Assessment Report

Will describe the effectiveness the design has and how it meets system and user requirements

K.P.I. Shenesh Perera

IDM-Negombo  System Analysis & Design

Contents

[The requirements that were fulfilled 2](#_Toc11783932)

[Elements of a System 2](#_Toc11783933)

[Major Tasks That Were Performed During the System Design Process 2](#_Toc11783934)

[Initialize design definition 2](#_Toc11783935)

[Establish design characteristics 3](#_Toc11783936)

[Assess alternatives for obtaining system elements 3](#_Toc11783937)

[Manage the design 3](#_Toc11783938)

[Factors that were considered during the system design process 3](#_Toc11783939)

[Time 3](#_Toc11783940)

[Cost 3](#_Toc11783941)

[Efficiency 4](#_Toc11783942)

[Criticism on methodologies used 4](#_Toc11783943)

[Why a unidirectional data flow? 4](#_Toc11783944)

[Why a component based design? 4](#_Toc11783945)

[Why was a data flow diagram used? 5](#_Toc11783946)

[Why were flowcharts used? 5](#_Toc11783947)

[Why was a use-case diagram used? 5](#_Toc11783948)

[Why was an entity relationship diagram used? 6](#_Toc11783949)

[Conclusion 6](#_Toc11783950)

# The requirements that were fulfilled

The following requirements were what this system design attempted to fulfill, the topics that follow will explain how this fulfillment was done.

* 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 a 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** - This are components that handle one specific tasks in a system. A combination of the modules makes up the system. The use case diagram shows the modules.

**Components** - This provides a particular function or group of related functions. They are made up of modules. The ERD diagram demonstrates the components.

**Interfaces** - This is the shared boundary across which the components of the system exchange information and relate with the system. This is shown through the Data flow diagram.

**Data** - This the management of the information and data flow. Data flow is expressed through the data flow diagram.

# Major Tasks That Were Performed During the System Design Process

## Initialize design definition

* Plan for and Identify the technologies that will compose and implement the systems elements and their physical interfaces.
* Determine which technologies and system elements have a risk to become obsolete, or evolve during the operation stage of the system. Plan for their potential replacement.
* Document the design definition strategy, including the need for and requirements of any enabling systems, products, or services to perform the design. This is done in the briefing section of the system documentation.

## Establish design characteristics

* Define the design characteristics relating to the architectural characteristics and check that they are implementable.
* Define the interfaces that were not defined by the System Architecture process or that need to be refined as the design details evolve.
* Define and document the design characteristics of each system element.

## Assess alternatives for obtaining system elements

* Assess the design options
* Select the most appropriate alternatives.
* If the decision is made to develop the system element, rest of the design definition process and the implementation process are used. If the decision is to buy or reuse a system element, the acquisition process may be used to obtain the system element.

## Manage the design

* Capture and maintain the rationale for all selections among alternatives and decisions for the design, architecture characteristics.
* Assess and control the evolution of the design characteristics.

# Factors that were considered during the system design process

## Time

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

## Cost

The design will evolve into a cost effective solution and will definitely allow IFR Belts to process their projects much faster with extremely less time. The design is simple to understand and easy to execute. Execution simpleness will grant a large saving of time, because you don’t have to think unnecessarily.

The Ratelimiter and the Access Grantor will prevent users from printing reports unnecessarily and thus save both electricity and tangible resources.

## Efficiency

The combined use of unidirectional data flow, simple to use interfaces, avoiding unnecessary checks and time-cutting approaches have made this system extremely efficient. It is common to sacrifice time for quality, but IFR belts doesn’t have to worry about quality or time because this design will completely scale accurately between time and quality.

# Criticism on methodologies used

## Why a unidirectional data flow?

This idea was inspired by the unidirectional data flow between stateful parent components and their children components of ReactJS, which is a very popular frontend framework made 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 to, because data only goes one way. This makes it very easy to debug errors, based on error stack traces.
2. Simple Interface – Because the design is predictable, it is simple to implement. The logic involved isn’t difficult to implement even for very seemly 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 the likelihood of one element affecting the other element is less. Most importantly however, the reason why a component based architecture exists is because this way, components will be reusable in the future.

