

Development of an open source ERP (Enterprise resource planning) system for a sales and distribution industries

Mr.parthiban M¹, Mr. Tamil vanan N²

¹Assistant Professor, Department of Mechanical Engineering, PSG College of Technology, Coimbatore, Tamilnadu, India.

²M.E (Lean manufacturing), second year, PSG College of Technology, Coimbatore, Tamilnadu, India.

Abstract- Lean is a process management philosophy. It is a business model that delivers a better performance towards customers, employees and stakeholders. By focusing on the customer, defining value and eliminating waste is a process has consistently delivered more from existing resources with little additional cost. Lean is a mindset that demands continuously improvement and care for people that results in elimination of waste from processes. Reducing the lean wastages and Improvement in sales and distribution is the main aim of project. It can be achieved with the help of implementing an appropriate ERP (Enterprise Resource Planning) package into the organization.

Key Words: Entity relationship diagram, web application framework, C#(c sharp coding), reusability, modularity, loose- coupling.

1.INTRODUCTION

The major problems faced by the sales and distribution industry are there is no proper transportation in distribution of materials to the customers, unwanted transportation occurs due to improper communication with the customer orders and there is no proper planning of distribution (i.e. transportation map routes). The one main problem in this distribution industry is there is no proper warehouse management as only traditional method of warehouse management were followed.

1.1 C# (c sharp)

C# (c has been chosen to write coding (or) developing sharp) is a high-level programming language which a database for this ERP package sharp is chosen as a programming language due to its unique characteristics. Some of the unique characteristics of this c# (c sharp) language are,

1. Reusability
2. Modularity
3. Loose-coupling.

1.2 Reusability

When we write code for the software, we should write in such a way that the code can be used multiple items in our package. To achieve this, put the coding in assemblies. Visual studio provides an easy way to write code in class libraries which exists as assemblies. This method of programming is

also called as API (Application Programming Interface). By doing this coding is become reusable. Suppose, in this package I have chosen to do a sales management system. Data access code and business access code has to be written. If you place this code in separate assemblies, then you will be able to re-use the same code in other projects. However, the ability to write code which is re-usable requires some level of expertise.

1.3 Modularity

Modularity in software application development means dividing the application into tiny parts with each part having a purpose (or) a task. The tiniest part you can make in visualstudio.Net is a class. Remember, each class should have single specific clearly defined task. When we divide the application into classes for modularity, we should also ensure that various classes interact with each other in a well-coordinated way to achieve the objective of the application.VS (visual studio) provides means to manage these classes by specifying which classes are interdependent and in what order they should be interfaced with others..In large applications Microsoft introduced what is called as a design patterns to make it easier to implement modularity in an application. A design pattern is a sort of a template (or) a guide, developed and tested and ensures high degrees of reliability. It has to be used as a guide at the implementation level for a specific purpose.

1.4 Loose - coupling

The rapid increase in usage of web applications and Smartphone applications resulted in a large demand for creation of loosely coupled applications. As explained above, some degrees of modularity is achieved in application by creating classes. However, for today's enterprise and smart phones applications a higher degree of modularity and hence reusability is required. When creating classes, we will observe that all dependent classes have to be instantiated in the constructor. Task such as these will be handled by IOC containers which are the latest solution offered by Microsoft for achieving high degrees of loose coupling. As we have seen in the earlier that each class should be designed to perform a certain given task. Visual studio provides a feature called interfaces using which we can define business rules. The classes we create will implement these interfaces. If any business rule is added (or) modified. When we change the interface, classes will adopt the changes immediately and

automatically. As enterprise applications contain hundreds of classes and interfaces, managing them are difficult. IOC containers ensure that these classes interact with each other and smoothly. IOC containers are the latest type of solution offered by the Microsoft to achieving the higher degree of loose coupling. Which is necessary for today's enterprise and latest Smartphone application.

2. ER Diagram (Entity relationship diagram)

The ER diagram represents the entities and relationship between them. To give an idea about ER diagram I have considered the mini version of sales order management system. Each sales order item lists a product which belongs to a particular category. Listed below are the entities and their purpose of a sales order management system.

