Zheng (Godric) Gu

400047153

1. Architecture

Master / Slaves software architecture.

2. Proposed system

(1)Requirement:

Bookshop management system

(A)The function modules of this system is divided by the class of the users(Manager and clerk). The graphics user interface is different for manager users and clerk users. Through login interface, manager users and clerk users can enter different interfaces with different funtion modules.

(B)When stocking, select among several different suppliers and stock the same book from the supplier with the lowest price.

(C)The system has a prime storage database and a reserved one. If the prime doesn't work, the prime could work instead and any operation to the prime database should be done to the reserved database too.

(D)manager interface:

(a)system setting module: A manager can add new users (manager or clerk) to this system, exit this system and change a user and re-login.

(b)stock management module: (i) Check the stock records. (ii)Select the lowest price among different suppliers

(d)sale statistics module: (i)check all the sale records (ii)check the ranking list of the books. The ranking could be done through different methods.

(E)clerk interface:

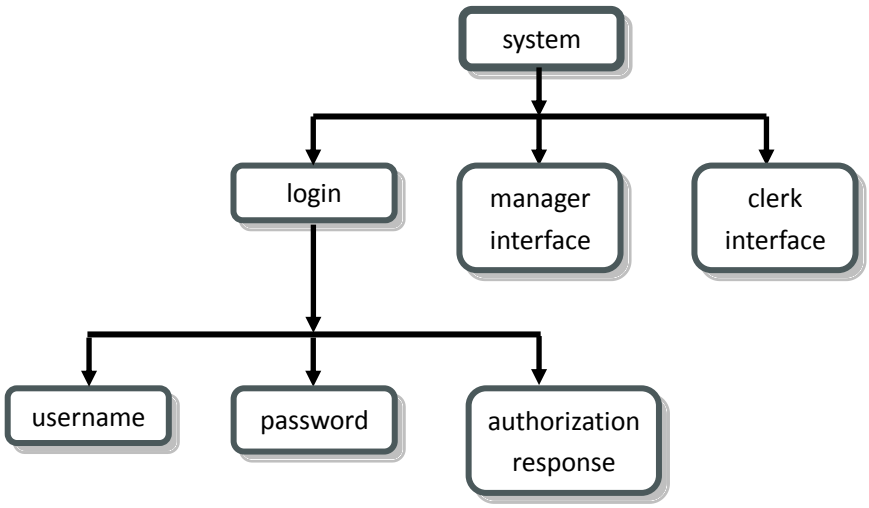
(a)system setting module: (i)Exit this system (ii)change a user and re-login.

(b)Vending function module: (i) show the storage from prime database or reserved database. (ii) book inquiry (iii) select the book and calculate the total price.

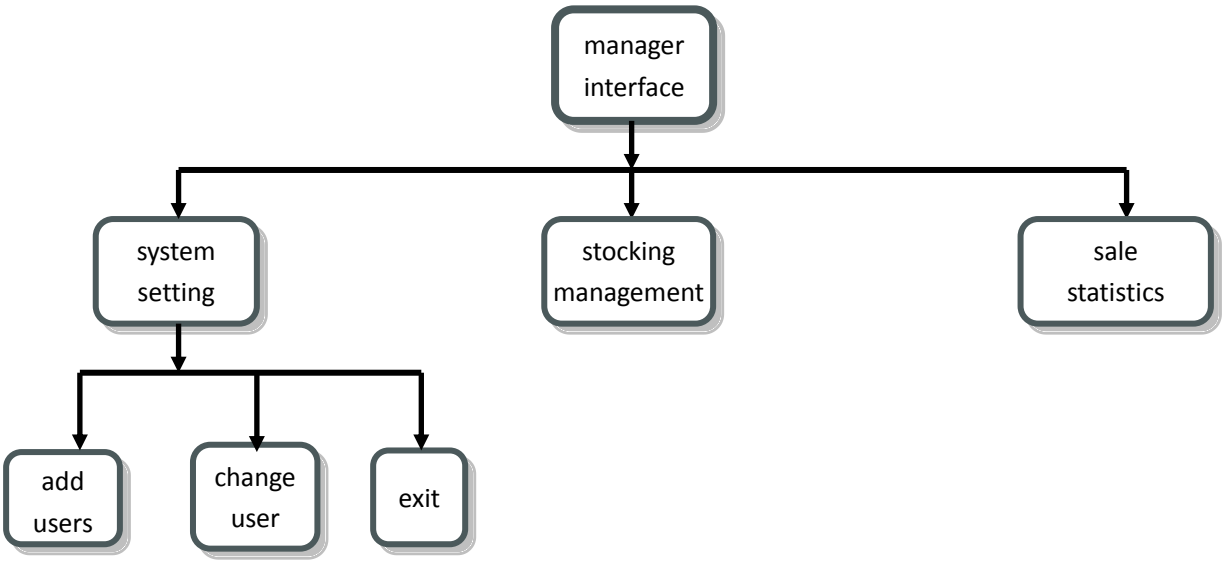
(c)Refund module: (i)check if the book sold out is from this bookshop. (ii) refund to storage.

