Assignment 3

Zac Crane

14151501

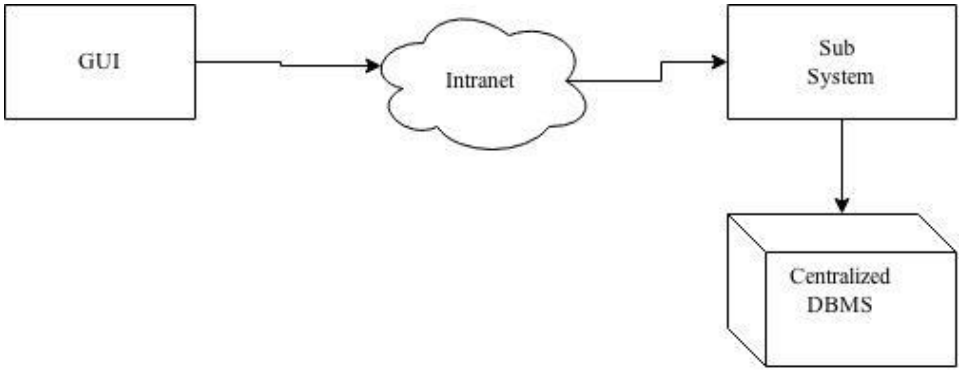
CS4320

15 February 2015

Inventory Management System

Part 1 Architecture Design for IMS:

Proposed Architecture Design for IMS



GUI: This is how the users of the system will access the program to accomplish whatever task they desire to complete.

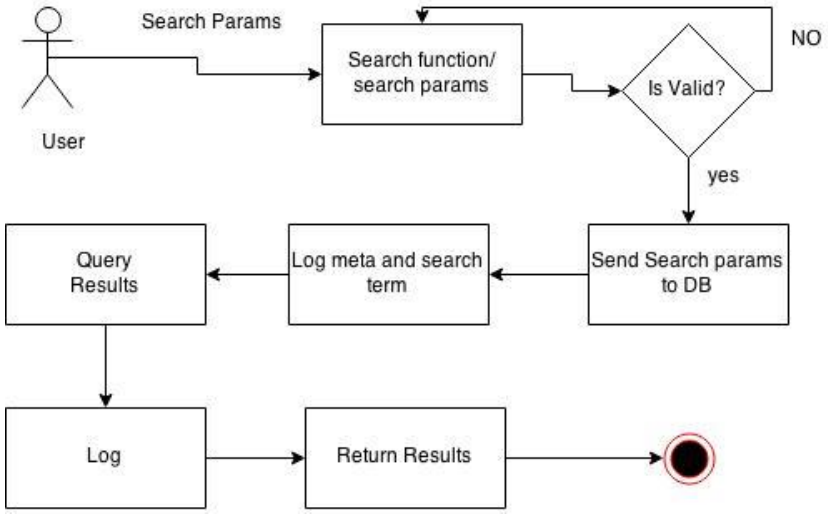
Intranet: This is how the data will be sent to the DBMS.

Sub System: I am a little fuzzy on the proper way for an Inventory management system should work, but it seems like there needs to be something separate from the DBMS to maintain ACID properties on the DB itself.

Centralized DBMS: This is where the inventory information, as well as any metadata will be stored.

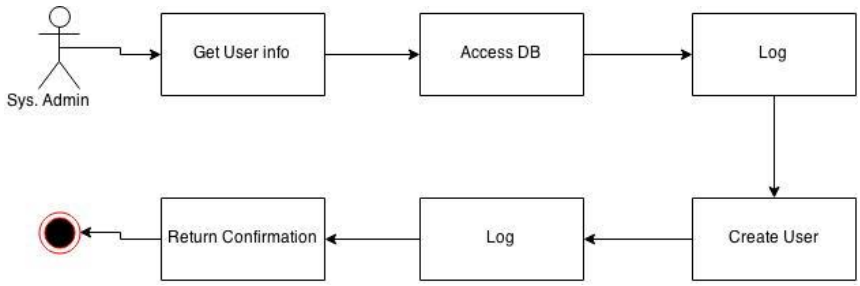
Part 2 System Design IMS:

