Structured Requirements Analysis – Q&A

1. What is the use of developing abstract models of a system during different stages of software development process? Why is it important to develop models of both the existing system and the system to be developed?

The process of developing abstract models of a system is called system modelling. Each model presents a different view or perspective of that system. Models are used during the requirements engineering process to help derive the detailed requirements for a system, during the design process to describe the system to engineers implementing the system, and after implementation to document the system's structure and operation. Models can be developed for both the existing system and the system to be developed: Models of the existing system are used during requirements engineering. They help clarify what the existing system does, and they can be used to focus a stakeholder discussion on its strengths and weaknesses. Models of the new system are used during requirements engineering to help explain the proposed requirements to other system stakeholders. Engineers use these models to discuss design proposals and to document the system for implementation.

2. Develop a sequence diagram showing the interactions involved when a student registers for a course in a university. Courses may have limited enrolment, so the registration process must include checks that places are available. Assume that the student accesses an electronic course catalog to find out about available courses.

3. Based on your experience with a bank ATM, draw an activity diagram that models the data processing involved when a customer withdraws cash from the machine.

4. Consider an automated library circulation system. Every book has a bar code, and every borrower has a card bearing a bar code. When a borrower wishes to check out a book, the librarian scans the bar codes on the book and the borrower's card, and enters C at the computer terminal. Similarly, when a book is returned, it is again scanned and the librarian enters R . Librarians can add books (+) to the library collection or remove them (−). Borrowers can go to a terminal and determine all the books in the library by a particular author (the borrower enters A= followed by the author's name), all the books with a specifi c title (T= followed by the title), or all the books in a particular subject area (S= followed by the subject area). Finally, if a borrower wants a book currently checked out, the librarian can place a hold on the book so that, when it is returned, it will be held for the borrower who requested it (H= followed by the number of the book).

   a) Write down precise specifications for the library circulation system.

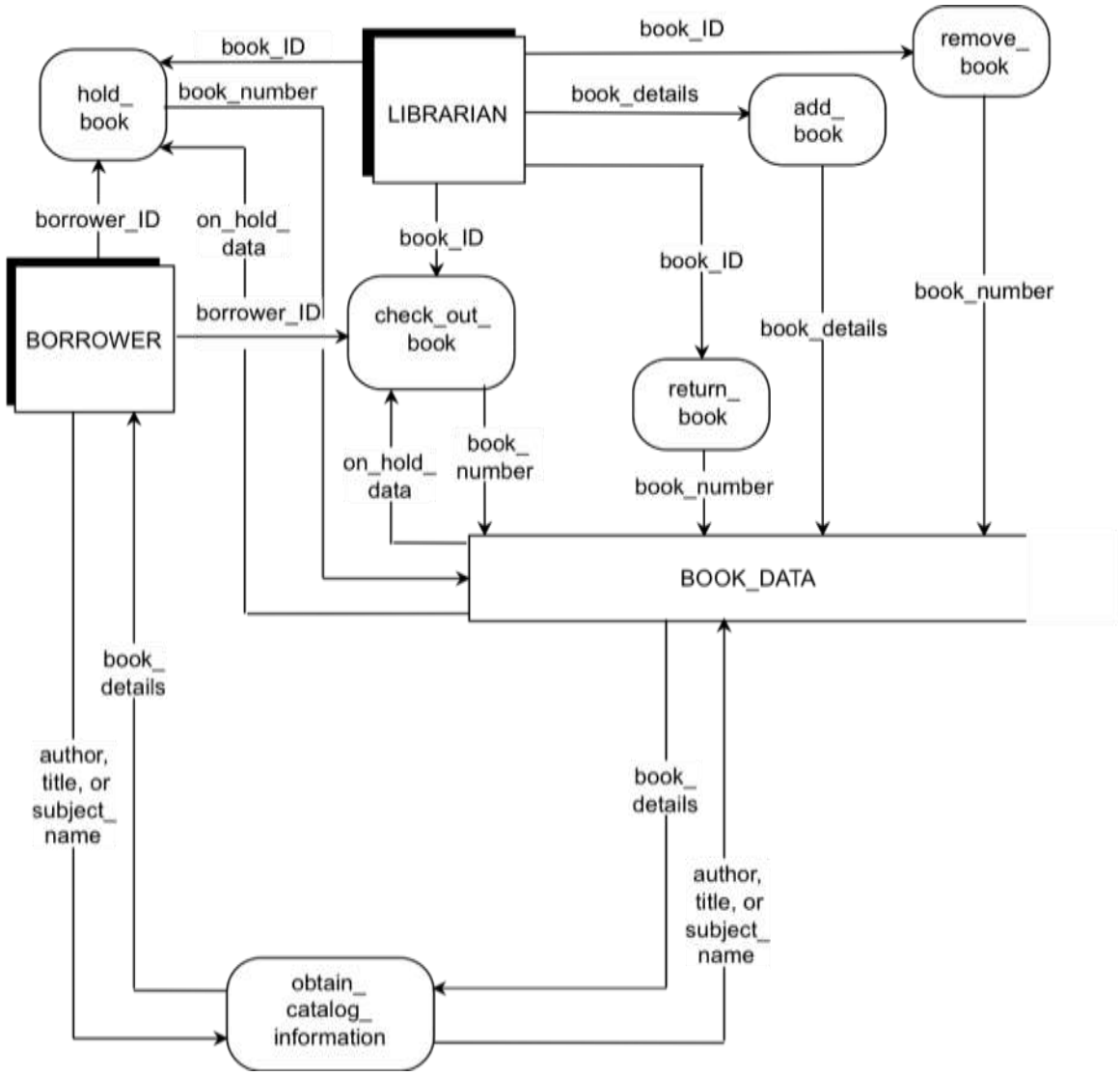
This product is concerned with books, borrowers, and librarians. The library has a collection of books that borrowers may check out and subsequently return. Each book has a unique book number. Affixed to each book is a label bearing that number in bar-code format.

