Object-Oriented Analysis and Design Answers to Exam 2 on Requirements Analysis

This test has 7 questions and pages numbered 1 through 6.

Reminders

This test is open book and notes. However, it is to be done individually and you are not to exchange or share materials with other students during the test.

If you need more space, use the back of a page. Note when you do that on the front.

This test is timed. We will not grade your test if you try to take more than the time allowed. Therefore, before you begin, please take a moment to look over the entire test so that you can budget your time.

For diagrams and programs, clarity is important; if your diagrams or programs are sloppy and hard to read, you will lose points. Correct syntax also makes some difference. 1. (5 points) In Java, how are objects that implement the interface ActionListener used in constructing user interfaces?

Answer: An action listener object has a method

```
public void actionPerformed(ActionEvent e)
```

which is called when an action, such as a button push, is performed in the user interface. These objects are registered by calling the addListener method on a component, which then makes the call back to the listener when the UI event happens.

Grading: For full credit they have to say something about the actionPerformed method being called when an event happens in the user interface.

-1 point for being vague about the callbacks, but mentioning the vocabulary of the observer pattern (subscription, notification).

-1 point for saying that they have to be added to a frame for buttons, etc., but not not mentioning the callbacks or registration.

-2 points for saying that the user interface should be kept separate from the application logic layer but not being specific.

-4 points for just giving a definition of a listener.

- -5 points for giving a definition of user interface.
- 2. (5 points) Why is using time-boxed iterations important for object-oriented development?

Answer: Because it helps give quick feedback on requirements, analysis, and design. It is difficult to get these correct the first time, and in any case, the system changes the customer's perception of their requirements, which would lead to changes in the system.

Grading: For full credit, they need to say something about getting feedback during the development process, or that there is incremental progress in each iteration and the most important things get done earliest (e.g., addressing risks).

-1 point if they just say that it is hard to get the requirements etc. right the first time.

-1 point if they just say that progress is incremental, or that it helps management keep track of what is going on, without mentioning feedback or doing the most important things first

-2 points if they say it helps promote a good working environment.

-3 points if they say it helps keeps projects on schedule, or is efficient.

-3 points if they just give a definition of time boxed iterations.

3. (12 points) Consider a book circulation system for a library. An example is the circulation system at ISU's Parks library that workers at the library use to help you check out and return books. Such a system allows workers to check out books for a library patron (e.g., a student or professor), check books back in when they are returned, recall books that are checked out, in addition to keeping track of what books are in the library, and managing information about the books.

For each of the following descriptions, you are to decide whether it satisfies the elementary business process (EBP) guidelines. Write "yes" or "no" (but not both) and give a brief description of why this does or does not follow the EBP guidelines.

(a) Determining the due date for a book being checked out by a patron.

Answer: No, this is too low-level. It doesn't have measurable value by itself.

Grading: 3 points for the answer above and a reason like that.

- -1 for the answer above with what seems like the reason above, but isn't very clear.
- -2 for the opposite answer with what seems like a sensible reason.
- -3 for the opposite answer without a good reason.

A sensible/code reason for answering "yes" must address all of the criteria for an EBP, not just one of them. Take -2 points for giving only one criteria for an EBP with the wrong answer.

(b) Recalling¹ that has been checked out by a library patron.

Answer: Yes, this adds measurable value, because it is something that library patrons want to do, and it occurs at one time, and with one worker doing it, and it leaves the system in a consistent state.

Grading: same as the previous part.

(c) Deciding what books can be archived by tracking all the books that are not checked out over a period of 5 years.

Answer: No, this does not satisfy the guidelines because it does not happen at one time.

Grading: same as the previous part.

(d) Return a set of books checked out by a patron.

Answer: Yes, this adds measurable value, because it is something that the workers need to do, and it occurs at one time, and with one worker, and leaves the system in a consistent state.

Grading: same as the previous part.

For this case, take off -1 if they say it is too large (multiple EBPs).

 $^{^1\,{\}rm ``To}$ recall" a book means to ask that a book that is out of the library, but not yet do, be brought back to the library.

4. (10 points) Briefly describe one *non-functional* requirement for a library's circulation system.

Answer: It should be reliable in that it should not lose records of what books have been checked out.

Other answers are possible, such as that it should perform quickly enough so that the workers do not notice how long it is taking.

Grading: Full credit for anything that is non-functional with a sensible description.

-1 or -2 if you think the requirement is not the best, but still non-functional.

-5 if you think the requirement is completely bogus, but still non-functional.

-7 for security-related requirements

-10 for functional requirements

We can take "space for storing books" and space for storing records of the books, but take -1 for ambiguity.

5. (10 points) Consider again a library circulation system. Using the brief format, write a use case titled "Recall Book," which would involve a library patron, who interacts directly with the circulation system, recalling a book that another patron has checked out so that they can check it out themselves. (The system should notify the other patron to bring the book back as part of this use case.)

