watch for
 - List of things analysts have to investigate
4. The extension handling steps provide the development team:
 - Record of (tricky) business policy decisions

STANDARD MISTAKES (Withdraw Cash)

Scope: ATM

Level: User Goal

1. The card gets inserted.
2. The card information gets validated.
3. The transaction information gets collected and validated.
4. The cash is issued, card returned, cash removed, account debited, screen reset.

No customer or customer point of view

STANDARD MISTAKES (Withdraw Cash)

Scope: ATM

Level: User goal

1. Collects ATM card, PIN.
2. Collects transaction type as “Withdrawal”
3. Collects amount desired.
4. Validates that account has sufficient funds.
5. Dispenses money, receipt, card
6. Resets

again, just the system perspective: where is the customer actor

STANDARD MISTAKES (Withdraw Cash) (WEUC)

1. Customer runs ATM card through the card reader.
2. ATM reads the bank id and account number.
3. ATM asks customer whether to proceed in Spanish or English.
4. Customer selects language.
5. ATM asks for PIN number and to press Enter.
6. Customer enters PIN number, presses Enter.
7. ATM presents list of activities for the Customer to perform.
8. Customer selects "withdraw cash".
9. ATM asks customer to say how much to withdraw, in multiples of \$5, and to press Enter.
10. Customer enters an amount, a multiple of \$5, presses Enter.
11. ATM notifies main banking system of customer account, amount being withdrawn.
12. Main banking system accepts the withdrawal, tells ATM new balance.
13. ATM delivers the cash.
14. ATM asks whether customer would like a receipt.
15. Customer replies, yes.
16. ATM issues receipt showing new balance.
17. ATM logs the transaction.

too long

STANDARD MISTAKES [UI] (Buy Something) (WEUC-19.3)

Scope: Purchasing application

Level: User goal

1. System **presents** ID and Password screen.
 2. Customer types id and password into system, clicks OK.
 3. System **validates** user id and password, displays Personal Information Screen
 4. Customer types in first and last names, street address, city, state, zip code, phone number, and clicks OK.
 5. System **validates** that user is a known user.
 6. System **presents** available product list.
 7. Customer clicks on pictures of items to be purchased, types in quantity next to each, clicks on DONE when finished.
 8. System **validates** with the warehouse storage system that sufficient quantity of the requested product is in stock.
- ...etc.

UI specific

STANDARD MISTAKES [data] (Purchase Goods) (PEUC 6.2)

Level: User Goal

1. System asks User for first name.
2. User enters first name.
3. System asks for middle initial.
4. User enters middle initial.
5. System asks for last name.
6. User enters last name.
7. System asks for first line of street address.

...

[And so on—we'll spare you the details.]

way too specific wrt data to enter

:-) “Purchase Goods” (PEUC6.5)

Level: User Goal

1. Visitor enters customer information (name, address, etc.).
2. System retrieves customer's profile information, and presents product search and selection mechanisms.
3. Visitor selects products until satisfied. After each selection, system adds the selected product to the customer's shopping cart and presents the ongoing running total of products selected.
4. Visitor selects to purchase the selected items.
5. System presents contents of shopping cart and requests customer's payment information.
6. Visitor enters manner of payment and other payment details.
7. System presents visitor with final sums, charges credit card, and delivers packing list to the shipping department.

good UC: notice use of abstraction

STANDARD MISTAKES (Register for Courses) **(PEUC 1.1)**

1. Display a blank schedule.
2. Display a list of all classes in the following way: The left window lists all the courses in the system in alphabetical order. The lower window displays the times the highlighted course is available. The third window shows all the courses currently in the schedule.
3. Do
4. Student clicks on a course.
5. Update the lower window to show the times the course is available.
6. Student clicks on a course time and then clicks on the "Add Course" button.
7. Check if the Student has the necessary prerequisites and that the course offering is open.
8. If the course is open and the Student has the necessary prerequisites, add the Student to the course. Display the updated schedule showing the new course. If no, put up a message, "You are missing the prerequisites. Choose another course."
9. Mark the course offering as "enrolled" in the schedule.
10. End do when the Student clicks on "Save Schedule."
11. Save the schedule and return to the main selection screen.

