RIHAM AHAMED ABDUL RAHEEM

HND COMPUTING IDM

Indiviudual report

Contents

[**THE REQUIREMENTS THAT WERE FULFILLED.** 1](#_Toc76889388)

[**ELEMENTS OF SYSTEM** 1](#_Toc76889389)

[Architecture 1](#_Toc76889390)

[Modules 1](#_Toc76889391)

[Components 1](#_Toc76889392)

[Interfaces 2](#_Toc76889393)

[Data 2](#_Toc76889394)

[**MAJOR TASKS PERFORMED DURING THE SYSTEM DESIGN PROCESS** 2](#_Toc76889395)

[Initialize design definition 2](#_Toc76889396)

[Identify design characteristics 2](#_Toc76889397)

[Assess alternatives for obtaining system elements 2](#_Toc76889398)

[Manage the design 3](#_Toc76889399)

[**FACTORS CONSIDERED DURING THE SYSTEM DESIGN PROCESS** 3](#_Toc76889400)

[Time 3](#_Toc76889401)

[Cost 3](#_Toc76889402)

[Efficiency 3](#_Toc76889403)

[**CRITICISM ON METHODOLOGIES USED** 4](#_Toc76889404)

[Why a unidirectional data flow? 4](#_Toc76889405)

[Why a component-based design? 4](#_Toc76889406)

[Why was a data flow diagram used? 4](#_Toc76889407)

[Why were flowcharts used? 5](#_Toc76889408)

[Why was use-case diagram used? 5](#_Toc76889409)

[Why was an entity relationship diagram used? 6](#_Toc76889410)

[**CONCLUSION** 7](#_Toc76889411)

# **THE REQUIREMENTS THAT WERE FULFILLED.**

The following were requirements that this system design attempted to meet, the following topics will explain how this compliance was accomplished:

* 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 work flow 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 work flow system should be able to print information of customer orders at a rate of 15 per hour

# **ELEMENTS OF SYSTEM**

## Architecture

This is the conceptual model that defines the structure, behavior, and more views of a system. I used flowcharts to represent and illustrate the architecture.

## Modules

These are components that handle a specific task on a system. A combination of modules forms the system. The use case diagram shows the modules.

## Components

This returns a particular function or a group of related functions. They are made up of modules. The ERD diagram shows the components.

## Interfaces

This is the shared boundary along which system components exchange information and interact with the system. This is shown by means of the data flow diagram.

## Data

This is the management of information and data flow. The data flow is expressed by means of the data flow diagram.

# **MAJOR TASKS PERFORMED DURING THE SYSTEM DESIGN PROCESS**

## Initialize design definition

* Plan and identify the technologies that will compose and implement the elements of the system and their physical interfaces.
* Determine which technologies and system elements are at risk of obsolescence or evolution during the system's operation phase. Plan its possible replacement.
* Document the design definition strategy, including the need and requirements for systems, products, or services that enable the design. This is done in the information section of the system documentation.

## Identify design characteristics

* Define design features related to architectural features and verify that they are feasible.
* Define interfaces that are not defined by the system architecture process or that require refinement as design details evolve.
* Define and document the design characteristics of each element of the system.

## Assess alternatives for obtaining system elements

* Evaluate design options
* Select the most suitable alternatives.
* If the decision is made to develop the resource, the rest of the design definition process and implementation process is used. If it is decided to purchase or reuse a system item, the purchasing process can be used to obtain the system item.

## Manage the design

* Lay and maintain the foundation of all choices between alternatives and decisions for design, features of the architecture.
* Evaluate and monitor the evolution of design features.

# **FACTORS CONSIDERED DURING THE SYSTEM DESIGN PROCESS**

## Time

For example, the reason why there is only a one-way data flow within the stock control system. The data only flows one way, the necessary verifications, the tests that need to be done, the implementation process, and everything that would otherwise be difficult becomes much easier because the data flow is very friendly and most importantly, predictable. . You will definitely know where the specific data is and where it came from.

## Cost

The design will evolve into a cost-effective solution and will certainly enable IFR Belts to process your projects much faster and with very little time. The design is easy to understand and easy to implement. The simplicity of implementation guarantees a great time saving because you do not have to think unnecessarily.

Rate limiter and Access Grantor prevent users from printing reports unnecessarily and thus save electricity and material resources.

## Efficiency

The combined use of one-way data flow, user-friendly interfaces, avoidance of unnecessary checks, and time-saving approaches have made this system highly efficient. It's common to sacrifice time for quality, but IFR Belts don't have to worry about quality or time, as this design can be scaled between time and quality with complete accuracy.

