

ANALYSIS, DESIGN AND IMPLEMENTATION OF PRODUCT- BASED BUSINESS MANAGEMENT WITH INTEGRATED E-COMMERCE SUPPORT

By

Sharafat Ibn Mollah Mosharraf

Exam Roll: 1639

Registration No: HA-3766

&

Abdullah Al Mamun

Exam Roll: 1641

Registration No: HA-4275

A Project Submitted in Partial Fulfillment of the Requirements for the
Degree of Bachelor of Science in Computer Science and Engineering

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF DHAKA, DHAKA, BANGLADESH**

January 2011

APPROVAL

The Project Report “Analysis, Design and Implementation of Product-Based Business Management with Integrated E-Commerce Support” submitted by Sharafat Ibn Mollah Mosharraf Roll No: FH-01 and Abdullah Al Mamun Roll No: AE-55, to the Department of Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering and approved as to its style and contents.

Board of Examiners

1. (Supervisor)
2. (Examiner)
3. (Examiner)

DECLARATION

We, hereby, declare that the work presented in this Project is the outcome of the investigation performed by us under the supervision of Shahed Anwar, Assistant Professor, Department of Computer Science and Engineering, University of Dhaka, Dhaka, Bangladesh. We also declare that no part of this Project and thereof has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

Signature

.....

.....

Shahed Anwar

Sharafat Ibn Mollah Mosharraf

Supervisor

.....

Abdullah Al Mamun

Candidates

ABSTRACT

Most of the small and medium business organizations in the third-world countries do not yet use technology for their business management. But by computerization, they can save a lot of head-aches, responsibilities, errors and most valuably, time. Also, by making use of the Internet, the organizations can broaden their business scope. The goal of this project is to implement a common platform that can be customized and used by any product-based business organization according to its business needs so that it can overcome the aforementioned issues with little effort. For this project, product inventory and sales management as well as e-commerce support for processing online orders have been taken as the primary features to implement. The uniqueness of the project comparing with other solutions is that business owners do not have to handle or maintain a Business Management System and Online E-commerce website separately – it will be automatically maintained by a single integrated system.

ACKNOWLEDGMENTS

First of all, we are thankful and expressing our gratefulness to Almighty Allah who bestowed upon us his divine blessings, patience, mental and psychical strength to complete this project work. We are deeply indebted to and would like to express our gratitude for our project supervisor Shahed Anwar, Assistant Professor, Department of Computer Science and Engineering, University of Dhaka, for his encouragement, help and support for the completion of our project. His scholarly guidance gave us courage from the very beginning of the project. Last but not the least, we are highly grateful to our parents, family members and friends for their support and constant encouragement, which have always been a source of great inspiration for us.

TABLE OF CONTENTS

ABSTRACT	iv
ACKNOWLEDGMENTS	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	ix
LIST OF TABLES	xi
Chapter 1: Introduction	1
1.1 Overview	2
1.2 Product-Based Business	2
1.3 E-Commerce Systems	3
1.4 Existing Business Management Systems	3
1.5 Objectives of the Proposed System	3
1.6 Proposed Features of the System	4
1.7 Project Report Overview	5
1.8 Summary	6
Chapter 2: Background Studies	7
2.1 Introduction	8
2.2 Integrated Solutions	8
2.2.1 BizAutomation	8
2.2.2 NetSuite	9
2.3 Business Management Software	9
2.3.1 Databiz Inc. Highlights	9
2.3.2 inFlow Inventory	9
2.4 E-Commerce Solutions	9
2.5 Summary	10
Chapter 3: Tools & Technologies Used	11
3.1 Introduction	12
3.2 Desktop Software	12
3.2.1 Application Programming Language: Microsoft C# .NET 4.0	12
3.2.1.1 Reasons for Choosing C#	12
3.2.2 DBMS: Microsoft SQL Server 2008 R2 Express	13
3.2.2.1 Reasons for Choosing Microsoft SQL Server	13
3.2.3 ORM: Entity Framework 4.0	14
3.2.3.1 Reasons for Choosing Entity Framework	14
3.3 Web Application	15
3.3.1 Website UI Development: XHTML 1.0 Transitional and CSS 2.1.....	15
3.3.2 Server-Side Programming Language: PHP 5.3	16
3.3.2.1 Reasons for Choosing PHP	16
3.3.3 Client-Side Scripting Language: JavaScript 1.8	17

3.3.3.1	Reasons for Choosing JavaScript	17
3.3.4	DBMS: MySQL 5.3	17
3.3.4.1	Reasons for Choosing MySQL	18
3.3.5	ORM: Custom ORM Framework	18
3.3.6	Web Service: SOAP 1.2	18
3.3.6.1	Reasons for Choosing SOAP	19
3.4	Summary	19
Chapter 4: Project Analysis		20
4.1	Introduction	21
4.2	General Issues	21
4.2.1	General Product Setup	21
4.2.2	Desktop and Website Synchronization	22
4.3	Desktop Software-Specific Issues	23
4.3.1	Storing Hierarchical/Multi-Level Category Information in Database	23
4.3.2	Handling Different Stocks of the Same Product with Different Prices	26
4.3.3	Handling Delete Operation on Entities Which Has Associations with Other Entities	27
4.3.4	Inheriting Attributes of Products of a Category by Products of Its Descendant Categories	27
4.4	Web Application-Specific Issues	28
4.4.1	Handling Info of the Same Product for Different Organizations	28
4.5	Summary	29
Chapter 5: Project Design		30
5.1	Introduction	31
5.2	Database Schema Design	31
5.2.1	Desktop Software Part	31
5.2.1.1	Category and Product Setup	31
5.2.1.2	Stock Entry and Available Items in Stock	32
5.2.1.3	Product Sales to Customer	33
5.2.2	Web Application Part	33
5.2.2.1	Category and Product Setup	33
5.2.2.2	Relating Categories and Products with Organization	34
5.2.2.3	Product Purchase Order by Customer	34
5.3	Entity-Relationship Model (UML)	35
5.4	Schema Diagram	37
5.5	Data Dictionary (Table Definitions)	39
5.5.1	Desktop Software	39
5.5.2	Web Application	41
5.6	Summary	43

Chapter 6: Project Implementation	44
6.1 Introduction	45
6.2 Desktop Software	45
6.2.1 Typical Use-Case Scenario	45
6.2.2 Main Window	46
6.2.3 Category Management	47
6.2.4 Product Management	48
6.2.5 Stock (Inventory) Management	49
6.2.6 Sales Management	50
6.3 Web Application	51
6.3.1 Typical Use-Case Scenario	51
6.3.2 Home Page	52
6.3.3 Registration and Login	53
6.3.4 Browsing Product Categories	54
6.3.5 Products List in Categories	55
6.3.6 Product Details	56
6.3.7 Shopping Cart	57
6.3.8 Shopping Cart Checkout	58
6.4 Summary	59
Chapter 7: Security Considerations	60
7.1 Introduction	61
7.2 Prevention of Unauthorized Access	61
7.3 Prevention of SQL Injection Attack	61
7.3.1 Measures Taken to Prevent SQL Injection Attack	61
7.4 Prevention of XSS Attack	62
7.4.1 Measures Taken to Prevent XSS Attack	62
7.5 Prevention of HTTP Response Splitting Attack	62
7.5.1 Measures Taken to Prevent HTTP Response Splitting Attack	63
7.6 Prevention of CSRF/XSRF Attack	63
7.6.1 Measures Taken to Prevent CSRF Attack	64
7.7 Summary	64
Chapter 8: Conclusion and Future Work	65
8.1 Comparison to Existing Similar Products	66
8.2 Future Enhancements	67
8.3 Final Remarks	67
REFERENCES	68

LIST OF FIGURES

4.1	Sample UML diagram for product setup.	22
4.2	UML Diagram for Category in Adjacency Model.	23
4.3	UML Diagram of Category in Tree Traversal structure model.	24
4.4	Pre-order tree traversal algorithm to find multi-level data.....	24
4.5	UML diagram for stocks of the same product with varying prices.	26
4.6	UML diagram for handling same product for different organizations.	29
5.1	Database schema for category and product setup.	32
5.2	Database schema for Stock Management.	32
5.3	Database schema for Sales Management.	33
5.4	Relationships among categories, products and organizations.....	34
5.5	Database schema for product purchase orders by customers.....	34
5.6	Entity-Relationship Model for Desktop Software.	35
5.7	Entity-Relationship Model for Web Application.	36
5.8	Schema Diagram for Desktop Application.....	37
5.9	Schema Diagram for Web Application.	38
6.1	Typical use-case scenario of the desktop application.....	45
6.2	Main window of application.	46
6.3	Screenshot of Category Management window.	47
6.4	Screenshot of Category Properties Setup window.	47
6.5	Screenshot of Product Setup window.	48
6.6	Screenshot of Product List window.	48
6.7	Screenshot of New Stock Entry window.	49
6.8	Screenshot of Stock Entry List window.....	49
6.9	Screenshot of New Sales Order placement window.....	50
6.10	Typical use-case scenario of the web application.....	51
6.11	Screenshot of home page.	52

6.12 Screenshot of <i>Sign Up</i> page.	53
6.13 Screenshot of <i>Sign In</i> page.	53
6.14 Screenshot of category tree.....	54
6.15 Screenshot of product list in category.	55
6.16 Screenshot of product details.....	56
6.17 Screenshot of shopping cart.....	57
6.18 Screenshot of shopping cart checkout.....	58
6.19 Screenshot of payment through PayPal.	59

LIST OF TABLES

5.1	Table definition for table <i>Attribute</i>	39
5.2	Table definition for table <i>Category</i>	39
5.3	Table definition for table <i>Category_Attribute</i>	39
5.4	Table definition for table <i>Product</i>	39
5.5	Table definition for table <i>Product_Attribute</i>	40
5.6	Table definition for table <i>StockEntry</i>	40
5.7	Table definition for table <i>CurrentStock</i>	40
5.8	Table definition for table <i>SalesInvoice</i>	40
5.9	Table definition for table <i>SalesItem</i>	40
5.10	Table definition for table <i>Customer</i>	41
5.11	Table definition for table <i>attribute</i>	41
5.12	Table definition for table <i>category</i>	41
5.13	Table definition for table <i>category_attribute</i>	41
5.14	Table definition for table <i>orders</i>	41
5.15	Table definition for table <i>organization</i>	42
5.16	Table definition for table <i>organization_product</i>	42
5.17	Table definition for table <i>product</i>	42
5.18	Table definition for table <i>product_attribute</i>	42
5.19	Table definition for table <i>product_order</i>	43
5.20	Table definition for table <i>user</i>	43
5.21	Table definition for table <i>sessions</i>	43

CHAPTER 1

INTRODUCTION

1.1 Overview

Inventory, sales management and profit/loss calculation are the major concerns of a business organization. By automating these parts, the businessmen can always be updated with the status of the profit of their organizations and invest more time for the profit of the organization rather than managing files and written documents of purchases and sales as well as calculating profit/loss. Also, by automatic linking and synchronization of inventory and sales management software with an e-commerce website can spread the business worldwide and help increase profit – all without putting the businessmen in concern of managing all these stuffs.

1.2 Product-Based Business

There are two types of business organizations:

1. Service-Based – e.g. hospitals, audit firms etc.
2. Product-Based – e.g. shops selling products etc.

The proposed project is concerned with product-based business organizations. The following activities have been identified to be commonly practiced by such organizations:

1. **Inventory management:** It includes purchase of products from various vendors and stock-taking.
2. **Sales management:** This includes preparing sales invoice and reduction from stock.
3. **Profit/loss report:** It includes providing monthly, yearly reports or reports for any time interval.

1.3 E-Commerce Systems

E-commerce systems provide support for online shopping. The generic features of all product-based e-commerce websites can be identified as follows:

1. Product Listing and advertising.
2. Shopping cart for product orders by customers.
3. Integration with payment gateways to collect payment from customers for product sales.
4. Notifying the organization which owns the e-commerce website about customer orders so that the shipping process can be started.

1.4 Existing Business Management Systems

Notwithstanding the advancement of technology worldwide, many mid-scale to small-scale business organizations are still using the old-style written-documents to keep track of their business and manual calculations to determine profit and loss, especially in third-world countries including Bangladesh. There are many significant limitations, drawbacks and risks associated with this manual system. For example,

1. Documents and papers can be lost, damaged or destroyed anytime.
2. As humans are doing all the management and calculations, errors can naturally occur.
3. Lack of automation of the management process lets the sales persons make a way out to cheat and steal from the business.