(2)Block diagram for this system:

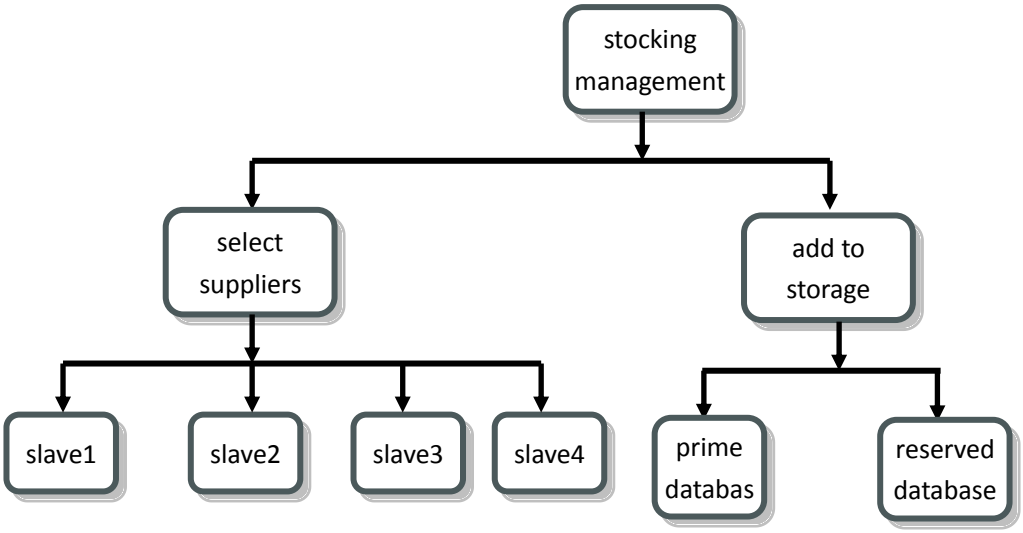
(A) Up level modules:



