CptS 322 Software Engineering Principles I

Spring 2021

Homework 5

(Due April 26th, 2021 on Blackboard)

1. We have studied four architectural styles: data-centered, data-flow, call-return, and layered. Now for each style, please give an example project that would be a good fit for the style. Briefly justify your choice. [12 points]

Data-Centered:
A database architecture such as a database of all WSU students would best fit a data-centered style. A database of all WSU students would be mainly used for keeping track of the students' information, so a data-centered architecture makes the most sense to use. A data-centered architecture is based on the data that it stores and can easily change the data.

Data-Flow:
The data-flow architecture system would adapt well to a ML project. ML algorithms read in data, perform a series of mathematical calculations, and create predictions based off these calculations.

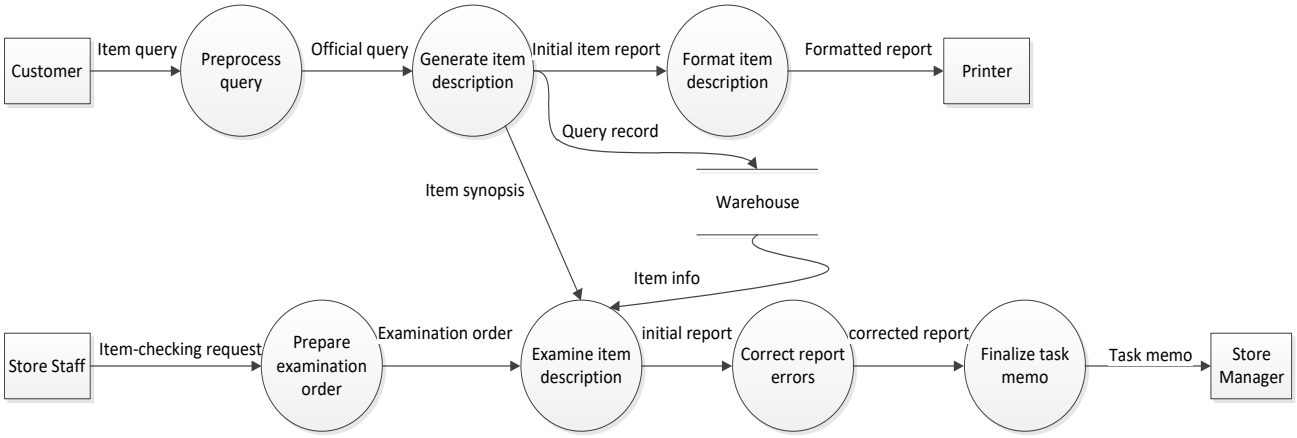
Call Return:
Call-return architecture is easy to scale and modify. A small business or organization's website would be a good fit for a call return system since as the organization grows, the website could be easily modified to better fit their needs.

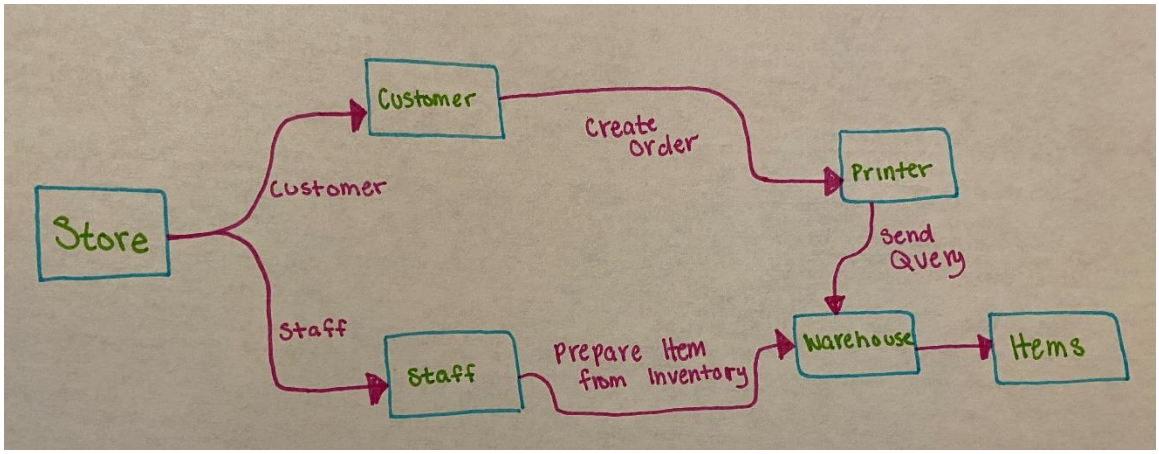
Layered:
A system such as Blackboard would be a good use of this style. A client-server application would fit a layered architecture well as the outmost layer is the UI and then inner layers will be the system interface.

2. There are two measures of functional independence in software design: cohesion and coupling. Explain in your words what each of these measures means. Is it better to have higher/lower cohesion? Why? Is it better to have higher/lower coupling? Why? [18 points]

Typically, high cohesion and low coupling is better. Cohesion is the amount that the code is focused on a task. Higher cohesion is desired as it will be more efficient at completing the task than a program with lower cohesion. Coupling is the amount that classes and functions are related to each other. Low coupling is better so that when one class or function needs to be altered, it does not affect as many other classes/functions. This makes it more efficient and simpler to change a single class/function.

3. The following data flow diagram (DFD) is part of the flow models for the requirements of the ACE store management software. It describes the process of responding to a customer request for querying about an item. Derive the architecture design from this DFD, and draw the (brief) component diagrams that represent your architectural design. [20 points]





(Sorry that my diagram is hand drawn, but my software wasn't working)