RIHAM AHAMED ABDUL RAHEEM

HND COMPUTING IDM

System Documentation

Contents

[**INTRODUCTION** 1](#_Toc76885630)

[**DATA FLOW DIAGRAM** 2](#_Toc76885631)

[**FLOW CHART** 4](#_Toc76885632)

[Access level grantor 4](#_Toc76885633)

[Rate limiter 5](#_Toc76885634)

[**USE CASE DIAGRAM** 6](#_Toc76885635)

[The actor ASSISTANT\_STOCK\_CONTROLLER is responsible for the following use cases: 6](#_Toc76885636)

[The actor STOCK\_CONTROLLER is responsible for the following use cases: 7](#_Toc76885637)

[The STOCK\_CONTROLLER is one of the main actors in the stock control system, because it effectively participates in the system. 7](#_Toc76885638)

[**ENTITY RELATIONSHIP DIAGRAM (ERD)** 8](#_Toc76885639)

**List of Images**

[Figure 1: DFD 2](file:///D:\IDM\Assigement\Pending%20Assigenments\SAD\06%20System%20Documentation%20-%20Copy.docx#_Toc76885603)

[Figure 2: Access level grantor flow chart 4](file:///D:\IDM\Assigement\Pending%20Assigenments\SAD\06%20System%20Documentation%20-%20Copy.docx#_Toc76885604)

[Figure 3: Rate limiter flow chart 5](file:///D:\IDM\Assigement\Pending%20Assigenments\SAD\06%20System%20Documentation%20-%20Copy.docx#_Toc76885605)

[Figure 4: Use Case Diagram 6](file:///D:\IDM\Assigement\Pending%20Assigenments\SAD\06%20System%20Documentation%20-%20Copy.docx#_Toc76885606)

[Figure 5: ER Diagram 8](file:///D:\IDM\Assigement\Pending%20Assigenments\SAD\06%20System%20Documentation%20-%20Copy.docx#_Toc76885607)

# **INTRODUCTION**

System design is the process of designing the elements of a system, such as the architecture, modules, and components, the various interfaces of those components, and the data passing through that system.

System analysis is the process of breaking down a system into its component parts to determine how well those components interact to meet its stated requirements.

The goal of the system design process is to provide sufficiently detailed data and information about the system and its system elements to allow for consistent implementation with the architectural entities defined in models and representations of the system architecture.

IFR Belts requires a stock control system that covers the following requirements:

* Receive information of customer orders
* Print reports of customer orders
* Create sales orders and send them to suppliers in order to satisfy the customer sales orders for the coming month
* Create lists of items that are required to complete a particular customer sale
* Create a daily report of customer orders that have been completed
* Delete customer orders from the system once they have been completed
* The new workflow system should have the following levels of access:
* Report and update - for the Assistant Stock Controller
* Report, update, and delete - for the Stock Controller
* The new workflow system should be able to print information of customer orders at a rate of 15 per hour

These requirements are taken as functions of the stock control system and are designed, step by step.

# **DATA FLOW DIAGRAM**

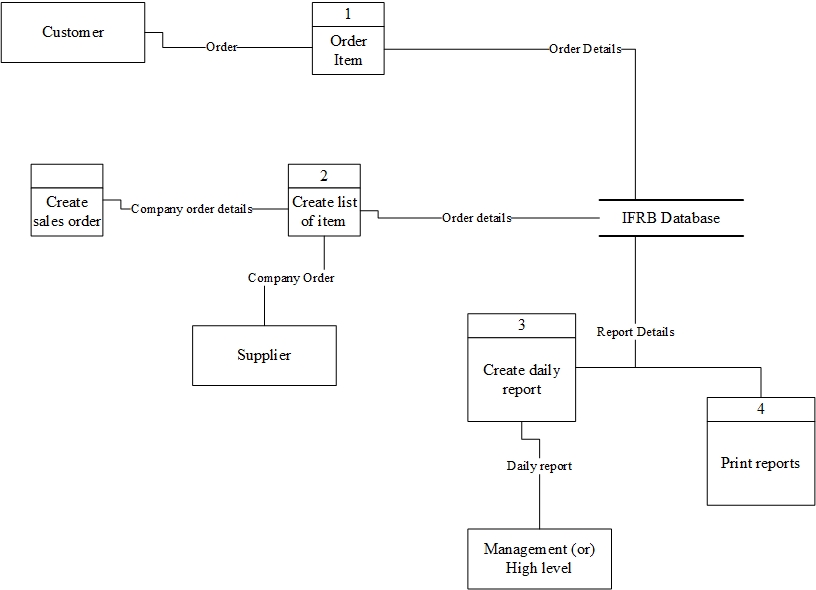
Initially, a sales order is created by the hands of a Stock Controller, this sales order is processed, validated and the data from this process flows to the next process to create a list of items to be added to the sales order. which is done by the assistant stock controller.

Figure 1: DFD

At this point, a successful sales order must be completed, after which it can be sent to suppliers as a successful sales order. This data then flows to the database and is stored there for later use.

Three of these downstream applications are report printing, reporting, and daily reporting, which are passive tasks and should be run only when initialized.

When the Stock Controller Wizard wants to print a report, the data flow from the database and processes the data for printing. Then a report is printed.