Each borrower has a unique borrower number. Each borrower has a card bearing that number in bar-code format that is scanned when the borrower checks out a book. Borrowers may determine all the books by a particular author in the library collection by going to a terminal and entering A = <author name>, all the books with a specific title by entering T = <book title>, or all the books in a particular subject area by entering S = <subject area>.

Librarians check out a book by entering C at a terminal, scanning the borrower bar code, and scanning the book bar code. If the book is not being held for another reader, it is issued to the borrower. When a borrower returns a book, the librarian enters R at a terminal, and scans the book bar code. To add a new book to the collection, the librarian enters + (plus sign), the number of the new book, and other book details. To remove a book from the collection, the librarian enters - (minus sign) and enters the book number (bar-code input is inappropriate — the book may have been stolen). A librarian can put a hold on a book for a borrower by entering H = <book number> and scanning the borrower's bar code. [The need for the borrower's number was intentionally omitted from the problem statement; students should not thoughtlessly rewrite the problem statement in more formal terms, but actually consider the implications of what they are writing!] A book may not be held if it is already being held for another borrower.

b) Write down detailed specifications for the library circulation system.

Step 1. Draw DFD.



Step 2. Decide what sections to computerize, and how. Computerize complete product as shown in DFD, online. DFD does not show functions like issuing new borrower with an ID number and card, or maintaining borrower details. These functions were not included in the specifications, and it is therefore assumed that they will be performed manually.

Step 3. Details of data flows.

book_ID
    book_number

book_details
    book_number
    book_bibliographic_details
        author
        title
        subject

other_details
borrower_ID
borrower_number

on_hold_data
borrower_number if book is on hold for another borrower, else 0

Step 4. Define logic of processes.

add book
Place book_details in BOOK_DATA

check out book
Read book_ID, borrower_ID
Check that book is not being held for another borrower
Update BOOK_DATA with borrower_ID

hold book
Read book_ID, borrower_ID
Check if book is already on hold
If not, update BOOK_DATA to reflect borrower_ID

obtain catalog information
Read borrower's query
Search BOOK_DATA
Answer borrower's query

remove book
Read book_ID
Delete book_details from BOOK_DATA

return book
Read book_number
Update BOOK_DATA

Step 5. Define data stores.

BOOK_DATA (512 bytes — assumption)
book_number
author
title
subject
other details
on_hold_data

borrower_number (0 if book is not checked out)

Indexed by book_number (primary), and also by author, title, and subject (secondary)

Step 6. Define physical resources.

BOOK_DATA
Indexed sequential file
Primary index book_number
Secondary indexes author, title, and subject

Step 7. Determine input/output specifications.

book_ID, borrower_ID are input with bar code wand

book_number, book_details are input at terminal

Input screens will be designed for add book, remove book, obtain catalog information

Acknowledgment messages will be designed for add book, check out book, remove book, hold book, return book

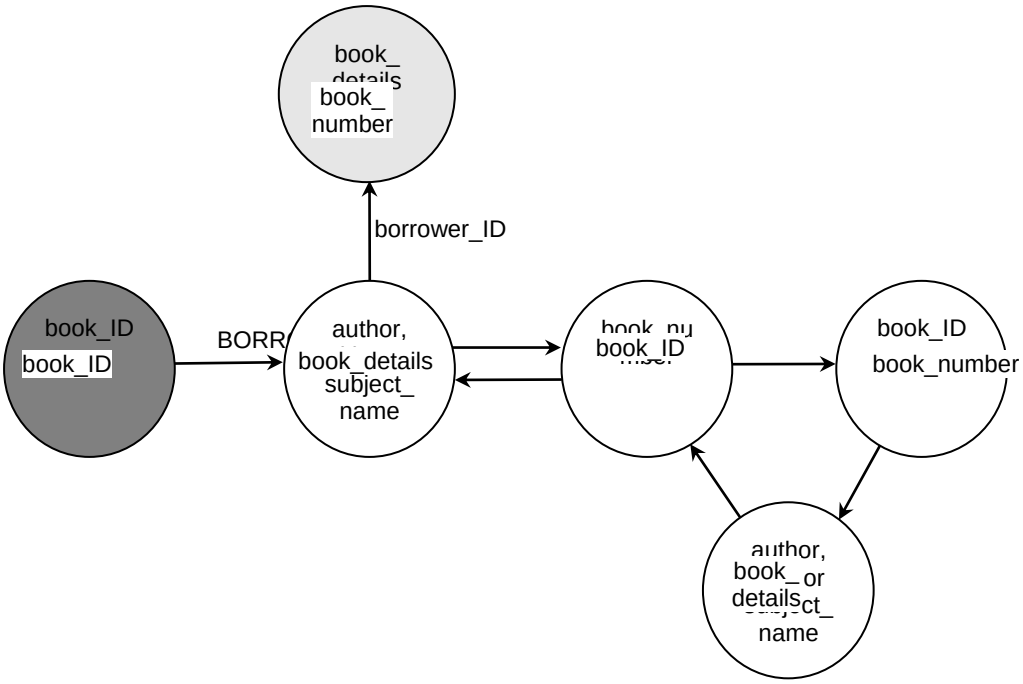
Output screens will be designed for obtain catalog information