# **CRITICISM ON METHODOLOGIES USED**

## Why a unidirectional data flow?

This idea was inspired by the one-way data flow between the stateful parent components and their child components of ReactJS, a very popular frontend framework created by Facebook.

The unidirectional data flow allows for a couple of **very** important features in this design:

1. **Predictability –** You know where the data comes from and where it goes because data only goes one way. This makes it very easy to debug errors based on the error stack traces.
2. **Simple Interface –** Because the design is predictable, it is easy to implement. The logic involved is not difficult to implement, even for seemingly complex components and modules in the design.

## Why a component-based design?

This is to isolate different elements in the system that have similar logic or purposes. This way, one element is less likely to affect another element. Most importantly, though, the reason there is a component-based architecture is so that the components can be reused in the future in this way.

For example, the licenser and the speed limiter components in the architecture can be reused in any other design architecture and then implemented very easily.

## Why was a data flow diagram used?

Dataflow diagrams are most effective for representing one-way dataflow in my design, the following are additional reasons I've used them:

* It helps to describe the boundaries of the system, leading to immediate implementation and understanding.
* It is useful to transfer knowledge of the existing system to users, so that users of the inventory management system at IFR Belts can quickly pick it up and put it into use.
* A simple graphical technique that is easy to understand, so IFR Belts developers don't have to struggle with understanding.
* DFDs can provide a detailed view of system processes.
* DFDs are easier to understand by technical and non-technical audiences.
* Most importantly, it supports the logic behind the data flow within the system.

## Why were flowcharts used?

The flowcharts have been used to give an idea of ​​the algorithms that need to be implemented to meet the requirements of the licensing and speed limiting system of the inventory management system, the reasons why I have instead used a flowchart to directly run the algorithm. declare. in the natural language it is:

* Flowcharts are a better way to communicate the logic of a system to non-technical personnel at IFR Belts.
* Flowcharts allow problems to be designed in a more effective way, reducing costs and wasted time, which is a major benefit as IFR Belts struggle with time.
* Flowcharts serve as design documentation, which is much needed in this process, making things more efficient.
* The flowcharts act as a guide or blueprint during the system analysis and development phase of the program and ensure that the algorithms are produced on time and delivered by IFR Belts on time.
* The flowchart will certainly help in debugging the algorithm, reducing the time required, which is an advantage for IFR Belts.
* Design maintenance becomes easy with the help of a flowchart. It helps the programmer to work more efficiently with those algorithms.

## Why was use-case diagram used?

The use case diagram directly shows how system users interact with the inventory management system and how the system responds to each interaction. The users of the inventory management system are identified as actors and each requirement, both of the user and of the system, is modeled as system roles. The other reasons it was used are:

* Helps to capture the system requirements of the inventory management system.
* Use cases can serve as the basis for estimation, programming and validation efforts, reducing the time required, which is an advantage for IFR Belts.
* The use case can evolve in each iteration from a requirement capture method, to develop guidelines for developers, to a test case, and finally to user documentation after implementation is complete, reducing downtime and maintaining quality.
* Use cases include alternative pathways that capture additional behaviors that can improve system responsiveness and sensitivity change.
* The use cases have proven to be easy to understand for non-technical users, making it an excellent bridge between developers and system users to show what roles a particular user has in the system and how the developer has implemented them. this paper.

## Why was an entity relationship diagram used?

The main reason is that this system requires a database, and databases are best designed with entity-relationship diagrams. Although the existence of the database is not clearly mentioned in the user or system requirements, it is an indirect inclination that a database should exist within the stock control system for robust data storage and persistence in the inventory. the grand scheme of things.

# **CONCLUSION**

I believe that the user and system requirements are perfectly met, not only in the standard but also beyond. This particular robust design with multi-component-based design and architecture considerations will certainly reduce implementation downtime and enable IFR Belts to provide an inventory management system very quickly.

When this design was developed, not only time-saving approaches were considered, but several quality assurance approaches were also considered. What you often see is that when time-saving approaches are implemented, the designs seem too simple and result in a lack of quality.

This design certainly reduces some quality, but it's not significant enough to be noticed. For example, for a one-way data flow, you could have used a multi-way data flow, which would have made it more complex, but added some robustness between processes. But I think the robustness between processes is not necessary, the overall robustness of the system is much more important. Thus, the one-way data flow was born.

It is completely within the scope of the user and system requirements and does not reduce any features. What is not needed just gets in the way of the system and requires additional time, resources, testing, and logic to implement it perfectly. Time is something IFR Belts struggles with, so these kinds of small quality sacrifices have to be made to fit the model of user and system expectations