1.5 Objectives of the Proposed System

The proposed system overcomes the limitations of manual business management process as well as adds more advantages to it, for example:

1. Reducing the time and complexity of keeping track of inventories and sales.
2. Error-free calculations.
3. Providing a great opportunity to increase the business and profit by leveraging the advantages of e-commerce.

1.6 Proposed Features of the System

The proposed project is an integrated software system consisting of two parts – a desktop software and a web application.

The desktop software part is primarily the complete customizable, full-featured business management software solution with support for managing orders from and synchronization of inventory with the e-commerce site. The software solution will contain the following features:

- Inventory management
- Sales processing and invoices
- Purchase orders and order processing
- Synchronization with the e-commerce part

The web part is the e-commerce part with the following features:

- Customer registration
- Product display
- Online shopping

1.7 Project Report Overview

The project report is organized into several chapters each describing a major aspect of the project.

Chapter 2 describes the background studies of the existing similar products – how those operate, their features, advantages and limitations.

Chapter 3 contains short discussions on the tools and technologies used to develop the proposed project and the reasons for preferring those to their alternatives.

Chapter 4 starts with the project-related issues. Several issues related to the implementation of the project are analyzed here and subsequently decisions for handling those issues are derived from the analyses.

Chapter 5 illustrates the design of the project software – both the desktop and the web part. Project designs include database schema design, entity-relationship models, schema diagrams and data dictionary (table definitions).

Chapter 6 describes the implementation of the project which includes detailed discussions accompanied with screenshots on various use cases and program flows.

Chapter 7 is presented with the security issues related to the project – their descriptions in short and how they are handled throughout the project.

Chapter 8 concludes the project report by presenting a discussion on the advantages of the proposed project compared to existing similar products, followed by pointing out some future enhancements on the project work.

1.8 Summary

To automate business management and spread business, an inventory and sales management software bundled with an e-commerce system is the main goal of the project. Although this product may not be high standard comparing to some existing products, it is highly suitable for mid and small-scale business organizations especially in third-world countries – as has been discussed in the final chapter. The following chapter discusses the project background and the preparation of the requirements for implementing the product functionalities.

CHAPTER 2

BACKGROUND STUDIES

2.1 Introduction

To develop a business management software, how a business is managed in reality must be learned beforehand. A few mid and small-scale business organizations were consulted to learn how they operate their business. Also, a few business management software and ecommerce websites were reviewed to see the features they offer. The outcome from this field work formed the features of the proposed project.

2.2 Integrated Solutions

There are a few business management software with integrated e-commerce application. During background studies, two such solutions were found to be noteworthy – BizAutomation and NetSuite.

2.2.1 BizAutomation

BizAutomation.com On-Demand is a hosted Business Management Software super-suite for an entire business. Often, Business Management Software focuses on only a few modules, but with BizAutomation.com, its business software solution for managing business includes CRM, ERP, E-commerce, and Accounting. Business Management software also includes Customer Relationship Management (CRM) online; Web based CRM, small business software for management for all departments, and online ERP & ERM. [1]

However, this solution is packed with lots of features most of which are not necessary for small-scale businesses. Also, as the solution is totally web-based, it is not well-suited to third-world countries as discussed in the final chapter of this project report (section 8.1).

2.2.2 NetSuite

NetSuite [2] is a similar solution like BizAutomation. However, it has some features less than BizAutomation.

NetSuite, too, has the same problems associated with BizAutomation and thus is not well-suited to the needs of the third-world business organizations.

2.3 Business Management Software

Among various desktop-based business management software studied, two are noteworthy – inFlow Inventory and Databiz Inc. Highlights.

2.3.1 Databiz Inc. Highlights

Highlights [3] is a complete business management solution designed for mid to large-scale businesses. It provides lots of features which are useful for large organizations, but configuring and using those features are quite complex. Moreover, it completely lacks the support for generalized product setup.

2.3.2 inFlow Inventory

inflow [4] is an inventory management software designed for small to mid-sized businesses. It has all the necessary features a business management system should have and it is easy to use. However, it, too, lacks support for generalized product setup.

2.4 E-Commerce Solutions

There are various commercial and open-source e-commerce applications available. All of those include business management systems as well. However,

as the desktop software of the proposed project handles these tasks, the e-commerce application would not include these. All the e-commerce applications include the features mentioned in section 1.3, among which product listing and notifying the owner organization about customer orders would be handled by synchronizing between the desktop software and the e-commerce website. (That is, the products would be added to inventory by means of the desktop software and later the e-commerce website will be updated with the product details through automatic synchronization.) The rest of the features (i.e. shopping cart and integration with payment gateways) would be handled by the proposed web application.

2.5 Summary

Identifying the common patterns in business management and e-commerce system functionalities provides the feature-set of the proposed project. The next step is to set up the features in a generalized manner so that they can be customized. The analysis work on how this can be achieved is described in detail in Chapter 4: Project Analysis. Before discussing the analysis, the next chapter is presented with a quick introduction to the tools and technologies decided to be used to implement the project.

CHAPTER 3

TOOLS & TECHNOLOGIES USED

3.1 Introduction

There are various tools and technologies available for developing software and website. Some are more efficient than others; some are better suited to specific applications than others. In this chapter, the chosen tools and technologies for developing the proposed project are discussed along with the reasons for choosing them.

3.2 Desktop Software

3.2.1 Application Programming Language:

Microsoft C# .NET 4.0

C# is a multi-paradigm programming language encompassing imperative, declarative, functional, generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within the .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270). C# is intended to be a simple, modern, general-purpose, object-oriented programming language. The most recent version is C# 4.0, which was released on April 12, 2010. [5]

3.2.1.1 Reasons for Choosing C#

The primary reason to choose C# over other object-oriented programming languages (e.g. Java, C++ etc.) is that it is a Microsoft product, and most business organizations in the world (especially in Bangladesh) use the Microsoft Windows operating system. Programs coded with C# integrate well with the native Windows operating system and thus more user-friendly programs can be developed aimed at Windows OS.

3.2.2 DBMS: Microsoft SQL Server 2008

R2 Express

Microsoft SQL Server is a relational model database server produced by Microsoft. Its primary query languages are T-SQL and ANSI SQL.

SQL Server 2008 R2 was released on April 21, 2010. SQL Server 2008 R2 adds certain features to SQL Server 2008 including a master data management system branded as Master Data Services, a central management of master data entities and hierarchies.

SQL Server Express Edition is a scaled down, free edition of SQL Server, which includes the core database engine. While there are no limitations on the number of databases or users supported, it is limited to using one processor, 1 GB memory and 4 GB database files (10 GB database files from SQL Server Express 2008 R2). The entire database is stored in a single *.mdf* file, and thus making it suitable for XCOPY deployment. [6]

3.2.2.1 Reasons for Choosing Microsoft SQL Server

Microsoft SQL Server integrates well with the Microsoft programming languages. Also, Express Edition is free of cost and 10GB is enough for small business organizations. Express Edition can be easily upgraded to commercial SQL Server with the growth of business which is important in the sense of scalability. So, as Microsoft C# .NET has been chosen as the programming language to implement the project, Microsoft SQL Server has been chosen as the database management system over other DBMS (e.g. MySQL, Oracle etc.)

3.2.3 ORM: Entity Framework 4.0

Object-relational mapping (ORM, O/RM, or O/R mapping) in computer software is a programming technique for converting data between incompatible type systems in object-oriented programming languages. This creates, in effect, a "virtual object database" that can be used from within the programming language.

Data management tasks in object-oriented (OO) programming are typically implemented by manipulating objects that are almost always non-scalar values. However, many popular database products such as structured query language database management systems (SQL DBMS) can only store and manipulate scalar values such as integers and strings organized within normalized tables. The programmer must either convert the object values into groups of simpler values for storage in the database (and convert them back upon retrieval), or only use simple scalar values within the program. Object-relational mapping is used to implement the first approach. [7]

ADO.NET Entity Framework (EF) is an object-relational mapping (ORM) framework for the .NET Framework. The second version of Entity Framework, named Entity Framework 4.0 (EFv4), was released as part of .NET 4.0 on 12 April 2010 and has addressed many of the criticisms made of version 1. [8]

3.2.3.1 Reasons for Choosing Entity Framework

EF is an ORM framework developed by Microsoft and is well-suited with the .NET programming languages. Similar to the reason of choosing SQL Server as the DBMS, EF has been preferred to other ORM frameworks (e.g. NHibernate) as C# .NET has been selected as the development programming language. Also, EF 4.0 has a better LINQ provider compared to contemporary ORM mapping solutions.

3.3 Web Application

3.3.1 Website UI Development: XHTML

1.0 Transitional and CSS 2.1

XHTML (eXtensible HyperText Markup Language) is a family of XML markup languages that mirror or extend versions of the widely used Hypertext Markup Language (HTML), the language in which web pages are written. XHTML was developed to make HTML more extensible and increase interoperability with other data formats. HTML 4 was ostensibly an application of Standard Generalized Markup Language (SGML); however the specification for SGML was complex, and neither web browsers nor the HTML 4 Recommendation were fully conformant to it. The XML standard, approved in 1998, provided a simpler data format closer in simplicity to HTML 4. By shifting to an XML format, it was hoped HTML would become compatible with common XML tools; servers and proxies would be able to transform content, as necessary, for constrained devices such as mobile phones.

XHTML 1.0 Transitional is the XML equivalent of HTML 4.01 Transitional, and includes the presentational elements (such as *center*, *font* and *strike*) excluded from the XHTML 1.0 Strict version. [9]

Cascading Style Sheets (CSS) is a style sheet language used to describe the presentation semantics (the look and formatting) of a document written in a markup language. Its most common application is to style web pages written in HTML and XHTML. CSS is designed primarily to enable the separation of document content (written in HTML or a similar markup language) from document presentation, including elements such as the layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple pages to share formatting, and reduce complexity and repetition in the structural content (such as by allowing for tableless web

design). CSS can also allow the same markup page to be presented in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. While the author of a document typically links that document to a CSS style sheet, readers can use a different style sheet, perhaps one on their own computer, to override the one the author has specified. The popularity of CSS as a design tool has increased steadily and is now regarded as the premier system for website design. [10]

3.3.2 Server-Side Programming Language:

PHP 5.3

PHP: Hypertext Preprocessor is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. As a general-purpose programming language, PHP code is processed by an interpreter application in command-line mode performing desired operating system operations and producing program output on its standard output channel. It may also function as a graphical application. PHP is available as a processor for most modern web servers and as a standalone interpreter on most operating systems and computing platforms. [11]

3.3.2.1 Reasons for Choosing PHP

PHP is the most widely used open-source server-side programming language. In contrast to ASP .NET, it can run on any operating system platform. As Linux servers are relatively cheap because of Linux being an open-source operating system, PHP has been chosen as it can run on Linux. Also, PHP has more community-support than other contemporary server-side

programming languages (e.g. CGI, Ruby etc.). In addition, writing applications in PHP is much easier which also makes it the preferred programming language over JSP.

3.3.3 Client-Side Scripting Language:

JavaScript 1.8

JavaScript is an implementation of the ECMAScript language standard and is typically used to enable programmatic access to computational objects within a host environment. It can be characterized as a prototype-based object-oriented scripting language that is dynamic, weakly typed and has first-class functions.

JavaScript is primarily used in the form of client-side JavaScript, implemented as part of a web browser in order to provide enhanced user interfaces and dynamic websites. However, its use in applications outside web pages — for example in PDF-documents, site-specific browsers and desktop widgets — is also significant. [12]

3.3.3.1 Reasons for Choosing JavaScript

JavaScript is supported by all major web browsers in contrast to other client-side scripting languages (e.g. JScript, VBScript etc.) which are primarily supported by only Microsoft Internet Explorer web browser.

3.3.4 DBMS: MySQL 5.3

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. Free-software projects that require a full-featured database management system often use MySQL. MySQL is a popular choice of database for use in web applications,

and is a central component of the widely-used LAMP web application software stack — LAMP is an acronym for “Linux, Apache, MySQL, PHP”. Its popularity is closely tied to the popularity of PHP. MySQL is used in some of the most frequently visited web sites on the Internet, including Flickr, Facebook, Wikipedia, and Google — though not for searches, Nokia.com and YouTube. [13]

3.3.4.1 Reasons for Choosing MySQL

As discussed above, MySQL is popular to use with PHP and Linux. As the chosen development programming language is PHP, hence MySQL has been chosen as the DBMS to use with PHP. Other RDBMS is not quite suitable for the proposed project. For example, Microsoft SQL Server runs only on Windows; Oracle is too complicated with security features, which are not necessary for the proposed project.

