Discussion-5 high-level DFD

<u>Discussion Topic:</u>

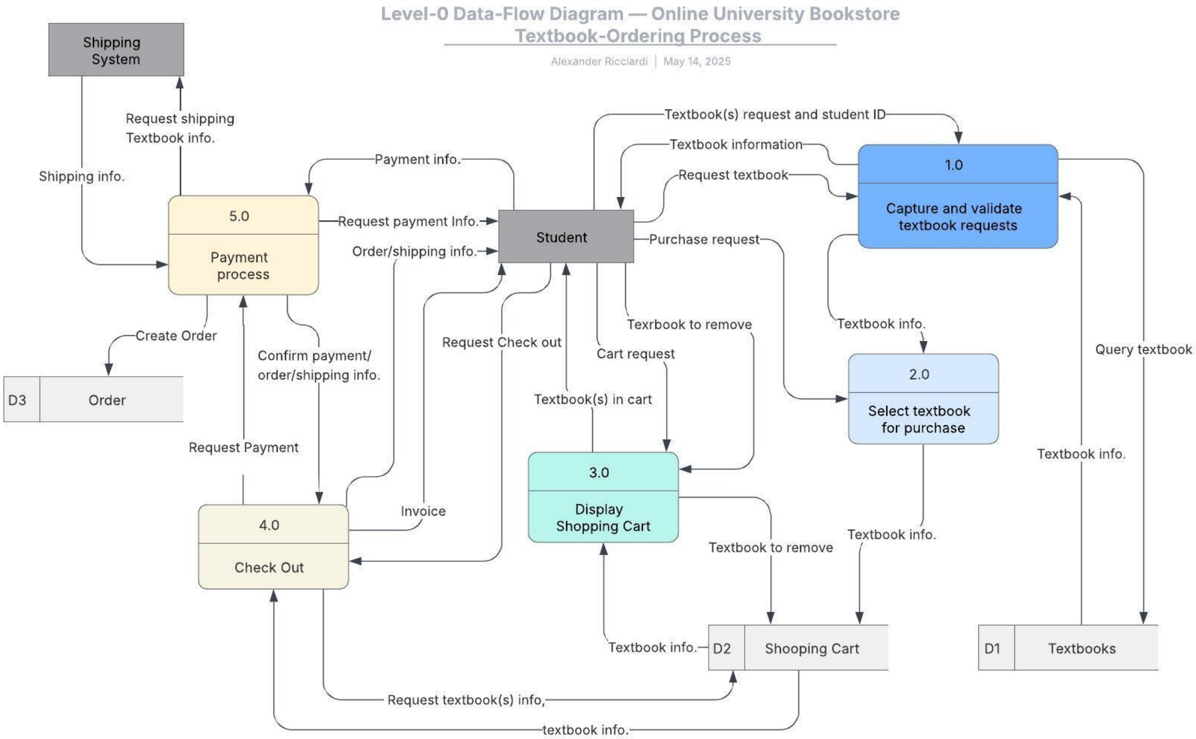
Choose a transaction that you are likely to encounter, perhaps ordering a cap and gown for graduation, and develop a high-level DFD or a context diagram. Share why you selected this transaction and share your thoughts on the data flow for the transaction with the class.

<u>My Post:</u>

Hello Class,

I chose to make a high-level Data-Flow Diagram (DFD) of ordering a textbook through the university's online bookstore.
The reason I chose this transaction is because, as a student at CSU Global, but also at LCCC, I use/used a university textbook ordering system quite often. Thus, I am familiar with most requirements of the system as a user.

Figure 1
High-Level DFD — Online University Bookstore Textbook-Ordering



A high-level DFD is an illustration of the primary individual processes in the system at the highest possible level (level-x.0). It is a decomposition of the single process illustrated by the system context diagram.

Although I am familiar with the process of ordering a textbook using an online university bookstore, making this diagram was more difficult than I thought it would be. I did not make a context diagram, I just went right into making the high-level DFD, that was a mistake, as I was not only decomposing the ordering process but also the shipping process. In other words, I was decomposing the entire online university bookstore process. As a result, my diagram was getting pretty big and crowded, making it difficult to read. So, I decided to make the context diagram for the ordering process to better define the context of the process. After creating the context diagram, I realized that the shipping system could be treated as an external entity, which allowed me to downsize my DFD.

The following table describes each process.

Table 1

Reflections on the data flow

Process	Description
1.0 Capture & validate textbook requests	Filters out invalid ISBNs or closed sections early.
2.0 Select textbook for purchase	Acts as the “add-to-cart” logic; pulls latest price & stock from **D1.**
3.0 Display shopping cart	Allow the student to review items and make changes. The loop back to **2.0** (item to remove) keeps CRUD operations balanced.
4.0 Check-out	From “shopping” to “ordering”; produces the legal invoice record that will be found in **D1 Order**.
5.0 Payment process	Combines order confirmation with payment authorization/confirmation so we never create orders we can’t bill.

Data stores

- D1 Textbooks is a real-time textbook inventory.
- D2 Shopping Cart is a transient store that tracks the student’s cart state.
- D3 Order store stores the students' order status.

Reflection

My DFD illustrates a very intertwined data flow, with the student entity at the center of it. I initially expected a more linear flow. However, the final diagram shows that there is considerable back-and-forth between processes, it is more like a web of interactions; yet, the overall data flow remains sequential.

I defined the 4.0 Check-out process and the 5.0 Payment process as two different primary processes, but I am unsure if this distinction is necessary at this level, or if it is an over-details segmentation of the flow that could be represented as a single process, e.g., just as 4.0 Check out. What do you think?

-Alex