UI specific, too long, no goal perspective

: -) “Register for Courses” (PEUC 1.3)

System: Course Enrollment System

Level: User Goal

1. Student requests to construct a schedule.
2. The system prepares a blank schedule form.
3. The system gets available courses from the Course Catalog System.
4. Student selects up to 4 primary course offerings and 2 alternate course offerings from the available offerings.
5. For each course, the system verifies that the Student has the necessary prerequisites, adds the Student to the course, marking Student as “enrolled” for that course in the schedule.
6. When the Student indicates the schedule is complete, the system saves the schedule.

Extensions:

- 1a. *Student already has a schedule:* System brings up the current version of the Student’s schedule for editing instead of creating a new one.
- 1b. *Current semester is closed and next semester is not yet open:* System lets Student look at existing schedules, but not create new ones.
- 3a. *Course Catalog System does not respond:* The system notifies the Student and terminates the use case.
- 5a. *Course full or Student has not fulfilled all prerequisites:* System disables selection of that course and notifies the Student.

SUBTLE MISTAKES (“Book Flight”) (PEUC 4.1)

Level: User Goal

1. This UC begins when a customer contacts the travel agency and requests a flight.
 2. The customer proxy captures the customer’s trip origin and destination.
 3. The customer proxy looks up the airport codes for the origin and destination.
 4. The customer proxy captures the preferred departure times for the customer.
 5. The customer proxy captures the customer’s preferred class of service.
 6. The system requests from the airline proxy a list of the flights available that match the customer’s preferences.
 7. The system retrieves the customer’s profile to get her preferred airline and frequent flyer number.
 8. The system sorts the list of available flights first by the customer’s airline preference and those that are closest to the customer’s preferred departure times.
- ... Etc., etc., for the next 29 pages.

- do not refer to a UC in a UC
- Why refer to a proxy? Refer to the system
- Not black box and not goal oriented
- Too long

:) “Book flight!” referencing “Find a flight-” (PEUC 4.2)

“Book flight!” Level: User Goal

1. The customer contacts the travel agency and requests a flight.
2. The customer finds a flight.
3. The system presents the travel options to the customer.
4. The customer selects a flight.
5. The system builds and reserves flight itinerary for the customer.
7. The customer provides a credit card number.
8. The system charges the flight against it and issues the ticket.

“Find a flight-”

Level: Subfunction

1. The customer requests to find a flight.
2. The travel agent captures the customer's trip origin and destination.
3. The travel agent looks up the airport codes for the origin and destination.
4. The travel agent captures the preferred departure times for the customer.
5. The travel agent captures the customer's preferred class of service.
6. The travel agent confirms that the customer's preferences are correct.
7. The system gets from the airline reservation system and presents a list of the available flights that match the customer's preferences.

Subtle MISTAKES [referencing UCs] (Book Flight)

(PEUC7.1)

Level: User Goal

1. The agent selects Set Travel Itinerary.

Call Capture Flight Information to get the travel itinerary.

2. The agent chooses Select Flight.

Call Reserve Flight Segments to reserve a flight segment.

3. The agent chooses Book Flight.

Call Obtain Payment Information to get payment choice.

4. Call Issue Ticket to print ticket.

do not call another UC: just refer to it with the convention
that underlined words are the title of another UC

:-) ‘Book Flight’ (referencing others) (PEUC 7.2’)

Level: *User Goal*

1. The agent captures the flight Information to get the travel itinerary.
2. The agent reserves flight segments.
3. The agent books the flights and obtains the payment Information.
4. The agent has the system issue the ticket.