The permission grantor and the rate-limiter components in the architecture for example, can be reused in any other design architecture and then implemented very easily.

## Why was a data flow diagram used?

Data flow diagrams are the most effective to show the unidirectional data flow in my design, following are additional reasons why I used it:

* It aids in describing the boundaries of the system, which leads to direct implementation and understanding.
* It is beneficial for communicating existing system knowledge to the users, so users of the stock control system at IFR Belts can quickly pick it up and start using it.
* A straightforward graphical technique which is eay to understand, so developers at IFR belts don’t have to struggle with understanding.
* DFDs can provide a detailed representation 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.

**Data flow diagrams alongside the use case diagrams help support the user requirements of creating sales orders, reports and printing reports. The data flow diagrams directly reflect how the data must flow in order for the user requirements to be met successfully.**

## Why were flowcharts used?

The flowcharts were used to give insight on the algorithms that will have to be implemented in order to meet the ratelimiting and permission granting system requirements of the stock control system, the reasons as to why I used a flowchart instead of directly declaring the algorithm in a natural language is:

* Flowcharts are better way of communicating the logic of a system to non-technical staff in IFR Belts
* With the help of flowcharts, problems can be designed in more effective way therefore reducing cost and wastage of time, which is a great plus since IFR Belts struggle with time.
* Flowcharts serve as a design documentation, which is much needed in this process, making things more efficient.
* The flowcharts act as a guide or blueprint during the systems analysis and program development phase, which makes sure that the algorithms will be produced in a timely manner while ensuring that IFR Belts provides it on time.
* The flowchart will definitely help in the debugging process of the algorithm, which reduces the time required which is an advantage for IFRB.
* The maintenance of design becomes easy with the help of flowchart. It helps the programmer to put effort more efficiently on those algorithms

## Why was a use-case diagram used?

The use case diagram directly reflects how the users of the system interact with the stock control system and how the system reacts to each interaction. The users of the stock control system were declared as actors and each requirement both user and system, are modeled as roles of the system. The other reasons why it was used are:

* Helps to capture the system requirements of the stock control system.
* Use cases can serve as the basis for the estimating, scheduling, and validating effort, which cuts down the time required which is an advantage for IFR Belts.
* Use case can evolve at each iteration from a method of capturing requirements, to development guidelines to programmers, to a test case and finally into user a documentation after implementation is complete, cutting down time while maintaining quality.
* Use cases contain alternative paths capture additional behavior that can improve system reactivity and sensitivity change.
* Use cases have proven to be easily understandable by users who have no technical knowledge, and so it is an excellent bridge between the developers and the users of the system to show what roles a particular user has in the system and how the developer has implemented this role.

## Why was an entity relationship diagram used?

The main reason is because this system requires a database, and databases are designed best 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 must exist within the stock control system for robust data storage and persistence in the grand scheme of things.

# Conclusion

I believe the user and system requirements have been perfectly met, not only to standard but beyond. This particular robust design with multiple architecture considerations and component based design, is sure to cut down time of implementation and allow IFR Belts to provide a stock control system very quickly.

When this design was put together, not only was time cutting approaches considered but also a several quality assurance approaches were also considered. What you see usually is that when time cutting approaches are implemented, the designs look too simple and result in lack of quality.

This design admittedly does cut some quality, but nowhere near significant enough to be noticed. For example, the unidirectional data flow, I could have used a multidirectional data flow which would’ve made it more complex, but added some robustness between processes. But I believe that robustness between processes are unnecessary, the total robustness of the system is much more important. Therefore, the unidirectional data flow was born.

It is entirely within the scope of the user and system requirements and does not cut down any feature. What’s unnecessary will just get in the way of the system, and will require extra time, resources, tests and logic to perfectly implement. Time is something IFR Belts struggle with so this kind of minor quality sacrifices must be made in order to fit the expectation model of the users and the system.

|  |  |
| --- | --- |
| **Bibliography** | |
| Reddit: Design | <https://www.reddit.com/r/Design/> |
| ReactJS docs | <https://reactjs.org/docs/thinking-in-react.html> |
| LucidChart | [https://www.lucidchart.com](https://www.lucidchart.com/) |