3.3.5 ORM: Custom ORM Framework

A self-implemented ORM framework has been used in the project. It is partly automated and partly manual-config.d. Existing ORM frameworks (e.g. Doctrine, Propel etc.) are quite complex to use and is thus avoided for implementing the proposed project.

3.3.6 Web Service: SOAP 1.2

SOAP is a remote procedure call (RPC) protocol which uses XML to encode its calls and HTTP as a transport mechanism. It is used as a means of providing web services to any client which needs the services.

3.3.6.1 Reasons for Choosing SOAP

In comparison to REST, where resources are transported, SOAP is designed to call methods.

SOAP is simpler more robust than XML-RPC because it

- defines multiple different encodings whereas XML-RPC allows only one method of method serialization
- has a robust security model
- requires the creation of WSDL service descriptions, which ensures the correctness and well-formation of the XML data. [14]

3.4 Summary

Along with the technologies described above, some other tools have been used to complete the development environment of the project. Visual Studio .NET 2010 has been used as the IDE (Integrated Development Environment) for developing the desktop software, and NetBeans IDE for developing the web application. Also, Apache Web Server has been used as the development and deployment web server. The following chapter discusses the various analytical aspects of the proposed project and decisions derived from those analyses.

CHAPTER 4

PROJECT ANALYSIS

4.1 Introduction

The biggest challenge in implementing the proposed project is to develop it in a generalized way so that it can be easily customized according to the business needs of each organization. Analysis of the domain to solve this issue as well as some other issues is discussed thoroughly in this chapter.

4.2 General Issues

There are a few issues common in both desktop and web parts. Both parts must be able to handle those issues effectively.

4.2.1 General Product Setup

The Problem

How can the properties of different types of products be managed so that any business organization can create and manage the properties of its products easily?

Analysis of the Sample Data Collected

1. Properties are diversified for different products.
2. If we do not consider the fixed properties (id, name, price etc.), then a product may have 0 to N (N is a finite number) number of properties.
3. The same property may apply to different products. For example, the property *Data Size* is applicable to Hard Disk Drives as well as CD-ROM Drives. However, the value of the property may be the same or different.

Proposed Solution

A product can have multiple properties. On the other hand, a property can be associated to multiple products. However, the value of the property might be different for each product.

Sample UML Diagram

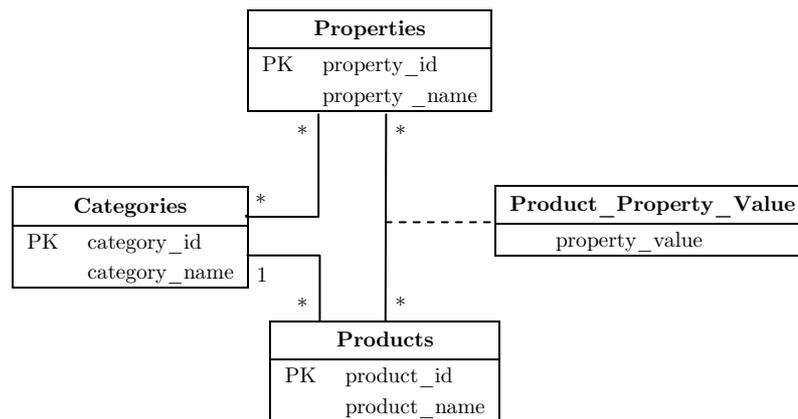


Fig. 4.1: Sample UML diagram for product setup.

4.2.2 Desktop and Website Synchronization

The Problem

How would the website be updated automatically with new product details and current stock? How would the desktop software get notifications automatically on online purchase of a product?

Proposed Solution

A web service would be provided to synchronize between the desktop software and the website. Whenever any change occurs in the database of the desktop software, the software would pass it to the website by using web services. Similarly, the software would periodically poll the website through web services to find out if any online purchase order is pending.

4.3 Desktop Software-Specific Issues

Issues related to the desktop application have been discussed in this section.

4.3.1 Storing Hierarchical/Multi-Level Category Information in Database

The Problem

Category of products often comes in a hierarchical manner. A category might have some sub-categories, and those sub-categories in turn might also have more sub-categories. As relational database doesn't have the implementation of inheritance or hierarchical data storing procedure, a problem arises on storing this type of data.

Solution with Adjacency List Model

One solution is to use Adjacency List Model. In this model, every category has an associated parent category. This parent category refers to some other category as a foreign key in the same entity. The Root category has null value as its parent category.

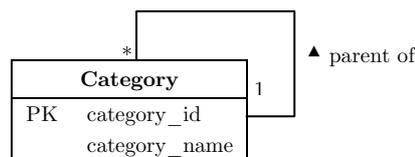


Fig. 4.2: UML Diagram for Category in Adjacency Model.

The advantage of this approach is Simplicity. No complex structure is present, thus easy to understand and the code it needs is simple, too.

But there are some great disadvantages. When it needs to know all the multilevel subcategories of a specific category or the path from root to selected category, then a recursive function and multiple queries are needed. As each

query takes some time, this makes the function very slow when dealing with large trees.

Proposed Solution with Modified Preorder Tree Traversal:

Modified Preorder Tree Traversal algorithm is the solution to the disadvantage of the Adjacency list model. Every node of a tree has two values associated with it – a left value and a right value.

Category	
PK	category_id
	category_name
	left
	right

Fig. 4.3: UML Diagram of Category in Tree Traversal structure model.

All subcategories under a category must have value (left and right) in between its parent’s left and right value. An example is shown below:

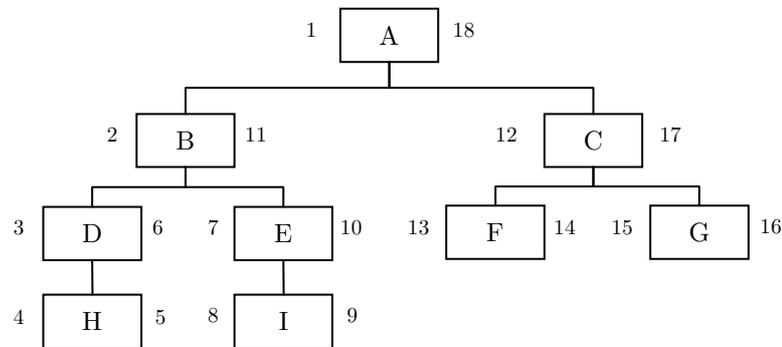


Fig. 4.4: Pre-order tree traversal algorithm to find multi-level data.

As it can be seen, left and right numbers indicate the relationship between each node.

In the above example, **B** has sub categories – **D**, **E**, **H**, **I** and the left and right values of these sub-categories are in-between the left and right values of **B**. So, when all sub categories of **B** is needed, a query like the following will yield the expected result:

```
SELECT * FROM tree WHERE left BETWEEN 2 AND 11;
```

Also, when it needs to know the path from the root to a node, for example from root **A** to **H**, a single query can give the result:

```
SELECT * FROM tree WHERE left <= 4 AND right >= 5 ORDER BY left ASC;
```

Inserting data in the tree is also simple. For adding a new data in the tree, it needs 2 extra queries besides the regular *insert* query. As an example, suppose, **J** will be added as a sub-category of **H**. Then, first all other nodes – which have left values greater than the left value of **H** – their left and right values need to be updated by adding 2.

```
UPDATE tree SET right = right + 2 WHERE right > 4;
```

```
UPDATE tree SET left = left + 2 WHERE left > 4;
```

Now **J** can be added as a new node. This node has the values left = 5 and right = 6.

```
INSERT INTO tree SET left = 5, right = 6, title = 'J';
```

The same procedure in reverse can be applied when a node needs to be deleted from the tree. (i.e., first all left and right values of nodes which have as its left value greater than the left value of **H** needs to be decreased by two, and then **J** should be removed.)

4.3.2 Handling Different Stocks of the Same Product with Different Prices

The Problem

A stock entry contains product purchase information with the quantity of products purchased and unit purchase price. But when another (new) stock of the same product is purchased with different quantity and different purchase price, how can different stocks be combined together?

Proposed Solution

Stock Entry and Current Stock will be considered to be different independent entities. Stock Entry entity contains only purchase information with purchased quantity and unit purchase price, while Current Stock maintains the total number of products available in stock. So, when a new entry is added to Stock Entry, the newly purchased product quantity will also be added to the total number of available products information in Current Stock entity. When a sale of that product occurs, corresponding product quantity will be reduced from the total available product information in Current Stock entity.

Sample UML Diagram

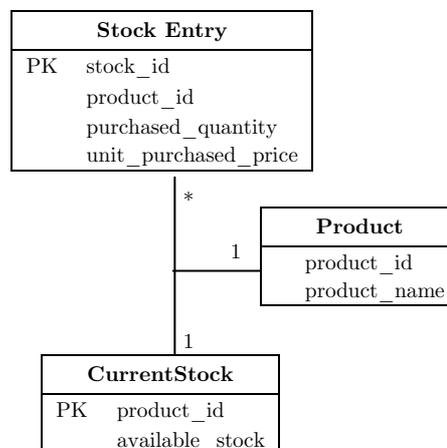


Fig. 4.5: UML diagram for handling different stocks of the same product with varying prices.

4.3.3 Handling Delete Operation on Entities Which Has Associations with Other Entities

The Problem

There might be a situation where a user needs to delete an entry from an entity, like Category, Product etc. That's because the user might want to discontinue that specific product's business or upgrade to a new product. But when a user deletes an entity like Product, which has an association with Stock and Sales, applying cascading delete would cause the stock and sales record to be deleted as well. This would produce dubious profit-loss report and other side-effects.

Proposed Solution

This type of problem can be handled by introducing an extra field in the database table, e.g. a *status* field. When a category or product is deleted, then its association with other entities should be checked. If any association can be found, then just the status bit is set to *deleted* and hence will remain out of view of user. If no association of it is found with other data, then it can be safely deleted.

4.3.4 Inheriting Attributes of Products of a Category by Products of Its Descendant Categories

If some attributes are associated with a category, then its subcategories would also have those attributes associated by inheritance, no new associations need to be made. For example, suppose that the category *Computers* has the *Manufacturer* attribute associated. If *Desktops* and *Laptops* are two subcategories of *Computer*, then those will also have the *Manufacturer*

attribute associated, no separated association with the attribute needs to be stored in the database.

4.4 Web Application-Specific Issues

Issues related to the web application has been discussed in this section.

4.4.1 Handling Info of the Same Product for Different Organizations

The Problem

There will be a single database for the websites of all the organizations. Then, as the stock and selling price of the same product would differ among organizations, how can these be handled in a single database?

Information Found From Analysis

1. Organizations may share the same product, but the product price and current stock can be different.
2. Quantities of the product sold are also different for each organization. This attribute is needed to list the best selling products of each organization.

Proposed Solution

Products and Organizations are considered to be different independent entities. An association of a product to an organization would be accompanied by the attributes *stock*, *unit selling price* and *quantities sold*.

Sample UML Diagram

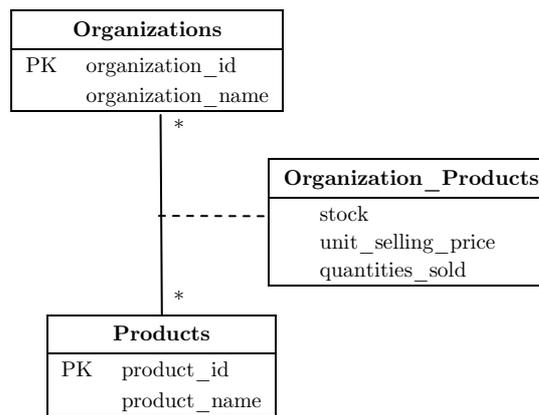


Fig. 4.6: UML diagram for handling info of the same product for different organizations.

4.5 Summary

Solving the issues of general product setup, automatic synchronization between the desktop software and the website and such are vital for the implementation of the proposed project. Once the issues are resolved, the overall design of the project and its implementation can be performed quite easily. The next few chapters provide the details of the design and implementation of the project features.