When the Assistant Stock Controller wants to supply information, the data flow from the database, and the data is processed and presented via a notification system.

When the Stock Controller Assistant wants to print a daily report, it performs a process similar to printing a report and sends the data to management if desired. This becomes a data stream that goes to an external entity, namely the administration.

The data flow is sequential, there is no data flowing backward. The flow is always constant forward; A one-way data flow is maintained.

This ensures that the data is not corrupted in any way, this flow also makes it extremely fast and predictable where the data is and makes the system less noticeable for errors. Because there is no verification during data processing and data flow, the tests that need to be performed can also be significantly reduced, leading to a further increase in performance.

This is the full idea of ​​the design of the whole implemented system, the logic of access level verification and the speed limit (15/h) will be expressed through the flow charts below.

# **FLOW CHART**

## Access level grantor

Figure 2: Access level grantor flow chart

This is the permission grantor or the access level checker.

* First the user will be tested if they have the REPORT permission, if they have it then the user is allowed to go to the next level of checking, if not the program exits with an UNAUTHORIZED message.
* Then the user will be tested if they have the UPDATE permission, if they have it then the user will be allowed to go to the next level of checking, if not the program exits with an UNAUTHORIZED message.
* Then the user will be tested if they have the DELETE permission, if they have it then they will be granted the STOCK\_CONTROLLER level of access, if they don’t have it then they will be granted the ASSISTANT\_STOCK\_CONTROLLER level of access.
* After all checks are completed, the user will have a level of access if authorized and the program exits

## Rate limiter

Figure 3: Rate limiter flow chart

This is the rate limiter design; this will decide whether or not a report can be printed.

1. First, the print request is check against the COUNTER to see if the COUNTER value is smaller than 15. If the COUNTER is lesser than 15, then the print request is sent to the next check, if not, then immediately the print request will be rejected and the program exits with a RATELIMITED message.
2. Now the print request’s time is check against the TIME stored, to see if this print request has come within an hour of the first print request in that hour, if it has then the print request is allowed to the next process if not, the print request will be rejected and the program exits with a RATELIMITED message.
3. Since all checks have been passed, the print request will be authorized and the report is printed.
4. The COUNTER value is incremented and the TIME value is appropriately changed, after which the program exits.

# **USE CASE DIAGRAM**

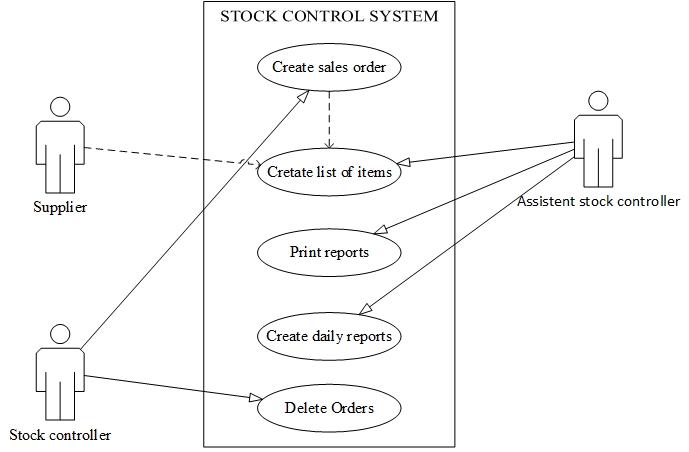
This diagram will try to explain all the interactions between the users of this system. While the provider does not participate in direct interaction with the system, it is a consumer and thus an external user, so the providers are in this use case diagram.

Figure 4: Use Case Diagram

## The actor ASSISTANT\_STOCK\_CONTROLLER is responsible for the following use cases:

1. Creating lists of items for completing custom orders.
2. Printing reports that have information about particular custom orders.
3. Creating daily reports of the custom orders that have been completed.

The ASSISTANT\_STOCK\_CONTROLLER is one of the main actors in the stock control system, because it effectively participates in the system.

## The actor STOCK\_CONTROLLER is responsible for the following use cases:

1. Creating sales orders and checking if they meet the satisfactory standards.
2. Deleting already completed sales orders.

## The STOCK\_CONTROLLER is one of the main actors in the stock control system, because it effectively participates in the system.

The actor SUPPLIER is only responsible for receiving sales orders from the STOCK\_CONTROLLER, this actor’s roles aren’t clearly scoped into the design as there is insufficient information to make assumptions whether or not they’re part of the stock control system as such it is an external entity is not a main actor of the stock control system.

# **ENTITY RELATIONSHIP DIAGRAM (ERD)**

Figure 5: ER Diagram

In the context of this business problem, the IFRB management system database must have at least 3 main entities.

The ordering entity, which tracks complete and incomplete sales orders to provide the function of creating sales orders and retrieving them for printing or analysis.

The Items entity, which contains a list of all items that the IFR Belts company supplies, calibrated by quantity so that sales orders can be created correctly with item lists.

The provider entity, which is an external entity, is therefore flagged. This entity does not need to be stored directly in the database as its attributes and the like are not specifically justified because its participation in the system is not directly justified.

The ordering entity shares a many-to-many relationship called [has] (read if Orders has items), participation is required on both sides. An order cannot exist without items and an item cannot exist without items.