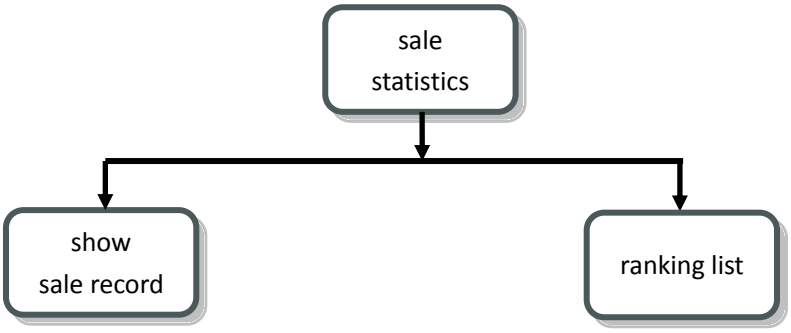
(B)manager interface



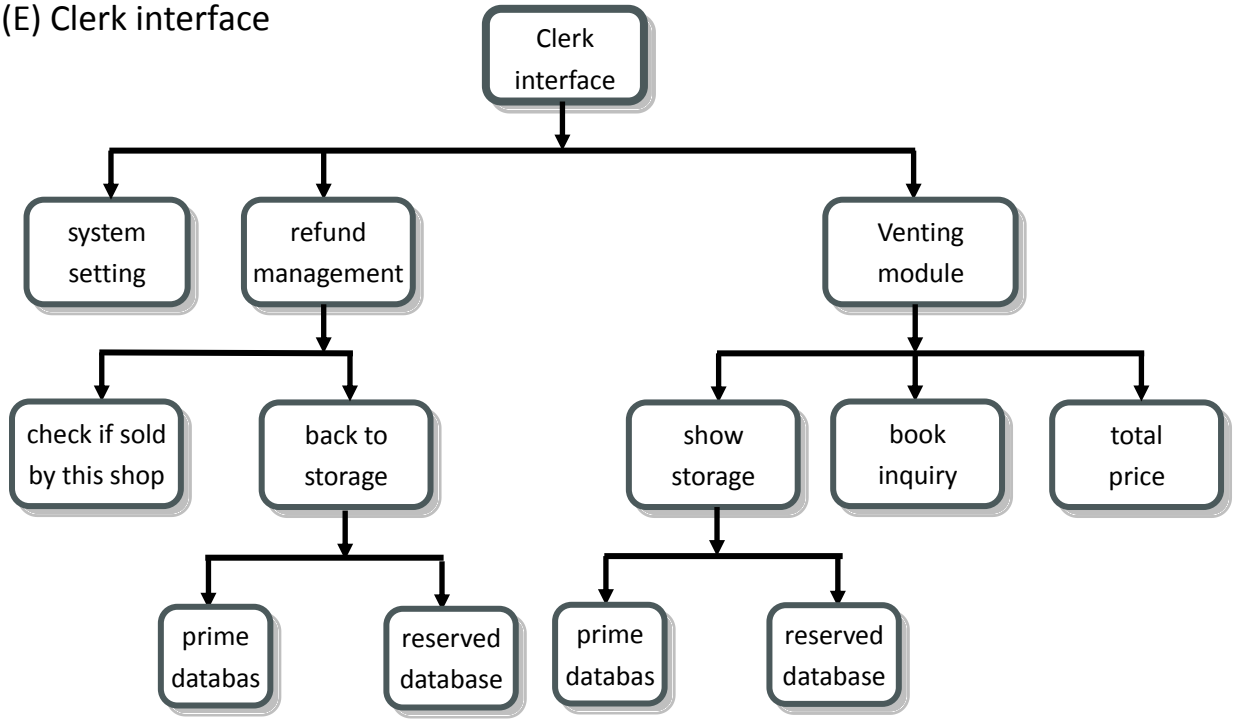
(C) stocking management module



(D) sale statistics module



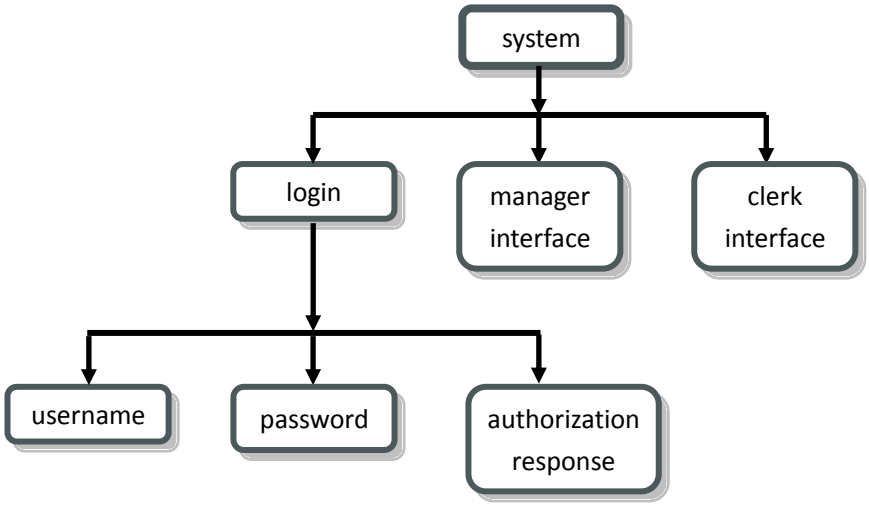
(E) Clerk interface



3. Justification

(A) This system is in hierarchical structure. The system is decomposed into different functional modules and the functional modules at different levels are connected by explicit method invocations. A lower level module provides services to its adjacent upper level modules.