CHAPTER 5

PROJECT DESIGN

5.1 Introduction

Every software product needs to be designed well before implementation so that in the middle of coding no flaw in project organization pops-up and lots of changes need to be made to fix it. In this chapter, the features of the proposed project are discussed in detail and related database schemas are provided. At the end of the chapter, the entity-relationship model, schema diagram and data dictionary for the overall project are also presented for a glimpse of the project design.

5.2 Database Schema Design

To implement each feature, initially only the related database schema is designed for that particular feature. Following is the list of features with corresponding database schemas represented in UML notation.

5.2.1 Desktop Software Part

This section describes the schema design specific to the desktop software.

5.2.1.1 Category and Product Setup

1. A category can contain multiple products.
2. Multiple attributes can be defined for each category.
3. When a product is associated with a particular category, then the values of the attributes defined for that category should be stored for that particular product.

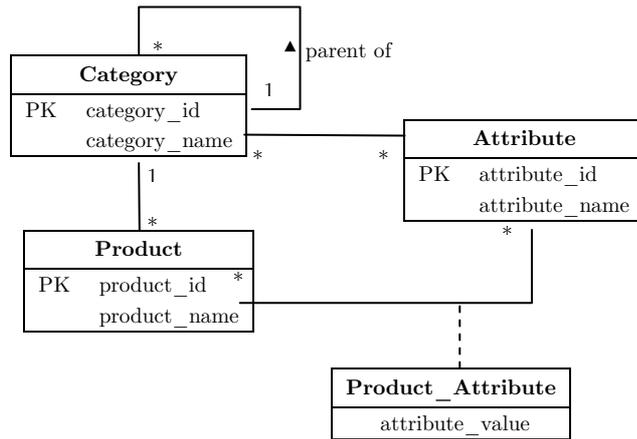


Fig. 5.1: Database schema for category and product setup.

5.2.1.2 Stock Entry and Available Items in Stock

1. Each product has many stock entries associated with it.
2. As multiple stocks of the same product can exist simultaneously, therefore, the available number of items in all the stock entries of a product is recorded in another table named ‘CurrentStock’.

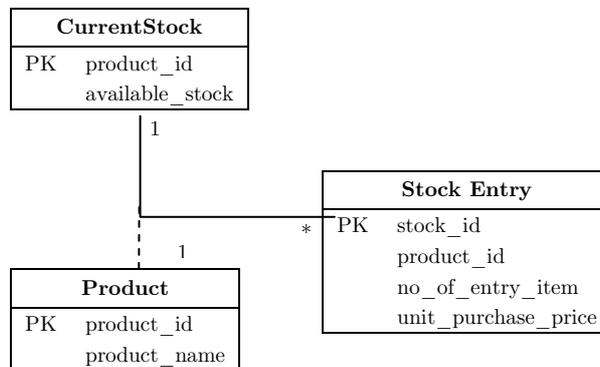


Fig. 5.2: Database schema for Stock Management.

5.2.1.3 Product Sales to Customer

1. A sales invoice can contain multiple product items.
2. Each product item can be sold in different quantities.
3. Selling an item will decrease the item quantity from CurrentStock table.

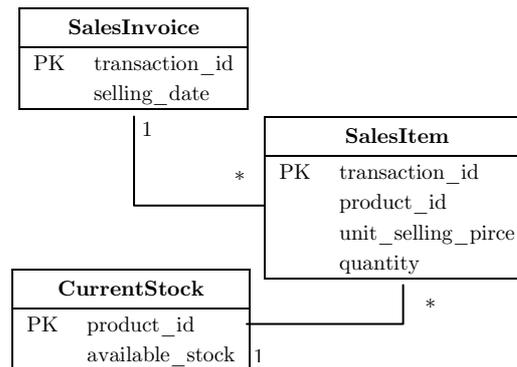


Fig. 5.3: Database schema for Sales Management.

5.2.2 Web Application Part

This section describes the database schema design issues specific to the web application.

5.2.2.1 Category and Product Setup

This is exactly the same as the category and product setup in desktop application part (section 5.2.1.1).

5.2.2.2 Relating Categories and Products with Organization

An organization can set up its own categories and assign products in those categories.

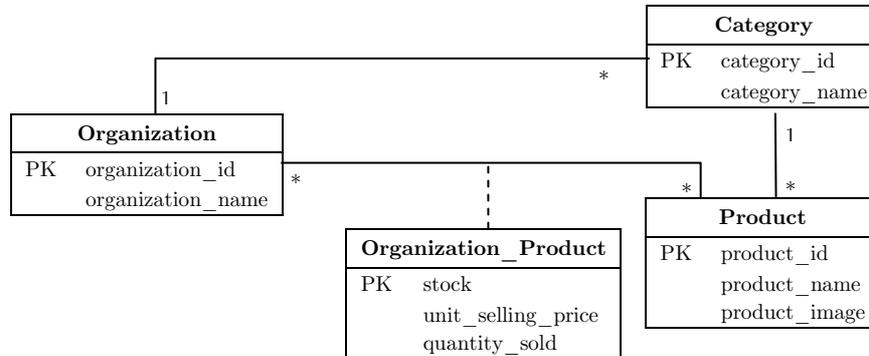


Fig. 5.4: Relationships among categories, products and organizations.

5.2.2.3 Product Purchase Order by Customer

1. A customer/user can place orders for products.
2. Each order can consist of multiple products with varying quantities.
3. A product price can change (i.e. new price update from desktop software synchronizes with the web application) between the time of checkout and the time of payment. So, to keep consistency, the price at the time of making the order should also be recorded with the order.

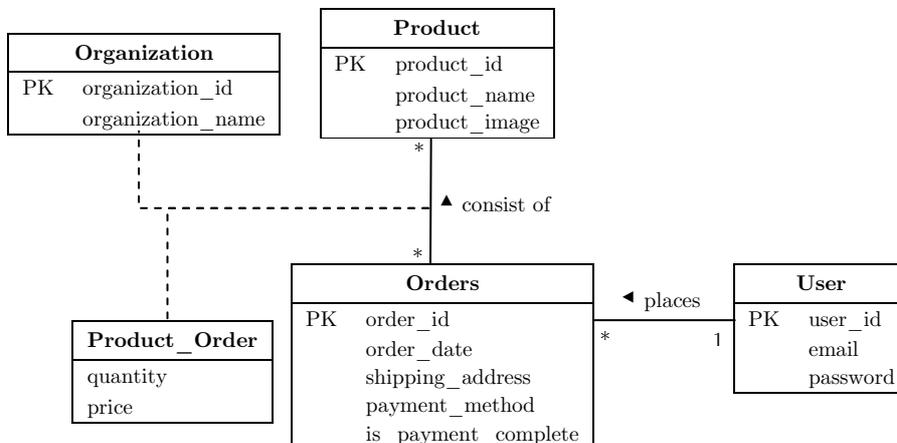


Fig. 5.5: Database schema for product purchase orders by customers.

5.3 Entity-Relationship Model (UML)

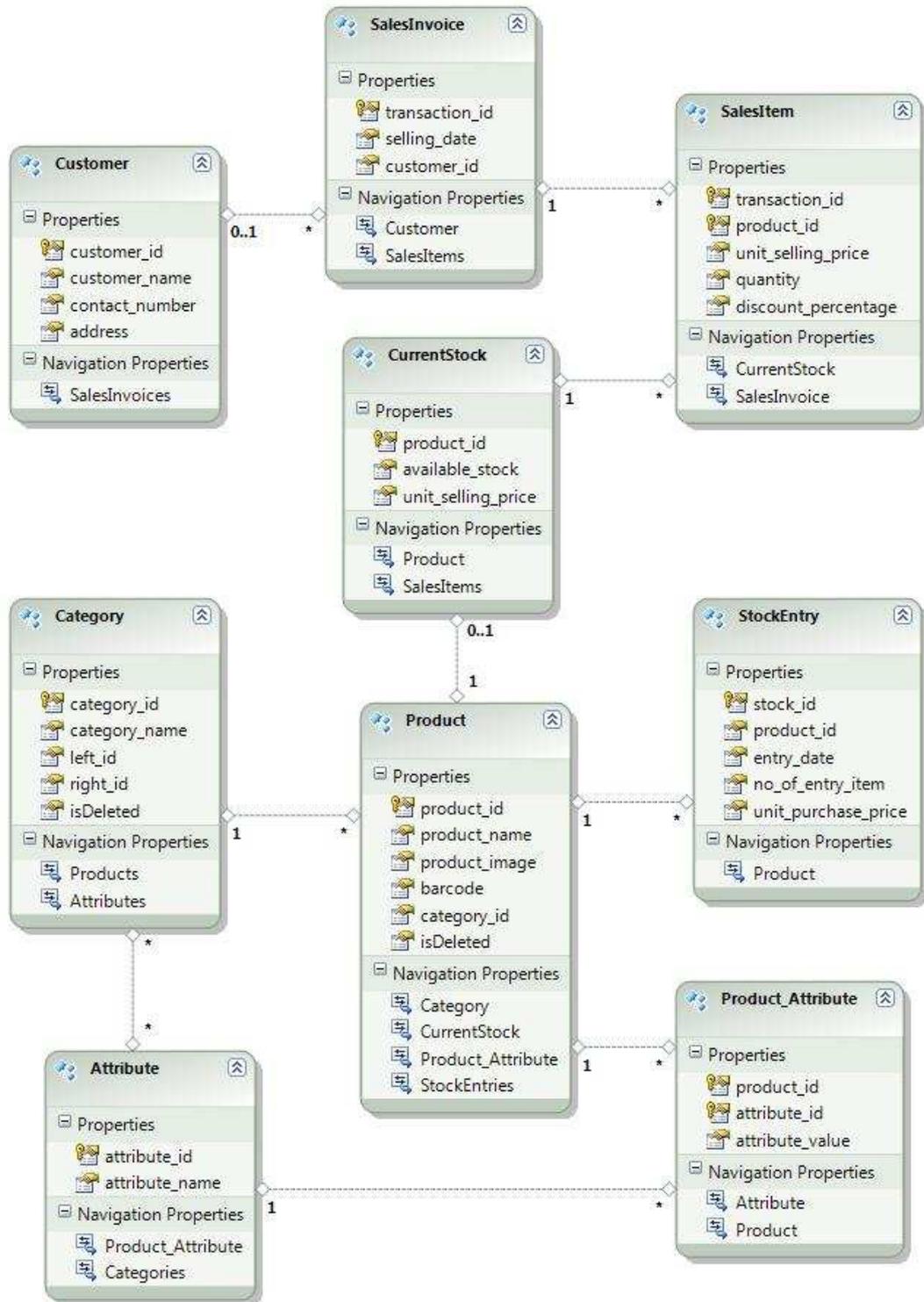


Fig. 5.6: Entity-Relationship Model for Desktop Software.

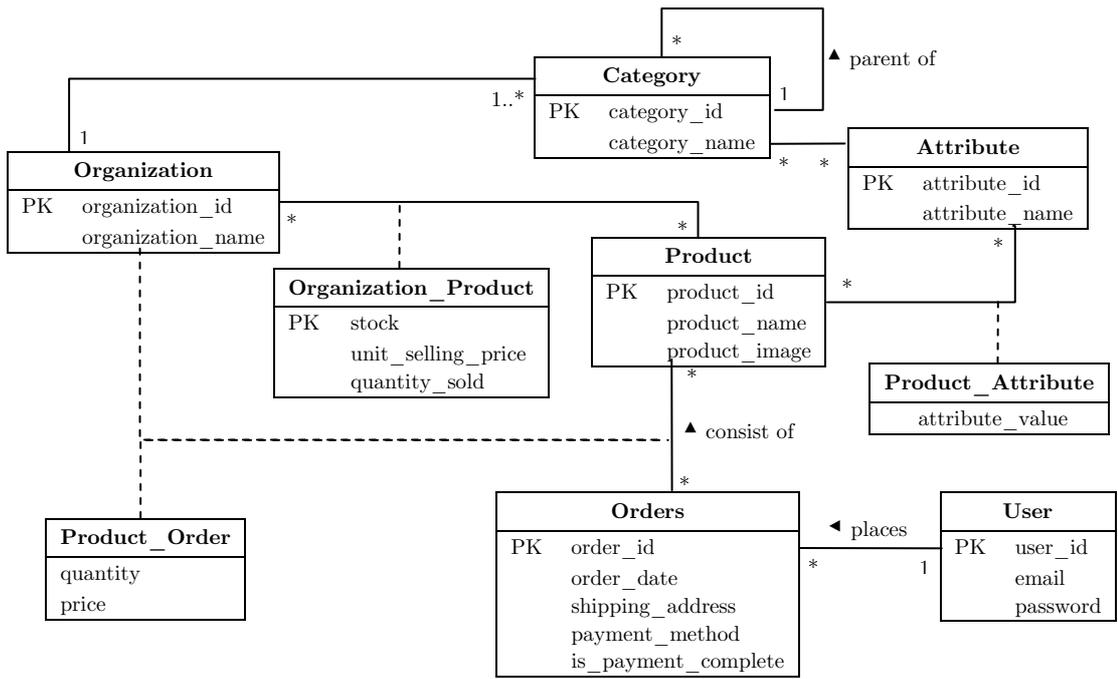


Fig. 5.7: Entity-Relationship Model for Web Application.