You may assume, if necessary, that the patron is already identified and authenticated by the library circulation system.

You should write your use case for a straightforward library circulation system, without lots of embellishments or fancy features. Keep it simple.

Be sure your use case is written in an essential, UI-free style. You only need to consider the "happy path."

Answer:

Recall Book: The Patron gives the System a query about a book that they wish to check out, telling the System the title, author, or other information. The System responds by telling the user information about all books that satisfy the Patron's query; this information includes the title, author, publisher, etc., and whether each book is out for circulation or not. The Patron tells the system that they wish to recall a particular books in the query result, which is checked out by another patron. The System remembers the book that is to be recalled, notifies the other patron, and confirms to the Patron using the system that the book has been recalled.

Grading: For full credit, they need to have an interaction between the Patron and the System, with confirmation, and notification of the other patron whose book is being recalled.

It's OK if they say that the use case just starts by the Patron telling the system the identity of the book they want to recall.

-1 for making the identification of the book use the ISBN number (too detailed). It's okay for the patron to tell the system the name of the book.

-1 if the notice to the other patron is only sent after confirming to the Patron that the book has been recalled. (A crash at this point would cause an inconsistency.)

-1 if they have a patron interacting with a librarian or worker who makes the recall, rather than the patron directly using the system?

-3 points for leaving out some crucial step

-4 points for only one round of communication (the user tells the system what to recall, and the system does it).

Otherwise: (as in homework)

-2 points for UI-specific vocabulary.

-2 if the notice to the other patron is given "by email" (too detailed).

-2 points for ambiguity.

-2 points for actions outside the system boundary that are not used as motivation.

6. (25 points)

Consider again a library circulation system. Using the casual format, write a use case titled "Check in Book," in which a library worker uses the circulation system to record the return of a single book by a patron, so that, in the end, the book is no longer out for circulation.

To save time, in addition to the main success scenario, just write two alternate scenarios: (a) what happens if a book being returned was never checked out, and (b) what happens if the book was overdue and returned after the library's grace period, but the patron returning the book is paying, in cash, the fine for its being late.

(To avoid complications, you can ignore the case of a book being returned that was recalled by another patron.)

Be sure your use case is written in an essential, UI-free style.

Answer:

Check in Book

Main Success Scenario: The Worker tells the System about a book that has been returned. The System confirms that the book was checked out and not overdue. The System remembers that the book has been returned, and confirms to the Worker that the book has been checked in with no problems.

Alternate Scenarios:

If the book being returned was never checked out, then the system makes no changes to its internal state. The system tells the Worker that the book had never been checked out. The use case ends.

If the book was overdue and returned after the library's grace period, then the system remembers that the book has been returned and charges the patron's account a fine for returning the book late. The System tells the Worker that the book was overdue, and also tells the Worker the amount due as a fine on the book. The Worker tells the System how much cash the patron has given to pay the fine. The System remembers that the fine has been paid, and tells the Worker the amount of change to give to the patron. The use case ends.

Grading: 10 points for the main success scenario, 5 points for the first alternative above, and 10 points for the second alternative above.

In the main success scenario, there is just one round of interaction necessary for full credit. In fact, it's bad to have multiple rounds of interaction for such a simple use case done by an expert:

-1 if multiple rounds of interaction in the main success scenario

-2 if the library patron is used, except for motivation, in the main success scenario

-3 for conditional logic.

-4 if there is no remembering or confirming done by the system

In the first alternative scenario, full credit is given for just saying that the use case ends; they don't have to write that the system state doesn't change.

It's OK in this first scenario to assume that there was some mistake previously, and to update the the system's records to show that the patron actually had the book out on loan, before and has returned it. Take -3 for having the book be recorded as currently loaned out in this scenario (the patron is trying to return it).

-2 for conditional logic in this alternative.

In the second alternative scenario, it's OK if the patron's account is charged only at the end. It's also OK if the system doesn't bother to make change.

-3 for conditional logic in this alternative.

- -3 points for leaving out some crucial step
- -4 points for only one round of communication.

Otherwise: (as in homework)

- -2 points for UI-specific vocabulary.
- -2 points for ambiguity.

-2 points for actions outside the system boundary that are not used as motivation.

7. (33 points) Consider again a library circulation system. Using the fully-dressed format, write a use case titled "Check out Books," in which the library worker records a set of one or more books that a patron is going to take out of the library.

To save time, in addition to the main success scenario, just write two alternate scenarios. These should describe (a) what happens if the book which the patron wishes to check out is non-circulating (e.g., a reference or reserve book), and (b) what happens if the patron is not allowed to check out books because they have violated some library policy (e.g., they have not paid fines for overdue recalled books). Within each of these alternative scenarios you need only consider the "happy path" (i.e., you don't have to consider alternatives to the alternatives).