Search System Design:



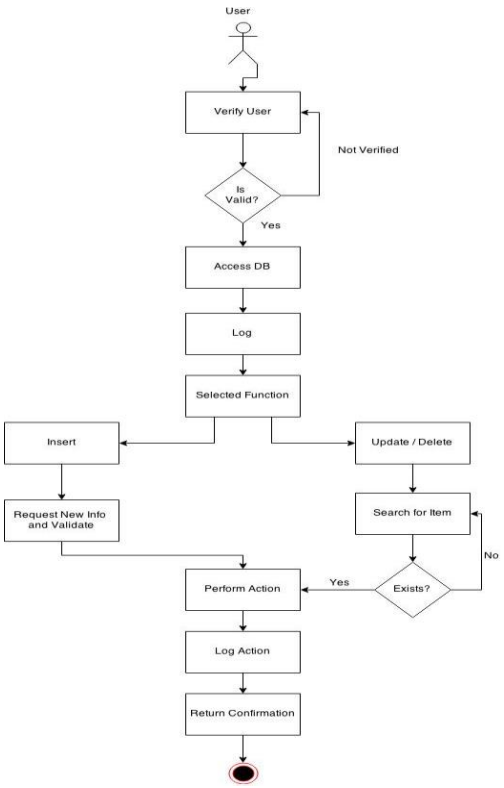
Components: The user in this situation would be access the IMS through a web UI. From the UI, the search values are submitted to the Sub system, which will validate the input before sending the parameters to the DB for further processing. When the data reaches the DB, it is logged, then the results will be generated and logged before being returned to the browers (in the long run).

System Admin S.D:



Components: The System Admin will log in via a web interface to create a new account on the server. The user's info will be gathered and verified to ensure integrity / accuracy of account info. The info will then be sent to the DB and the action will be logged. The user account will then be created on the system and another log will be created. After that, a confirmation will be returned to the user.

Department Employee:

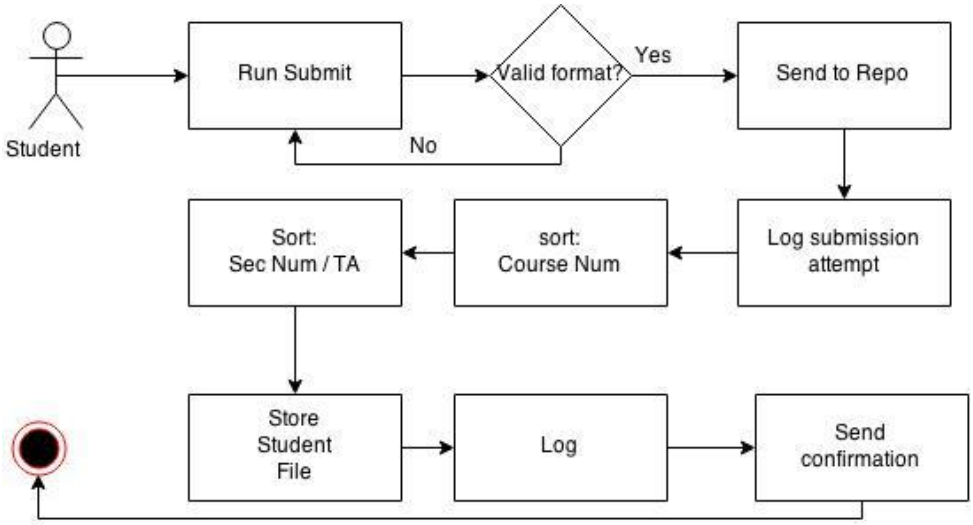


Components: in this diagram, the user will be authenticated as a department employee before proceeding any further. After validation, the user's request will be handed over to the DB and logged. Once the info is sent, the system will fork to allow for either an insert, or an update / delete. For the insert, the new info will be validated and inserted into the system. For an update/ delete, the item will be searched for to ensure that the requested item is valid. After these two steps are completed, the system merges again to allow the action to occur. This is then logged and a confirmation of action is returned to the user.