- **Customer:** A person who places orders.
- **Sales order:** An order placed by a customer. A single order may have multiple products.
- **Product:** A product is a thing or goods purchased by a customer.
- **Category:** A category is a description to which a product or products belong.
- **Salesorderitem:** The product and all the details of the product form the Salesorderitem. An order can have multiple Salesorderitem.

2.1 Application entity relationship diagram

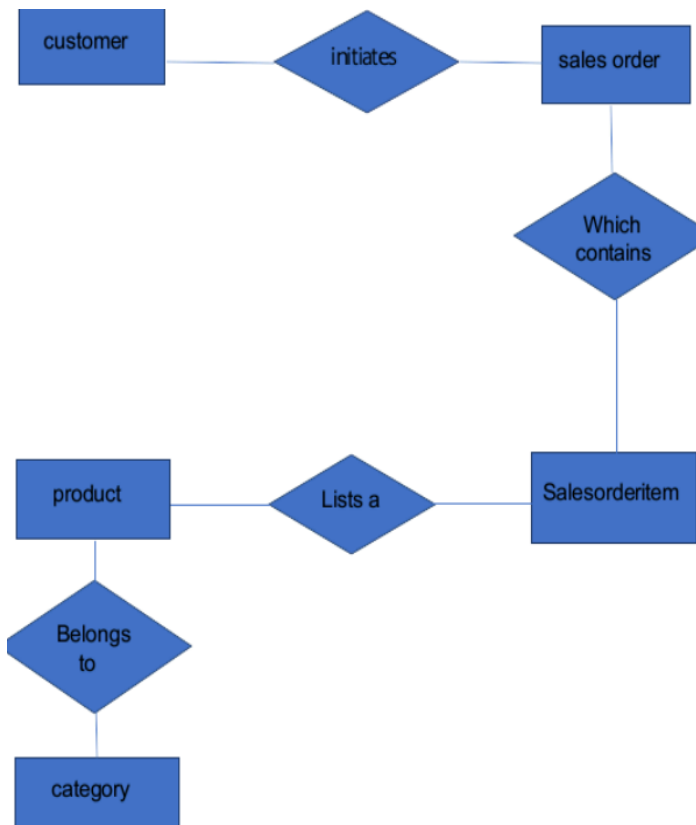


FIGURE.1.ENTITY RELATIONSHIP DIAGRAM

2.2 Sales order management application framework using visual studio

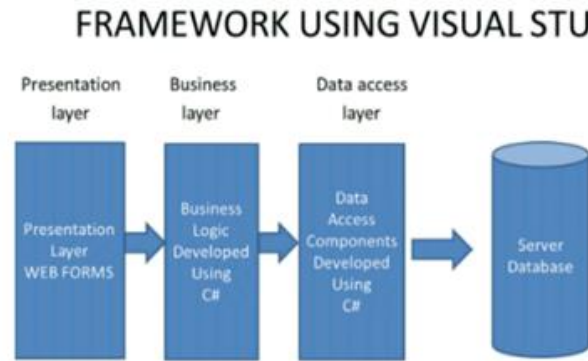


FIGURE.2 SOM application framework

3. CONCLUSIONS

In the development of this ERP package the entity relationship diagram is constructed successfully. By which the source data like customer details, bill of materials, sales data were collected successfully. After the source data collections the database is created and coding is developed in the c# (c sharp) language. In which we can able to develop the next version of the ERP package if it is necessary in future work. In this web architecture the data access layer is created using the c# (c sharp) language. Next the business layer and presentation layer should be created using the visual studio.net (2010) to make complete web architecture.

REFERENCES

- [1] A. Leon, Enterprise Resource Planning, Tata McGraw – Hill Education, 2007.
- [2] M.Ngadiman, et al., "Development of low cost Enterprise Resource Planning Prototype system for Malaysian small and medium industry using open source software," 2008.
- [3] [5] N.Athituv, et al ., "A system development methodology for ERP systems," Journal of Computer Information Systems, vol. 42, pp. 5667, 2002.
- [4] [14] I.Levent. (2008, 12 October). Java as ERP Programming Language Available.
- [5] [15] J.Park and N.Y.Lee, "A conceptual model of ERP for small and medium size companies Based on UML," International Journal of computer science and Network security, Vol.6, pp.42-49, 2006.