5.4 Schema Diagram

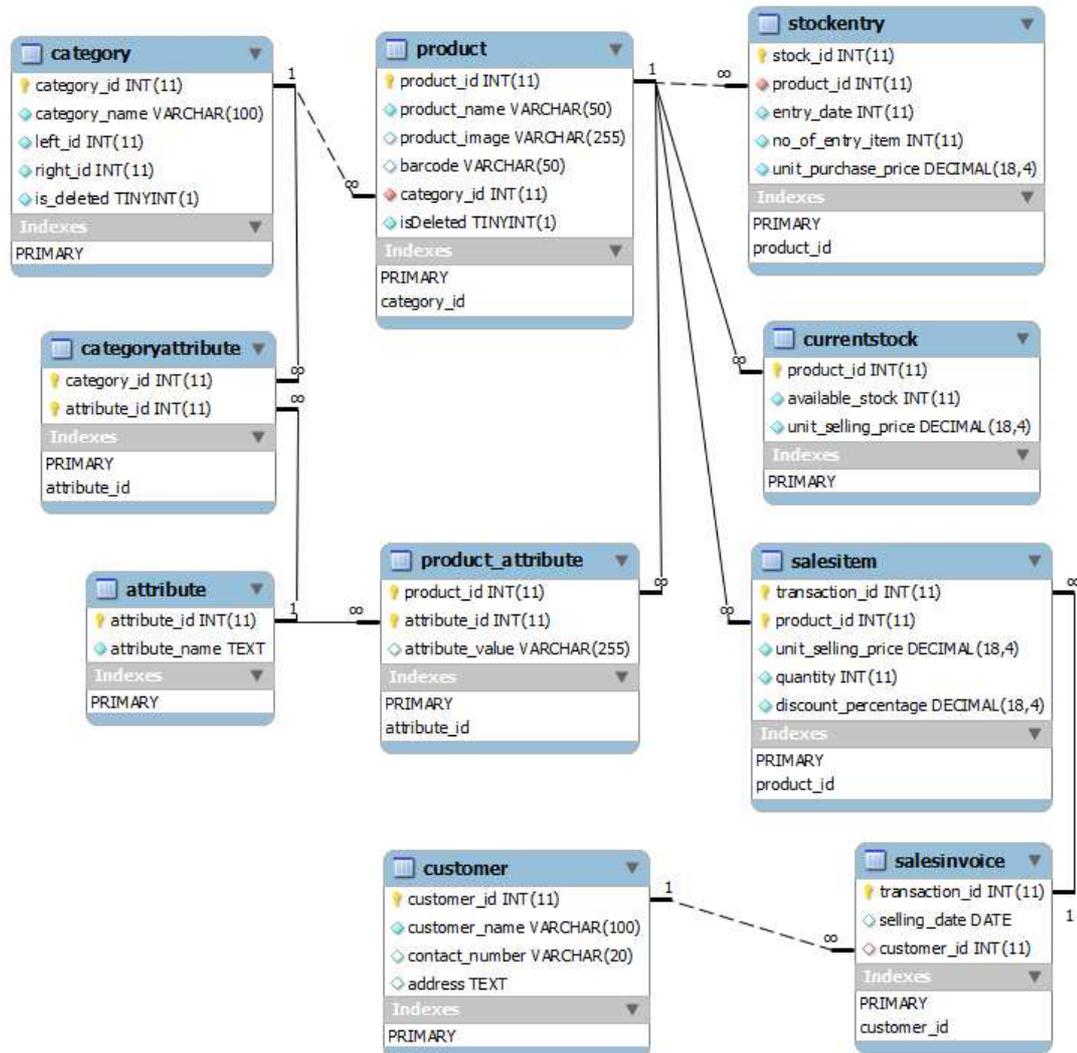


Fig. 5.8: Schema Diagram for Desktop Application.

5.5 Data Dictionary (Table Definitions)

The database table structures are have been shown in this section.

5.5.1 Desktop Software

Table 5.1: Table definition for table *Attribute*.

Field	Type	Null	Default
<u>attribute_id</u>	int(10)	No	
attribute_name	varchar(255)	No	

Table 5.2: Table definition for table *Category*.

Field	Type	Null	Default	Links to
<u>category_id</u>	int(10)	No		
category_name	varchar(255)	No		
left_id	int(10)	No		
right_id	int(10)	No		
isDeleted	bool	No	false	

Table 5.3: Table definition for table *Category_Attribute*.

Field	Type	Null	Default	Links to
<u>attribute_id</u>	int(10)	No		attribute → attribute_id
<u>category_id</u>	int(10)	No		category → category_id

Table 5.4: Table definition for table *Product*.

Field	Type	Null	Default	Links to
<u>product_id</u>	int(10)	No		
category_id	int(10)	No		category → category_id
product_name	varchar(255)	No		
product_image	varchar(255)	Yes		
barcode	varchar(50)	Yes		
isDeleted	bool	No	false	

Table 5.5: Table definition for table *Product_Attribute*.

Field	Type	Null	Default	Links to
<u>product_id</u>	int(10)	No		product → product_id
<u>attribute_id</u>	int(10)	No		attribute → attribute_id
attribute_value	varchar(MAX)	No		

Table 5.6: Table definition for table *StockEntry*.

Field	Type	Null	Default	Links to
<u>stock_id</u>	int(10)	No		
product_id	int(10)	No		product → product_id
entry_date	date	No		
no_of_entry_item	int(10)	No		
unit_purchase_price	decimal(18, 4)	No		

Table 5.7: Table definition for table *CurrentStock*.

Field	Type	Null	Default	Links to
<u>product_id</u>	int(10)	No		product → product_id
available_stock	int(10)	No		
unit_selling_price	decimal(18, 4)	Yes		

Table 5.8: Table definition for table *SalesInvoice*.

Field	Type	Null	Default	Links to
<u>transaction_id</u>	int(10)	No		
selling_date	datetime	No		
customer_id	int(10)	Yes		

Table 5.9: Table definition for table *SalesItem*.

Field	Type	Null	Default	Links to
<u>transaction_id</u>	int(10)	No		
product_id	int(10)	No		product → product_id
unit_selling_price	decimal(18, 4)	No		
quantity	int(10)	No		
discount_percentage	double(18,4)	Yes	0.0	

Table 5.10: Table definition for table *Customer*.

Field	Type	Null	Default	Links to
<u>customer_id</u>	int(10)	No		
customer_name	varchar(100)	No		
contact_number	varchar(20)	Yes		
address	text	Yes		

5.5.2 Web Application

Table 5.11: Table definition for table *attribute*.

Field	Type	Null	Default
<u>attribute_id</u>	int(10)	No	
attribute_name	varchar(255)	No	

Table 5.12: Table definition for table *category*.

Field	Type	Null	Default	Links to
<u>category_id</u>	int(10)	No		
category_name	varchar(255)	No		
parent_category_id	int(10)	Yes	<i>NULL</i>	category → category_id
<u>organization_id</u>	int(10)	No		organization → organization_id

Table 5.13: Table definition for table *category_attribute*.

Field	Type	Null	Default	Links to
<u>attribute_id</u>	int(10)	No		attribute → attribute_id
<u>category_id</u>	int(10)	No		category → category_id

Table 5.14: Table definition for table *orders*.

Field	Type	Null	Default	Links to
<u>order_id</u>	int(10)	No		
order_date	datetime	No		
user_id	int(10)	No		user → user_id

shipping_address	text	No		
payment_method	varchar(255)	No		
is_payment_complete	tinyint(1)	No	0	

Table 5.15: Table definition for table *organization*.

Field	Type	Null	Default
<u>organization_id</u>	int(10)	No	
organization_name	varchar(255)	No	

Table 5.16: Table definition for table *organization_product*.

Field	Type	Null	Default	Links to
<u>organization_id</u>	int(10)	No		organization →
<u>product_id</u>	int(10)	No		product → product_id
category_id	int(10)	No		category → category_id
stock	int(10)	Yes	0	
unit_selling_price	double(100,2)	Yes	<i>NULL</i>	
quantity_sold	int(10)	No	0	

Table 5.17: Table definition for table *product*.

Field	Type	Null	Default
<u>product_id</u>	int(10)	No	
product_name	varchar(255)	No	
product_image	varchar(255)	Yes	<i>NULL</i>

Table 5.18: Table definition for table *product_attribute*.

Field	Type	Null	Default	Links to
<u>product_id</u>	int(10)	No		product → product_id
<u>attribute_id</u>	int(10)	No		attribute → attribute_id
attribute_value	text	No		

Table 5.19: Table definition for table *product_order*.

Field	Type	Null	Default	Links to
<u>order_id</u>	int(11)	No		orders → order_id
<u>product_id</u>	int(11)	No		product → product_id
quantity	int(11)	No		
unit_price	float	No		

Table 5.20: Table definition for table *user*.

Field	Type	Null	Default	Links to
<u>user_id</u>	int(10)	No		
email	varchar(255)	No		
password	varchar(255)	No		
organization_id	int(10)	No		organization → organization_id

Table 5.21: Table definition for table *sessions*.

Field	Type	Null	Default	Links to
<u>session_id</u>	char(32)	No		
time_created	timestamp	No	0000-00-00 00:00:00	
last_updated	timestamp	No	0000-00-00 00:00:00	
session_data	longtext	No		
user_id	int(11)	Yes	<i>NULL</i>	user → user_id

5.6 Summary

The design presented in this chapter is flexible enough to accommodate new features. The coding of the project is also done bearing in mind the addition of new features. The next chapter provides the implementation of the project.

CHAPTER 6

PROJECT IMPLEMENTATION

6.1 Introduction

The implementation of the proposed project is discussed in this chapter. For each part (desktop and web), a complete use-case scenario has been depicted. In other words, how a user would use the software or website has been shown in a step-by-step process accompanied with brief discussions for each step.

6.2 Desktop Software

Using the desktop software is quite straightforward. A typical use-case scenario has been presented below followed by screenshots of the features of the software.

6.2.1 Typical Use-Case Scenario

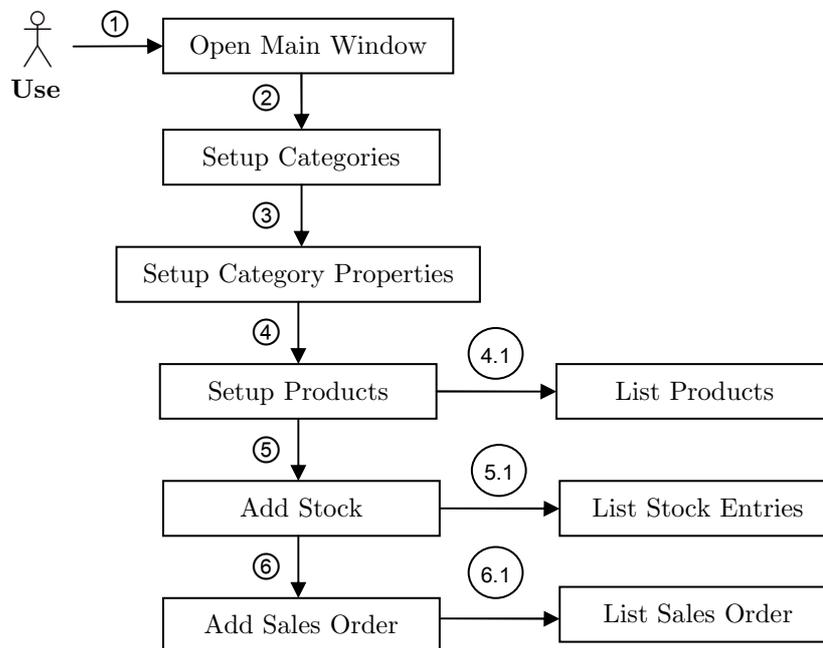


Fig. 6.1: Typical use-case scenario of the desktop application.

