Software Engineering (400071) – 18 August 2004
Schema solution

The following is a schema solution: it provides only the answers that cannot be found directly in books, or that are especially difficult. Answers are just sketched out.

Q1: Life cycle models

a) see course material.
b) see course material.
c) perfective maintenance is the most expensive maintenance activity: it requires the 50% of the total maintenance effort (see page 449 text book).

Q2: Requirements engineering

a) Functional requirements and Moscow List:

Order	**Name UC**	**Motivation**
Must have	open/close flights	airline companies
Must have	open booking	
Must have	cancel/confirm booking	
Must have	issue ticket	
Should have	Search flight	
Should have	Search booking	
Should have	Authentication	
Could have	Update customer information	
Could have	Update passengers information	
Won't have	frequent traveler points gathering	

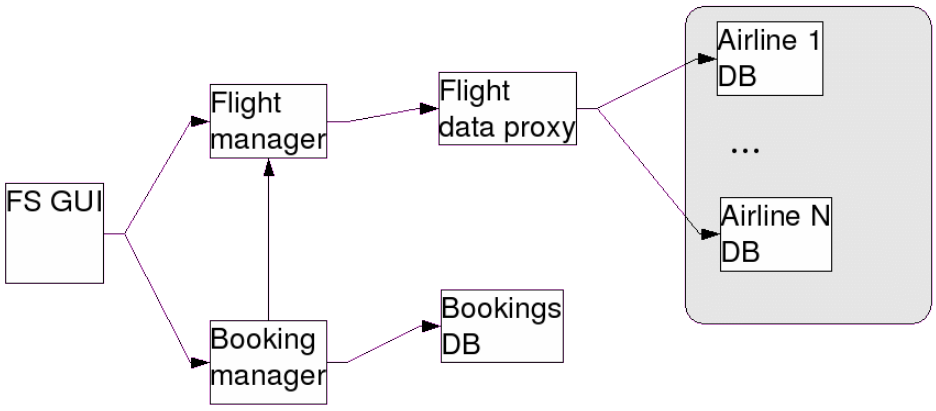
b) UCD for Must have 's: similar to previous exam, but for use cases the first four use cases above.
c) NFR: reliability (requested in the last sentence of the case study description).
d) Domain model: data classes (and their attributes) are: booking (id, passenger data, status ...); customer (name, address, data, payment method, ...); flight (id, dept date/time/airport, arrival date/time/airport, list of stopovers, status, ...). Associations are: booking 1-reservation by-* customer; booking *-reservation for-1 flight.
e) UC description: "open booking"

Basic c.o.e.	booking successful
Description	1. User inserts customer data and creates a new booking; he/she assigns a flight to the booking 2. System opens the booking with the customer and flight information 3. User enters data about passenger 4. System stores and then displays the data and asks for confirmation 5. If "yes" the booking is stored and the functionality is closed (UC stops); If "no" the booking is not stored (UC stops).
Alternative c.o.e.	No seats available for the assigned flight

	1. (the same) 2. System opens the booking with the customer information; the flight information is checked for availability but the flight is full: a message is displayed saying that there are no seats; the user is asked to either remove the booking or assign a different flight. 3. if "remove" the empty booking is deleted (UC stops); if "new flight" the UC goes back to step 1.
Exceptional c.o.e.	Flight information is incorrect
	1.-3. (the same) 4. System tries to store the data but when passenger data must be assigned to a seat, it turns out that no seat number is available – probably a concurrent booking occurred and at this moment the flight is full. The error is displayed and (again) the user is asked to either remove the booking or assign a different flight. 5. see step 3. of alternative c.o.e. "No seats available for the assigned flight".

Q3: Software design

a) Models the solution decisions, how to support requirements
b) Correct is 3b.1
c) Correct is 3c.3
d) Design view – global structure (boxes represent components; gray area represents the remote networked node of airline companies):



e) An example of design decision concerns the way flight data is handled: either by getting a notification that a flight is open/close for bookings; or by accessing the information on-line each time a booking is updated.

Q4: Testing

a) Correct is 4a.3)
b) Correct is 4b.3)