WSUASTIS Fault Models

|  |  |  |
| --- | --- | --- |
| **Type** | **Specific**  faults related to the implementation of structures and constructs | **Non-specific**  faults that prevent conformance to the specification |
| Updating inventory | In a *sales transaction* decrement quantity in inventory database based on quantity sold  *Return transaction* – increment quantity in inventory database based on quantity returned  *Buy out case -* Don’t complete sale if quantity desired exceeds quantity available in inventory  *Return items not already in database –* not currently possible | *Manual editing inventory –* only performed by manager. Input validation is limited in this implementation. |
| Transaction ID | A *Transaction object* and corresponding database have yet been defined in this implementation but would be a necessary component in a real system. Should work on this project exceed the scope of the course work, this type of table should be included in the next implementation. | Must be infinitely unique  Specification could be interpreted to imply that a transaction object and corresponding database are created. Implementation of such data table would also solve the problem with unique Transaction IDs. |
| User permissions |  | Only managers can enter or change inventory records manually.  Only managers may change discount percentages and amounts.  Both managers and users can apply a single discount to a transaction (see Applying Transactions below.) |
| Object definitions  (hopefully limited to this implementation) | Frankly, the current implementation was not given sufficient amounts of consideration in terms of architecture and design. The objects for categories, subcategories and products may define unused attributes or be missing attributes.  A transaction object should be created as well as a corresponding database of transactions. |  |
| Modifying database contents | Database should be modified based on each transaction (sale and return) and also by manual changing of inventory by manager. It should not be modified under any other circumstances. | Manual entry of inventory records require manager permissions |
| Sales transactions | *Accurate calculations –*   * Items total * Tax total * Net total * Decrement inventory database according to how many and of which items are sold. |  |
| Return transactions | Find the product, view product details page and choose this as a return item. Choose the quantity of items being returned and increment the inventory database accordingly. |  |
| Applying discounts | . | Bulk discounts should only apply if there are a bulk number of items being purchased. Must gather from customer what qualifies a bulk discount.  Only one discount may be applied per transaction. (I.e., you cannot apply faculty transaction on top of a staff transaction).  Discounts can be applied by both basic users and managers. |
| Change discount amount | Invalid input is accepted in this implementation. 0 < valid < 1. | Only managers can change discount percentages. |
| Creating receipts | Only available via e-mail in this implementation. **Solution to implement:** A transaction record accessed from an on-going database would count as a file. Creating such a database and populating it with transaction objects as transactions occur would make handling this fault easier. | Specifications call for receipts to be outputted to a file. |
| Output | Wrong format or information  Out of date information  Unclear format  Missing information |  |
| Input | Very limited input validation is included in this implementation. |  |