6.2.2 Main Window

The main window of the application shows the user to choose from the following available tasks:

- 1) Manage Category and Category Properties
- 2) Setup New Product and assigning values to corresponding Product Properties
- 3) Create, manage and view list of Stocks of Products
- 4) Place Sales Order and Display List of Sales Order

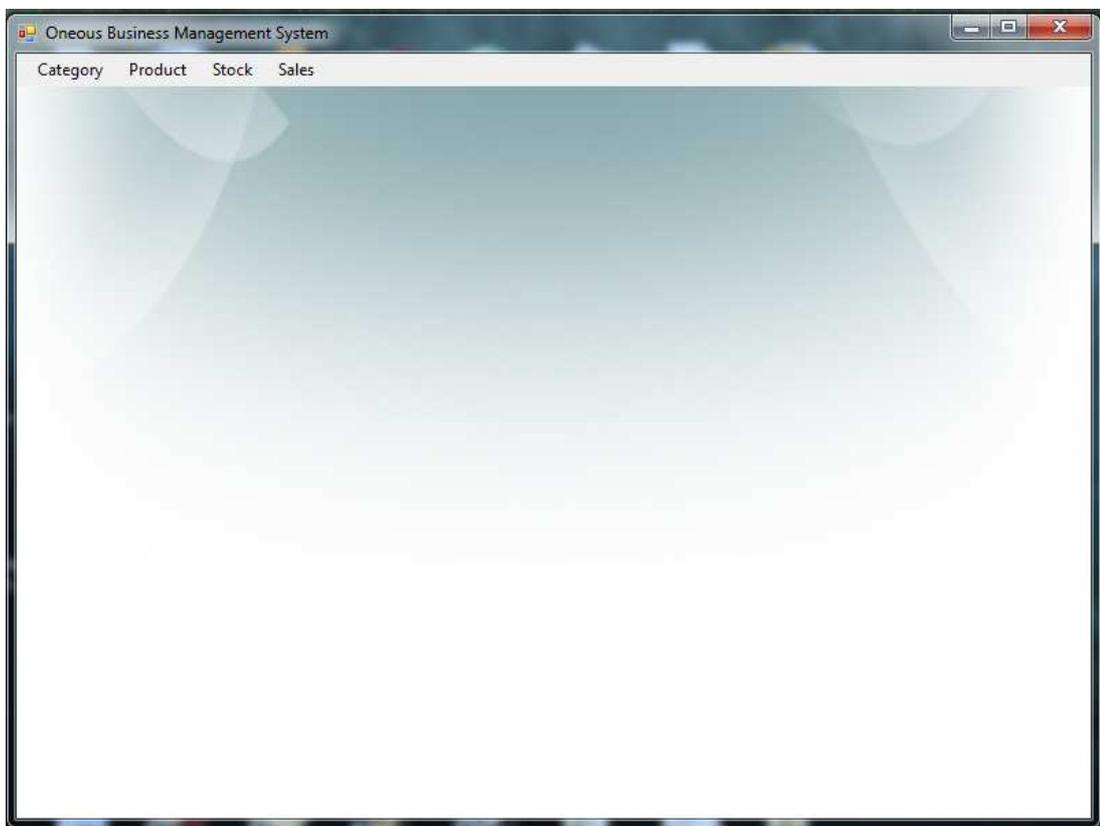


Fig. 6.2: Main window of application.

6.2.3 Category Management

Before adding products and sales details, category and its properties should be well defined.

By clicking on **Category** menu and then on **Manage Category** submenu, the user will see existing categories as a tree list. The user can directly create or delete category from the tree.



Fig. 6.3: Screenshot of Category Management window.

By clicking on **Category** and then on **Setup Category Properties**, the user can provide a set of properties for the category. For details on how this works, please refer to the analysis in section 4.2.1 and the design decision in section 5.2.2.1.

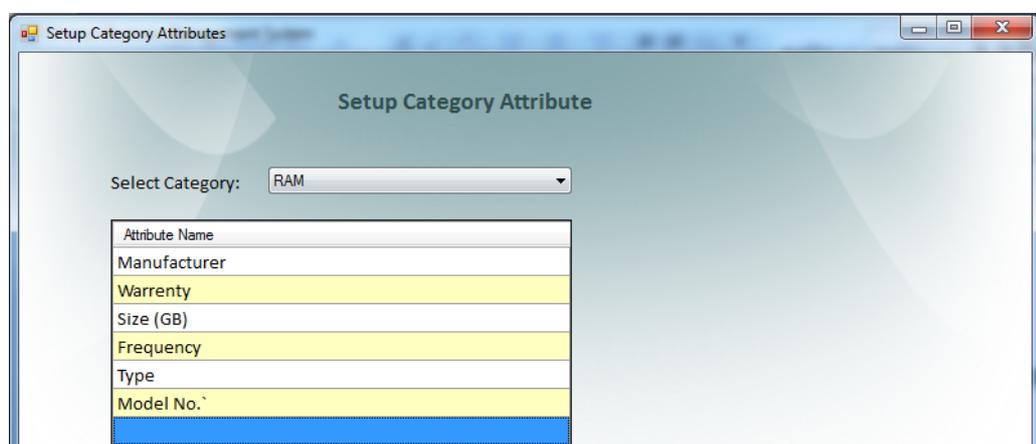
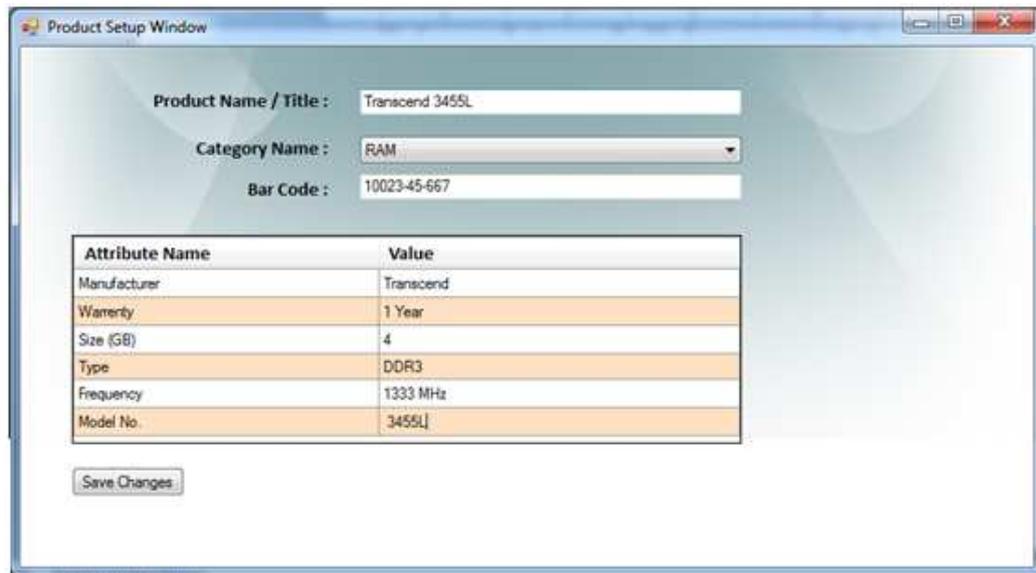


Fig. 6.4: Screenshot of Category Properties Setup window.

6.2.4 Product Management

Product setup is the next step after creating Category and Category Properties. The details of a product can be defined through product setup.

By Clicking **Product** and then **Setup a New Product**, a user can create a new product under appropriate category and assign appropriate values to its properties.



The screenshot shows a 'Product Setup Window' with the following fields and table:

Product Name / Title : Transcend 3455L
Category Name : RAM
Bar Code : 10023-45-667

Attribute Name	Value
Manufacturer	Transcend
Warranty	1 Year
Size (GB)	4
Type	DDR3
Frequency	1333 MHz
Model No.	3455L

Save Changes

Fig. 6.5: Screenshot of Product Setup window.

By Clicking **Product** and then **List of Product**, a user can view existing product list with detailed information.



The screenshot shows a 'List Of Products' window with a table titled 'Product List' containing the following data:

Product Name	Category	Bar Code
Acer Extensa 4230	Acer	01101102200
Lenovo ThinkPad 3544	Lenovo	112244332
Dell Inspiron 4433	Dell	112244332
Lenovo Idea Pad 3243	Lenovo	
Lenovo 7688	Lenovo	

Fig. 6.6: Screenshot of Product List window.

6.2.5 Stock (Inventory) Management

Stock/Inventory Management is one of the most important aspects of business management. Using this application, a user can add stocks of products and view as well as manage/adjust the current available stock.

By clicking **Stock** and then **New Stock Entry**, a user can add new stock entry of a product with product quantity, purchase price etc.



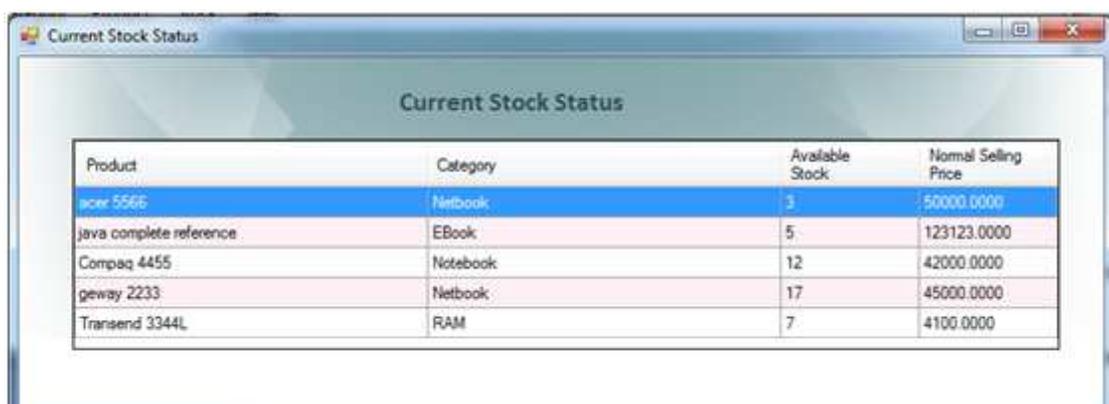
The screenshot shows a window titled "New Stock Entry" with a sub-heading "Add New Stock". The form includes the following fields:

- Stock Entry Date: Saturday, January 08, 2011
- Category: Hard Disk
- Product: Transcend USB J25M
- Quantity: 30
- Unit Purchase Price: 5000

A "Submit New Stock" button is located at the bottom of the form.

Fig. 6.7: Screenshot of New Stock Entry window.

A user can view the Stock Entry list by clicking **Stock** and then **Stock Entry List**. Users can also view the Current Stock status by clicking **Stock** and then **Current Stock**.



The screenshot shows a window titled "Current Stock Status" with a sub-heading "Current Stock Status". It displays a table with the following data:

Product	Category	Available Stock	Normal Selling Price
acer 5566	Notebook	3	50000.0000
java complete reference	EBook	5	123123.0000
Compaq 4455	Notebook	12	42000.0000
geway 2233	Notebook	17	45000.0000
Transcend 3344L	RAM	7	4100.0000

Fig. 6.8: Screenshot of Stock Entry List window.

6.2.6 Sales Management

Keeping track the sales order and processing orders from customer is another most important part of a business management system. A new sales order can be placed by clicking **Sales** and then **New Sales Order**.

In this window, customer name can be selected based on previously recorded customer information; or even new customer name, contact and address can be added from the same option. For new customer, his information will be added to customer database. Items are displayed as a list and any quantity can be set for sales; Sub-Total and Total will be automatically calculated based on the Unit Purchase Price and Quantity with Discount (if any).

Item	Quantity	Unit Price	Discount (%)	Sub Total
Netbook - acer 5566	1	50000.0000	5	47500
RAM - Transend 3344L	2	4100.0000	0	8200
Harddisk - Samsung 1 Tera 4533LL	1	4800.0000	0	4800.0000
*				

Total Amount : 60500

Complete Order

Fig. 6.9: Screenshot of New Sales Order placement window.

After clicking **Complete Order**, the user will be asked to confirm order placement. The user can also view the Order list by clicking **Sales** and then **Sales Order List**.

6.3 Web Application

Similar to the desktop software, the web application is also quite straightforward to use. A typical use-case scenario followed by screenshots of features is presented in this section.