Subtle MISTAKES [heavy extensions] (Book Flight) (PEUC 7.3)

1. The agent specifies a travel itinerary for a client.
2. The system finds and presents appropriate flights.
3. The agent selects a flight.
4. The system verifies space is available and reserves a seat.
5. The agent finalizes the booking by supplying payment information.
6. The system books the seats and issues the ticket.

- Extensions *

2a: *Client is a frequent flier*: The system retrieves the client's profile and displays the flights sorted by the client's airline preference.

4a: *Client is a frequent flier*: The system offers premium seats for the client.

4b: *Seat is not available in ticket category*: The system informs the agent and the agent specifies another price preference.

4c: *Flight is fully booked*: The system informs agent, who specifies another set of preferences.

4d: *Client is a frequent flier and seats are not available*:

The agent puts the client on a priority wait list for the seats.

5a: *Client is a frequent flier*:

1. The system verifies that client has upgrade coupons .
2. The system wait-lists the client for an upgrade.

heavy extensions indeed: is 4b an alternative to 4a or the step after...

:-) ‘Book Flight (Normal / Frequent Flier)’ separated.

“Book Flight (Normal Flier)” (as on previous, but minus all references to “frequent flier”.)

“Book Flight (Frequent Flier)”

1. The agent specifies a travel itinerary for a client.
2. The system finds and presents appropriate flights sorted by the client’s airline preference.
3. The agent selects a flight.
4. The system verifies space is available and reserves a seat .
5. The agent finalizes the booking by supplying payment information.
6. The system verifies that client has upgrade coupons, wait-lists the client for an upgrade, books the seats and issues the ticket.

Extensions

- 4a: Seat is not available in ticket category: (Do as for regular client.)
- 4c: Flight is fully booked: (Do as for regular client.)
- 4d: Seats are not available: Agent puts client on priority wait list.

STANDARD MISTAKES [business rules]

(Book Flight) (PEUC 5.3)

1. Traveler selects preferred seat.
2. The system verifies a seat has been assigned to the traveler.
3. The system verifies the seat is not in a designated exit row.
4. The system verifies the ticket is a full-price economy ticket, class Y, or is a business-class ticket, class J. If not, the system verifies the ticket is either class L, M, or N, and the traveler is a platinum member of the frequent flier program. The system verifies the seat assignment was not issued as part of a companion ticket. If not, the system verifies . . .
5. The system . . . yada yada yada . . . and the UC terminates.

do not pollute a UC with rules in the scenario: they go in the rule section

: -) “Change Seat” (with Business Rules) (PEUC 5.4)

Level: User Goal

1. Traveler asks to change seat assignment on flight number and date.
2. System verifies that the traveler is eligible for a seat change and presents the current seat map for the aircraft.
5. The traveler selects her preferred seat.
6. The system verifies that the seat is still available, assigns her chosen seat, and confirms her new seat assignment to her.

Extensions

2a: Traveler is not eligible for seat change . . .

Business Rules for Seat Reassignment:

1. Unconditional seat reassignment for full-fare economy, business class tickets, and ticket classes J and Y.
 2. Unconditional seat reassignment for discount fare codes L, M, and N if passenger is a platinum-tier frequent flier.
 3. Seat reassignment to Premium Economy seats for ticket classes L, M, N within 72 hours prior to departure time.
 4. Seat reassignment to Premium Economy seats for ticket class K 24 hours prior to departure time.
 5. No reassignment for award tickets, class C.
-

:-) “Handle Purchase ” (Larman, Applying UML and Patterns)

Level: User Goal System: Cashier's Check-out Stand

1. Customer arrives at checkout with items to purchase.
2. Cashier starts a new sale.
3. Cashier enters item identifier.
4. System records sale line item and presents item description, price (from a set of price rules), and running total.
- ...Cashier repeats steps 3-4 until indicating done.*
5. System presents total with taxes calculated.
6. Cashier tells Customer the total and asks for payment.
7. Customer pays and System handles payment.
8. System logs completed sale and sends sale/payment information to the external Accounting and Inventory systems.
9. System presents receipt.
10. Customer leaves with receipt and goods.