Sample Functional Requirements

The following sample functional requirements are from BackPack, a system that will act as an online auctioning system for used textbooks and various items for the staff, faculty, and students of a university. Users of the system will be able to post and bid on books, as well as browse items currently up for auction. The system will also afford buyers and sellers the ability to post reviews.

# Functional Requirements

* 1. **Login**
     1. Authentication must be performed at the user level and via a valid NCSU student, faculty, or staff login ID.
     2. Authentication will be required for all BackPack functions.
  2. **Help**
     1. The user should be able to access context sensitive help from any place within the BackPack domain by clicking on one link.
     2. The user should have basic instructions available on the current page.
     3. Users will be able to view all their current transactions (requests for items, buying and selling) at one time.
  3. **Browsing**
     1. The user should be able to view items available for purchase.
     2. The user should be able to view items requested to be made available for purchase.
     3. The user should be able to view the review of a buyer or seller as posted by other users.
     4. The interface will include instructions and contextual prompts for performing the tasks of logging on, subscribing, selling, buying, and browsing items.
     5. The user should be able to contact persons involved with the site.
        1. The user should be able to send email to the System Administrator.
           1. There should be a contact or voice mailbox available for people to leave messages.
        2. The user should be able to send email to the System Designer (web layout administrator).
        3. The user should be able to send email to any buyer or seller from a buy, sell, or request page.
  4. **Sell**
     1. The user should be able to post an item for sale.
     2. The user should be able to set the initial bidding price.
     3. The user should be able to close the auction 24 hours after requesting that an auction with outstanding bids be closed.
     4. The user should be able to close the auction immediately if no one has a bid upon the item.
     5. The user should be able to set the default length of the auction at any point between three and 30 days.
     6. The user should be able to post a text description of the object.
     7. The user should be able to post images of the item.
     8. The user should be able to set a minimum “Go” price (the price at which the item can be sold; if the auction does not pass this point, the auction outcome is not binding), which will be kept private.
  5. **Bidding on Items**
     1. The user should be able to bid on items available for sale.
     2. All bids are binding.
        1. Non-winning bids are not binding.
        2. Winning bids may be negated by agreement of both parties.
     3. Variable bidding types will be available.
        1. The user can set a ceiling bid over which they will not bid. The system will automatically increment their bid according to their bid style, up to the ceiling bid.
  6. **Searching the Database**
     1. The user should be able to search available items according to category.
     2. The user should be able to search available items according to title.
     3. The user should be able to search available items according to seller.
     4. The user should be able to search requested items according to category.
     5. The user should be able to search requested items according to title.
     6. The user should be able to search requested items according to seller.
     7. The user should be able to search available books according to category.
     8. The user should be able to search available books according to title.
     9. The user should be able to search available books according to seller.
     10. The user should be able to search available books according to author.