6.3.1 Typical Use-Case Scenario

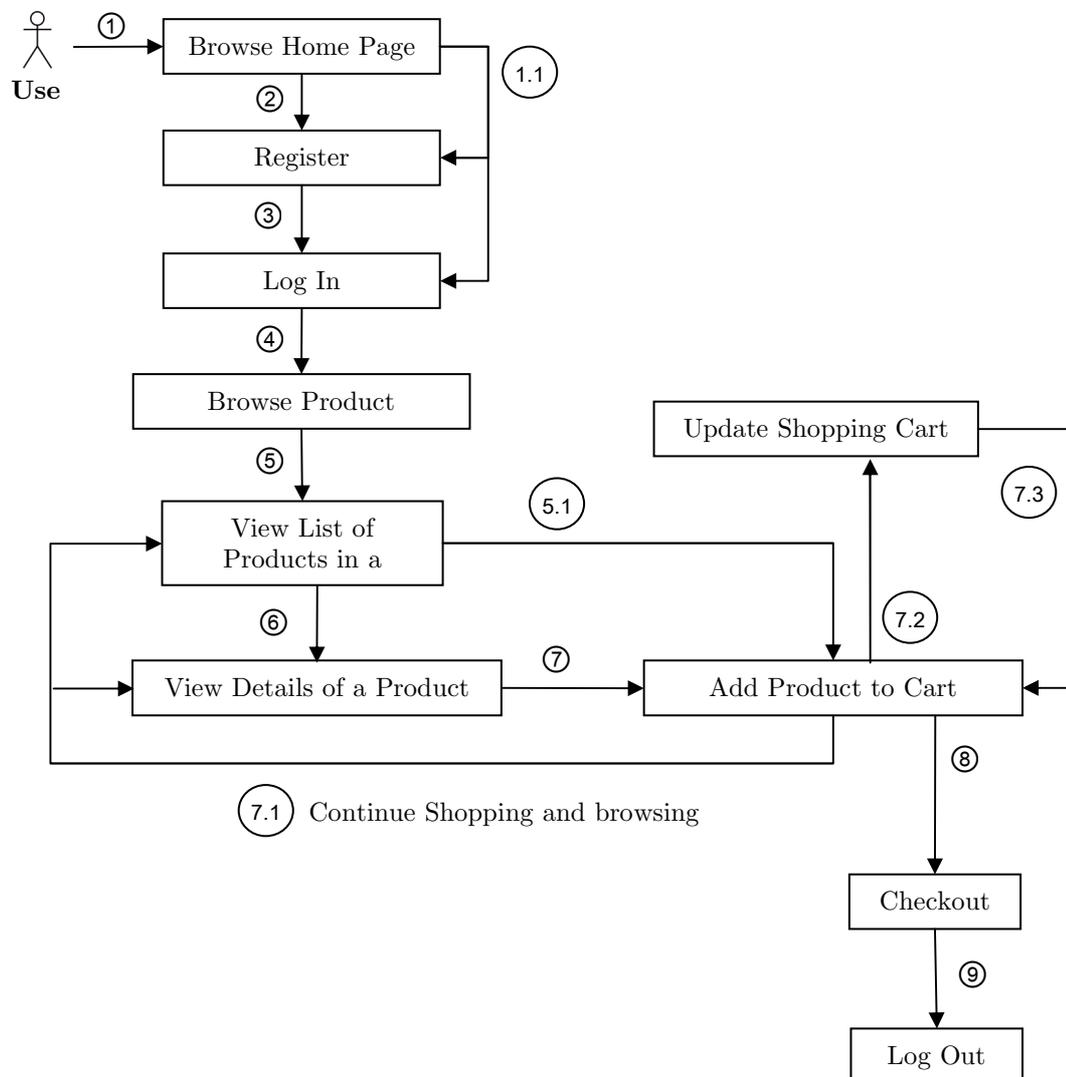


Fig. 6.10: Typical use-case scenario of the web application.

6.3.2 Home Page

The home page displays a listing of some of the best selling products. From the home page, a user can do the following:

1. Jump to the details of the products and product category listing
2. Register and log in
3. View his shopping cart and checkout the cart
4. View advertisements

Oneous IT Product Store [Browse Product Categories](#) | [Log In](#) | [Sign Up](#)

Buy Canon *REBEL*
Starting at **\$499**

Best Selling Products

- Sony VAIO 11.1" Notebook PC**
See all [Laptops](#)
- ASUS VH232H Glossy Black 23" 5ms Widescreen Full HD 1080p LCD Monitor**
See all [Monitors](#)
- SAMSUNG P2770FH 27" 1ms Full HD HDMI LCD Monitor**
See all [Monitors](#)
- Western Digital Scorpio Blue 640GB 2.5" SATA 3.0Gb/s Internal Notebook Hard Drive -Bare Drive**
See all [Hard Disk Drives](#)
- SAMSUNG Spinpoint F4 2TB 3.5" SATA 3.0Gb/s Internal Hard Drive -Bare Drive**
See all [Hard Disk Drives](#)

MY CART
There are **4 items** in your cart.
Cart grand total: \$3179.96
[Checkout](#)

BACK TO SCHOOL
Keep your eyes open for our special **Back to School** items and save **BIG!**

NEWSLETTER
Sign Up for our newsletter.

[Subscribe](#)

NOW ACCEPTING PayPal

[About Us](#) | [Customer Service](#) | [Site Map](#) | [Advanced Search](#) | [Contact Us](#) | [RSS](#)

© 2010 Oneous IT Solutions. All Rights Reserved.
Powered By: **Oneous Business Solutions**

Fig. 6.11: Screenshot of home page.

6.3.3 Registration and Login

A user must be registered with the website before he can make any purchase. Users can register with the website from the *Sign Up* link in the webpage header.

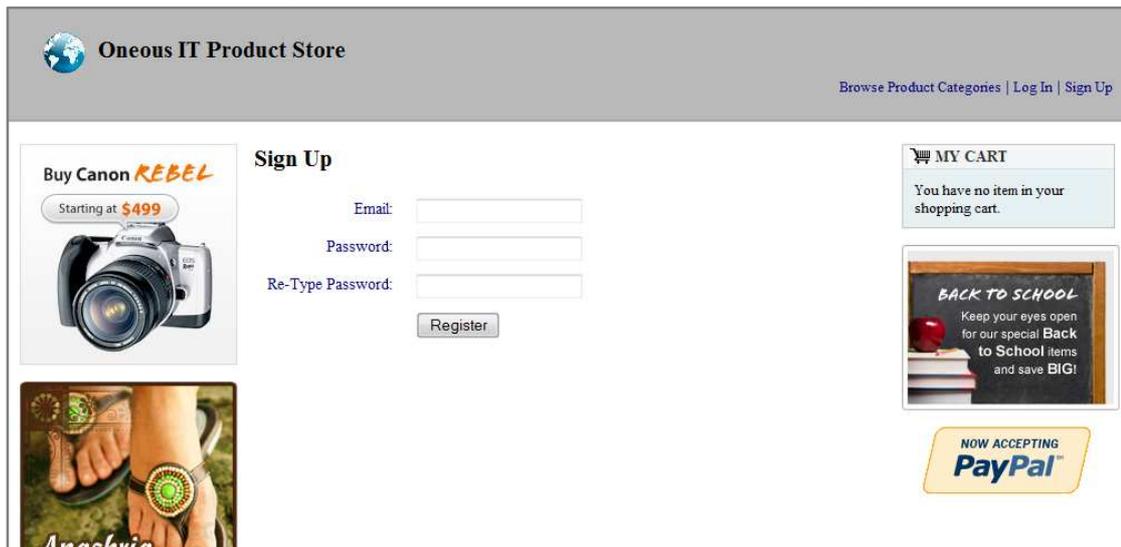


Fig. 6.12: Screenshot of *Sign Up* page.

After registration, the user will be redirected to the page where he can log in. The user can also log in from the *Log In* link in the webpage header anytime.

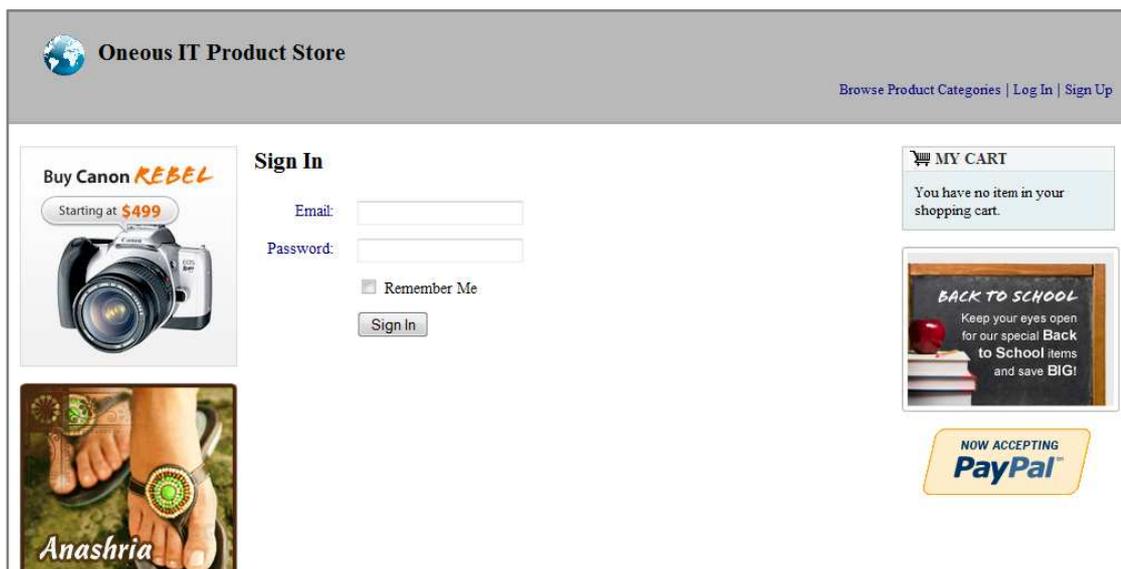


Fig. 6.13: Screenshot of *Sign In* page.

6.3.4 Browsing Product Categories

Products are organized by categories. Users can browse through the product categories by clicking the “Browse Product Categories” link at the top-right corner of the website.

The screenshot displays the Oneous IT Product Store website. At the top left, there is a globe icon and the text "Oneous IT Product Store". At the top right, there are links for "Browse Product Categories", "Log In", and "Sign Up".

The main content area is divided into three columns:

- Left Column:** A promotional banner for "Buy Canon REBEL" with a price tag "Starting at \$499" and an image of a camera. Below it is a banner for "Anashria" with the text "Get it while they last Limited Offer 20% OFF" and an image of feet wearing sandals.
- Middle Column:** A "Browse Categories" section with a bulleted list:
 - [Laptops](#)
 - [Desktops](#)
 - [Monitors](#)
 - [Hard Disk Drives](#)
- Right Column:** A "MY CART" section with the text "You have no item in your shopping cart." Below it is a "BACK TO SCHOOL" promotional banner with the text "Keep your eyes open for our special Back to School items and save BIG!" and an image of a red apple. At the bottom of the right column is a "NOW ACCEPTING PayPal" logo.

At the bottom of the page, there is a footer with the following text: "About Us | Customer Service | Site Map | Advanced Search | Contact Us | RSS", "© 2010-2011 Oneous IT Solutions. All Rights Reserved.", and "Powered By: Oneous Business Solutions".

Fig. 6.14: Screenshot of category tree.