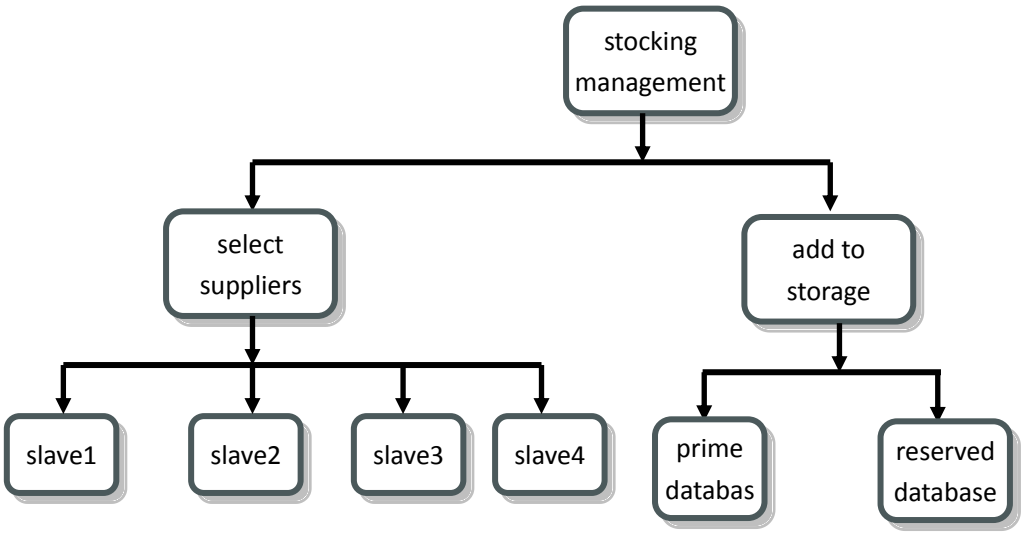
For example: in the up level,



The login module calls username function and password function to get username and password and query in the database to get the authorization response. Then the system calls the login module to get the permission to enter manager interface or clerk interface.

(B)This system uses Master/Slaves architecture to support fault tolerance and system reliability. The slaves in this system provide replicated services to the master and the master selects a particular result among all the slaves.

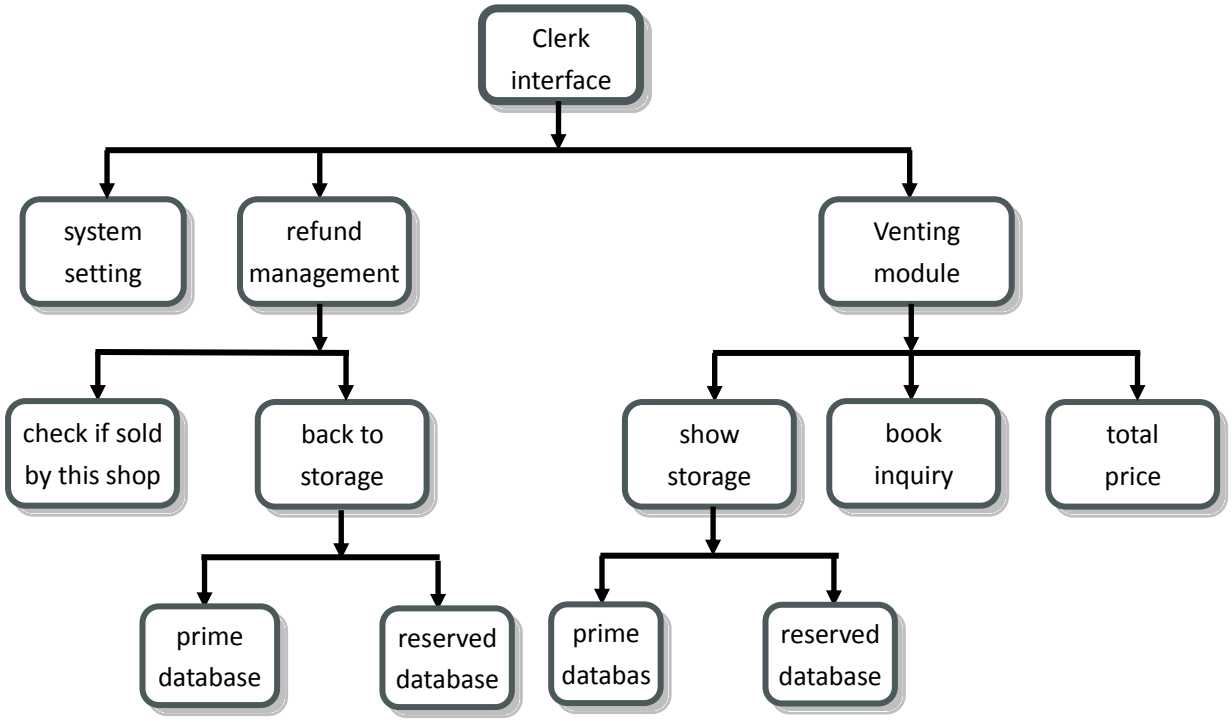
For example:



In the stocking management module, it first calls select suppliers module. All the slaves send supplier information to the upper level. In this system, slave1, slave2, slave3 send supplier information from 3 different database(maybe provided by different suppliers or different purchasing agents), and in slave4, supplier information can be input through the interface. The upper level(select suppliers module) select

the lowest price of the same books among the supplier information sent by all these slaves. It has another benefit: if the supply information of one database is wrong or missing or if one database can't be accessed, we can still get stock from other suppliers. So it will support fault tolerance and system reliability.

In the clerk interface:



The storage information of the bookshop is stored in prime database and reserved database and the information in these two database is identical. The stocking information sent by the stocking module and the refund information sent by the refund management module will be inserted both in prime database and reserved database.

In the venting module, the prime database and reserved database will all provide storage information to the upper level and if the prime database works the show storage module will always choose data from the prime database. This will provide system reliability if the prime database breaks down.