CS Submit System Design

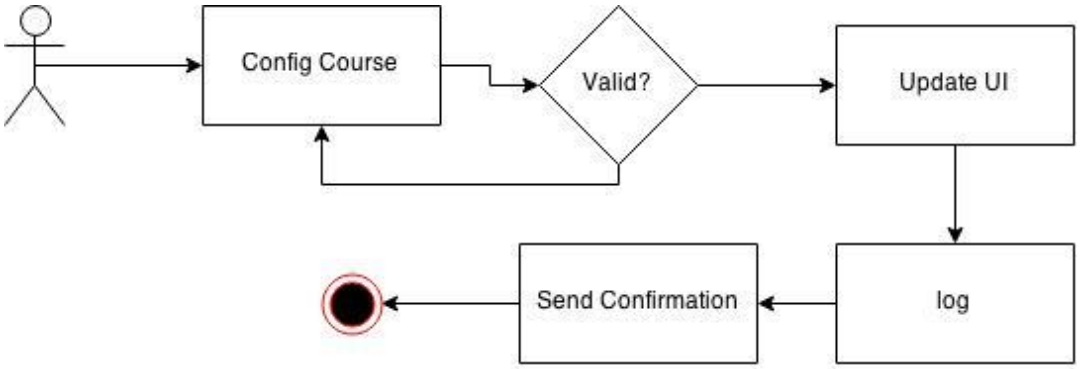
(Part 3)

Student System Diagram:



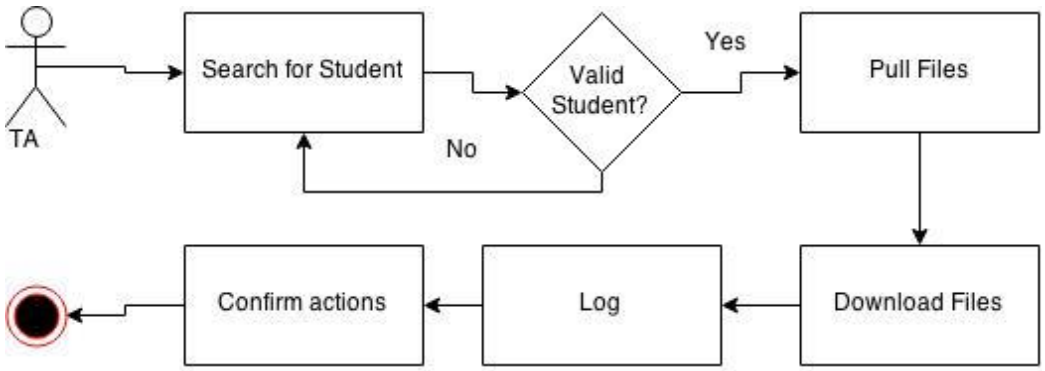
Components: The student will access the program via typing in the Submit command in a terminal. The format is checked for validity before it proceeds. After validation, the submission is sent to the repository, where it is first logged, then sorted into an appropriate location based upon the supplied info in the format. When the file is stored, it is logged again, and the confirmation is sent back to the user.

Instructor Configuration:



For the instructor to configure the course, they will provide some valid input into the program UI (however that is set up?). The program will then update the rule set / UI and log the action. After logging the change, a confirmation will be displayed to the user.

T.A / Instructor file acquisition:



Components: The T.A / Instructor will give a command to search for a student (for the case of pulling one by one, a class pull would be different slightly). Upon searching for a valid student, the user can then pull the files and download the ones they want. This will be logged and a confirmation will be sent to the user.