Step 8. Perform sizing. Assuming a maximum of 500,000 books, BOOK_DATA will need to have 256 megabytes for the catalog itself, and another 25 megabytes for indexes. A further 2 megabytes will be needed for software.

Step 9. Determine hardware requirements.

CPU
5-gigabyte disk
Tape backup system
Two terminals for borrowers
Two terminals for librarians

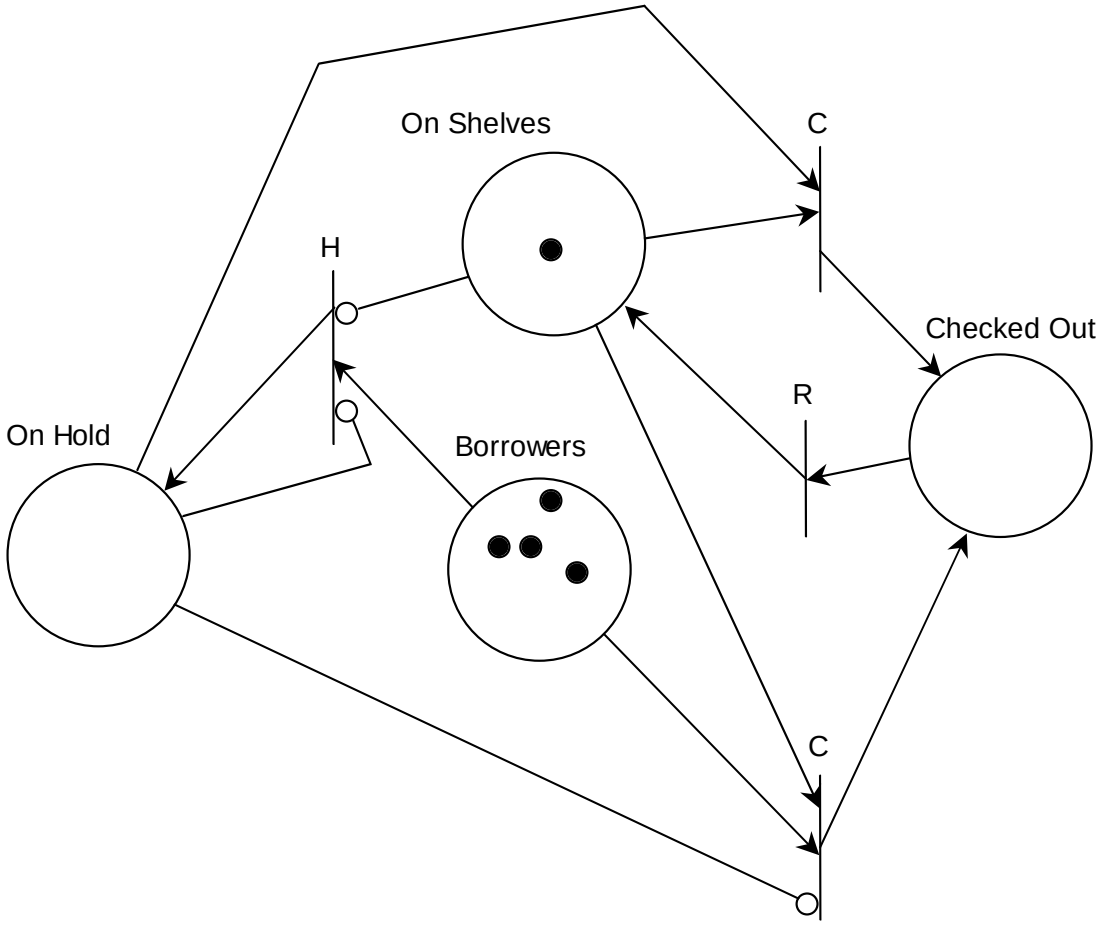
5. Use the finite state machine approach to specify the library circulation system



FSM for library circulation system.

Set of states is {New Book, Removed, On Shelves, Checked Out, On Hold, Returned And Held}
Set of inputs is {C, H, R, +, –}
Initial state is New Book
Set of final states is {Removed}

6. Use a Petri net to specify the circulation of a single book through the library Include operations H , C , and R in your specification.



Petri net specification of library circulation system.