

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: XP principles

see chapter 3 of the book (revised version, online on BB).

Q2: Traditional versus evolutionary models: 2.c

Q3: Testing activities during design: 3.c, 3.d, 3.e

Q4: Moscow list

Order	Name UC	Comments
Must have	Register as member	End- users
Must have	Login/logout	End- users
Must have	Browsing	End- users
Must have	Offer for sale	End- users, seller
Should have	Start auction	Can be separated from 'offer for sale' or automated depending on the date or immediate
Must have	Bid on item	End- users, buyer
Could have	Put into contact	Administrator
Should have	Management of data	Administrator
Won't have	Payment on- line	End- users

Q5: Data model

Schema: the data model cannot contain any software component (like database, browser,

Q6: UC description

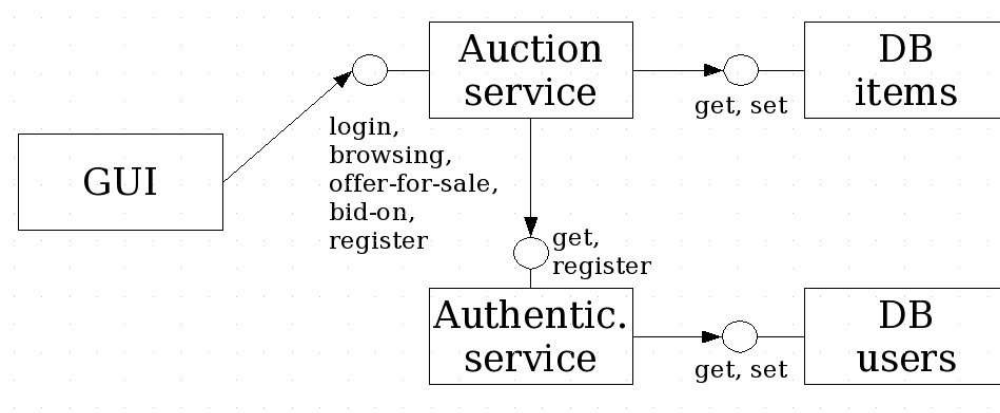
UC name “offer for sale”

Basic c.o.e.	Offer for sale successful
Description	<ol style="list-style-type: none">1. User selects option “offer for sale”.2. System returns the page where to fill in item and auction information.3. User enters data about the item to offer for sale, and selects option 'ok'4. System checks that the filled information is correct, stores this information on the database, and returns a confirmation message to the user. (the auction is started and the UC stops).
Alternative c.o.e.	Offer for sale not confirmed
	<ol style="list-style-type: none">1. to 2. (the same)3. User enters data about the item to offer for sale, and selects option 'cancel'.2. 4. System returns to the main screen (UC stops).
Exceptional c.o.e.	Offer for sale unsuccessful due to a data access failure

	1. to 3. (the same)
3.	4. System first checks that the filled information is correct. Then it tries to store the data but the database does not respond to its access requests. A failure message is displayed and the user is asked to repeat the operation later on (UC stops).

Note: it is important to remember that: firstly, the description of each c.o.e. must be written like a colloquium between user and system: each step describes only operation from either the user or the system, and they are intertwined (user- system- user- system etc.). Secondly, the exceptional c.o.e. Always concerns a failure that occurs somewhere during the normal c.o.e.; instead, the alternative c.o.e is just a different execution.

Q7: Design view



For this design view I chose the component diagram. I labelled all interfaces of the components with precisely the name of the use cases (the Must have's of the Moscow list). This allows me to check which functions must be delivered by which components. Notice that the UC "register" is repeated twice: in component Auction service, it is a function directly accessible by the user (GUI); in component Authentication service, it is an operation that is in charge of checking and storing the registration information in the DB of the users. In short, "register" is repeated twice but overloaded to execute two different methods.