Also, you can leave the "Technology and Data Variations List" section empty. You can consider failures and recovery to be an open issue.

Be sure your use case is written in an essential, UI-free style.

Answer:

Use Case: Check out Books

Primary Actor: Worker

Stakeholders and Interests:

- Worker: wants fast, and easy checkout of books.
- Patron: wants fast checkout, and does not want to be charged for books they did not check out.
- Library: wants fast checkout of books, and wants to make sure that all books that leave the library have been checked out. Wants to allocate books fairly.
- Government: wants to protect investment in books and keep costs down. Wants to promote learning and citizen happiness.

Preconditions: The Worker has been authenticated.

Success Guarantee (Postconditions): The System remembers that the Patron has checked out the books.

Main Success Scenario (or Basic Flow):

- 1. The Worker tells the System the identity of a patron who wishes to check out books.
- 2. The System confirms that the patron is allowed to check out books, and remembers the patron's identity.
- 3. The Worker tells the system the identity of a book this patron is checking out.
- 4. The System confirms that the book can circulate, calculates the due date, and records that the patron has checked out this book, which is due on the calculated due date.
- 5. The System tells the Worker the due date (which also confirms that the book has been checked out).

The Worker repeats steps 3–5 until indicates done.

Extensions (or Alternative Flows):

2a. If the patron is not allowed to check out books because they have violated some library policy,

- 1. The System tells the Worker that the patron is not allowed to check out books and the reason for this prohibition.
- 2. The use case ends.
- 4a. If the book that is being checked out is non-circulating,
 - 1. The System tells the Worker why the book is non-circulating.
 - 2. The use case continues from step 3.

Special Requirements:

- The System must respond to the Worker, at least giving some progress indication, within 3 seconds, 95% of the time.
- Workers are experts, because they use the system continuously, so the interface should have minimal interaction and should minimize the physical effort involved; for example, workers should not have to be prompted.
- Displays for the worker should be visible from one meter away.
- The system should be quiet.

Technology and Data Variations List:

- 3a. Barcode scanners are normally used to identify books.
- 3b. Books without barcodes have to be entered manually.

Frequency of Occurrence: nearly continuous.

Open Issues:

- How to deal with failures and recovery?
- How to deal with overnight checkout of reserve items.
- How to deal with library books that are unknown to the circulation system.

Grading: 2 points for the primary actor, 4 points for the stakeholders, 2 points for the preconditions, 2 points for the postcondition, 10 points for the main success scenario, 5 points for each of the alternatives, 2 points for special requirements, 1 point for frequency of occurrence. Students can omit completely the technology and a variations and open issue sections without losing points

-2 if the primary actor is the "Patron", if it's "User" try to figure out whether they meant the worker (full credit) or the patron (-2).

For the stakeholders they receive full credit if they name the workers, the library patrons, and the library. Take -1 point for leaving out the libray, -2 for the worker or the library patrons (each). Add in one point for naming the government or the University etc. Take off one point if the motivation for one of the this is not reasonable.

For the precondition, they receive full credit if they say something about authentication or the worker being "logged in". Give one point for other preconditions.

For the postcondition, they get full credit for the system remembering or knowing something sensible.

In the main success scenario, I think all the interactions are necessary for full credit.

-4 if there is no repetition to handle multiple books. But only -2 if they mentioned that there were multiple books, but didn't handle that through a repitition.

-4 points for only one round of communication.

-2 if the patron is not identified to the system; it's OK if the patron interacts directly with the system to accomplish this.

-2 if the system does not confirm that the patron has allowed to take out books.

-2 if the system does not confirm that the book is allowed to circulate in.

-2 if the library patron is used, except for motivation, in the main success scenario

-4 if there is no remembering or confirming done by the system

In the first alternative scenario, full credit is given for just saying that the use case ends; they don't have to write that the system state doesn't change.

In the second alternative scenario, full credit is only given for continuing with the next book; having the use case at that point is -3.

In the special requirements section, anything reasonable receives full credit. You can take off a point if you don't think the requirement is reasonable, -2 if this section is omitted or empty.

In the frequency of occurrence section, they give full credit for making the frequency to be fairly common such as "frequent" or "nearly continuous". Take off one point if this is omitted or empty. Give a comment but allow cannot take off any points if they say it is not frequent or "sporadic".

They don't have to have the technology and dated variations list, or open issues. They don't receive any points for these.

Otherwise: (as in homework)

-2 for conditional logic in alternatives or in the main success scenario.

-3 points for leaving out some crucial step

-2 points for UI-specific vocabulary.

-2 points for ambiguity.

-2 points for actions outside the system boundary that are not used as motivation.