6.3.5 Products List in Categories

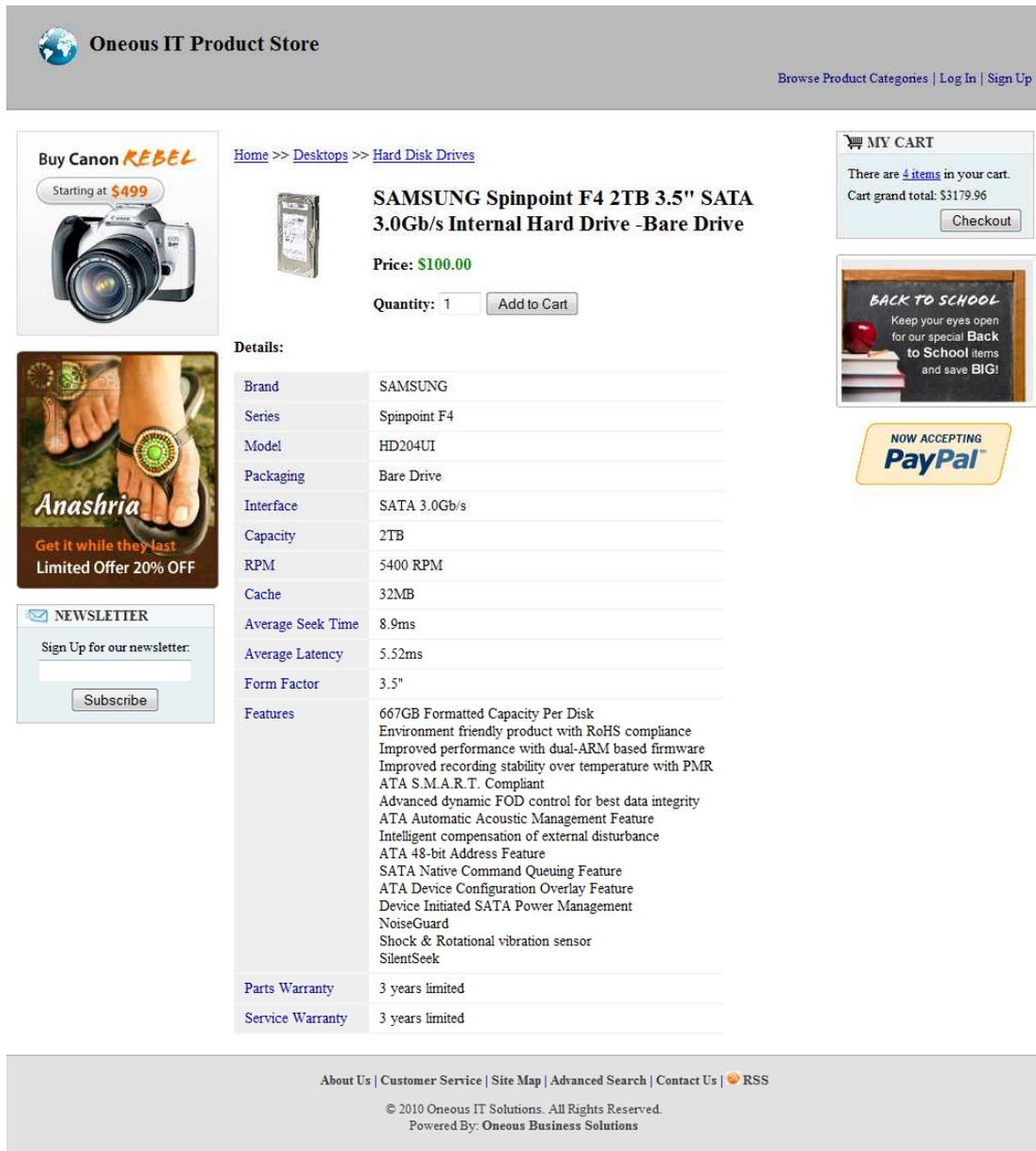
Users can click on a category name to view the list of products in that category. From the list of products, users can jump to product details as well as add products to shopping cart.

The screenshot displays the Oneous IT Product Store website. The header includes the store name and navigation links for product categories, login, and sign up. The main content area is titled 'Home >> Desktops >> Monitors'. It features a list of two monitors: an ASUS VH232H and a SAMSUNG P2770FH. Each product listing includes a thumbnail image, the product name, price, and an 'Add to Cart' button. On the left, there is a promotional banner for Canon REBEL cameras and another for Anashria jewelry. On the right, there is a 'MY CART' section showing the number of items and total price, along with a 'Checkout' button. Below the cart is a 'BACK TO SCHOOL' promotion and a 'NOW ACCEPTING PayPal' badge. At the bottom, there is a newsletter sign-up form and a footer with site navigation and copyright information.

Fig. 6.15: Screenshot of product list in category.

6.3.6 Product Details

Clicking on a product title would display the page containing product details. Users can add the product to shopping cart from this page.



The screenshot displays the product details page for a Samsung Spinpoint F4 2TB 3.5" SATA 3.0Gb/s Internal Hard Drive. The page layout includes a header with the store name and navigation links, a left sidebar with promotional banners and a newsletter sign-up, a main product area with an image, title, price, and quantity selector, a detailed specifications table, and a right sidebar with a shopping cart summary and a 'Back to School' promotion. The footer contains site navigation and copyright information.

Oneous IT Product Store [Browse Product Categories](#) | [Log In](#) | [Sign Up](#)

[Home](#) >> [Desktops](#) >> [Hard Disk Drives](#)

Buy Canon REBEL
Starting at \$499

SAMSUNG Spinpoint F4 2TB 3.5" SATA 3.0Gb/s Internal Hard Drive -Bare Drive
Price: \$100.00
Quantity: 1

MY CART
There are 4 items in your cart.
Cart grand total: \$3179.96

BACK TO SCHOOL
Keep your eyes open for our special Back to School items and save BIG!

Now Accepting PayPal

Anashria
Get it while they last
Limited Offer 20% OFF

NEWSLETTER
Sign Up for our newsletter:

Details:

Brand	SAMSUNG
Series	Spinpoint F4
Model	HD204UI
Packaging	Bare Drive
Interface	SATA 3.0Gb/s
Capacity	2TB
RPM	5400 RPM
Cache	32MB
Average Seek Time	8.9ms
Average Latency	5.52ms
Form Factor	3.5"
Features	667GB Formatted Capacity Per Disk Environment friendly product with RoHS compliance Improved performance with dual-ARM based firmware Improved recording stability over temperature with PMR ATA S.M.A.R.T. Compliant Advanced dynamic FOD control for best data integrity ATA Automatic Acoustic Management Feature Intelligent compensation of external disturbance ATA 48-bit Address Feature SATA Native Command Queuing Feature ATA Device Configuration Overlay Feature Device Initiated SATA Power Management NoiseGuard Shock & Rotational vibration sensor SilentSeek
Parts Warranty	3 years limited
Service Warranty	3 years limited

[About Us](#) | [Customer Service](#) | [Site Map](#) | [Advanced Search](#) | [Contact Us](#) | [RSS](#)

© 2010 Oneous IT Solutions. All Rights Reserved.
Powered By: Oneous Business Solutions

Fig. 6.16: Screenshot of product details.

6.3.7 Shopping Cart

The shopping cart of a user displays the items in the cart with the item image, item name, unit price, quantity, subtotal price and grand total price.

From this page, users can update the quantity of products, delete products and checkout the shopping cart.

Oneous IT Product Store [Browse Product Categories](#) | [Log In](#) | [Sign Up](#)

Shopping Cart

	Product Name	Unit Price	Quantity	Subtotal
	Sony VAIO 11.1" Notebook PC	\$2699.99	1	\$2699.99
	Western Digital Scorpio Blue 640GB 2.5" SATA 3.0Gb/s Internal Notebook Hard Drive -Bare Drive	\$79.99	2	\$159.98
	SAMSUNG P2770FH 27" 1ms Full HD HDMI LCD Monitor	\$319.99	1	\$319.99
Grand Total:				\$3179.96

[Update Shopping Cart](#) [Proceed to Checkout](#)

MY CART
There are [4 items](#) in your cart.
Cart grand total: \$3179.96
[Checkout](#)

BACK TO SCHOOL
Keep your eyes open for our special **Back to School** items and save **BIG!**

NOW ACCEPTING PayPal

NEWSLETTER
Sign Up for our newsletter.

[Subscribe](#)

[About Us](#) | [Customer Service](#) | [Site Map](#) | [Advanced Search](#) | [Contact Us](#) | [RSS](#)

© 2010 Oneous IT Solutions. All Rights Reserved.
Powered By: **Oneous Business Solutions**

Fig. 6.17: Screenshot of shopping cart.

6.3.8 Shopping Cart Checkout

When checking out, the user has to provide the shipping address and choose a payment method/gateway from this page.

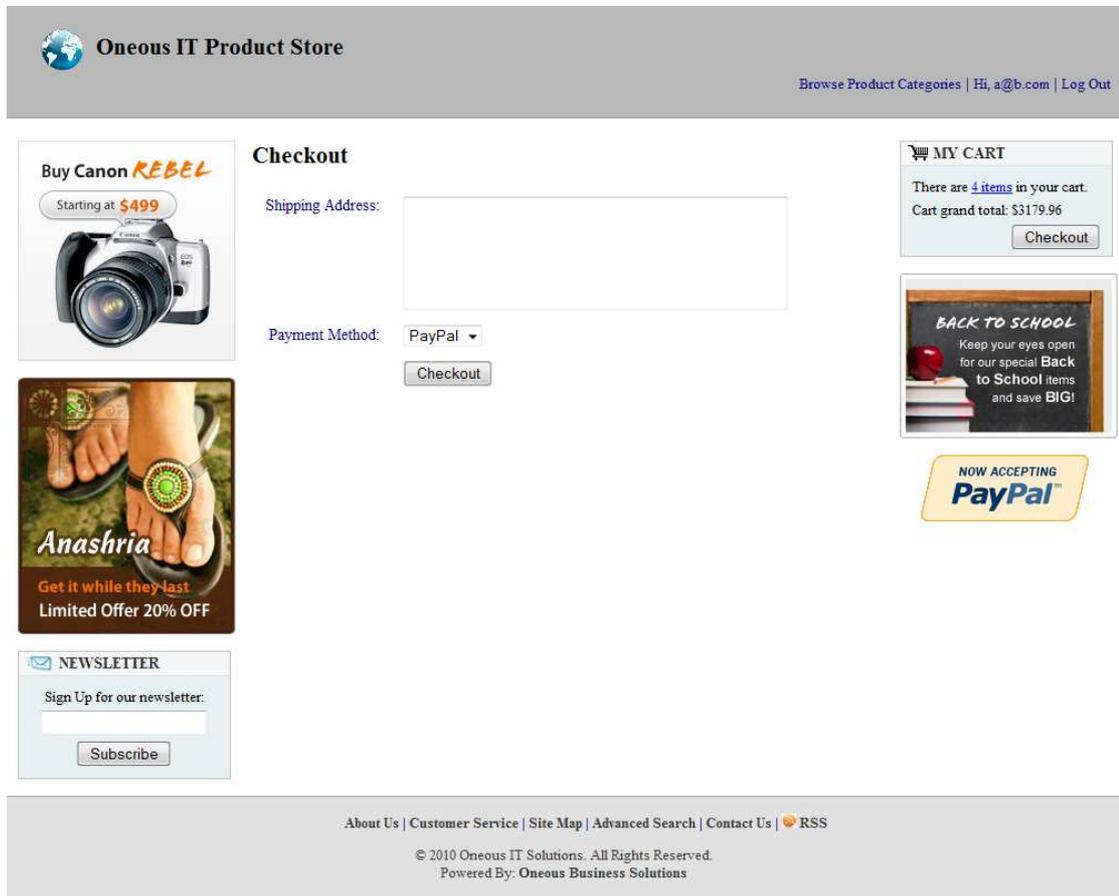


Fig. 6.18: Screenshot of shopping cart checkout.

After providing the necessary information, checking out would cause the browser to redirect to the selected payment gateway organization's checkout page. The following Fig. depicts the payment process of PayPal, the most popular payment gateway.

Oneous IT Product Store

Description	Unit Price	Quantity	Amount
Purchase at Oneous IT Product Store Item # 7	\$519.99	1	\$519.99
		Item total:	\$519.99
		Total:	\$519.99 USD

PayPal is the safer, easier way to pay



PayPal securely processes payments for **Oneous IT Product Store**. You can finish paying in a few clicks.

Why use PayPal?

- Use your credit card online without exposing your card number to merchants.
- Speed through checkout. No need to enter your card number or address.

Don't have a PayPal account?

Use your credit card or bank account (where available). [Continue](#)



LOG IN TO PAYPAL

Email:

Password:

Forgot [email](#) or [password](#)?

PayPal. The safer, easier way to pay.

For more information, read our [User Agreement](#) and [Privacy Policy](#).



Fig. 6.19: Screenshot of payment through PayPal.

6.4 Summary

Only the major functionalities of the proposed project have been shown in this chapter. The next chapter discusses the security issues handled by the proposed project.

CHAPTER 7

SECURITY CONSIDERATIONS

7.1 Introduction

Security is a vital issue for any software or website. An insecure application can cause unauthorized access to data, loss of valuable data, violation of customer privacy and lot more undesired incidents. In this chapter, various security measures taken in developing the proposed project is discussed in short.

7.2 Prevention of Unauthorized Access

Rigorous authentication checking has been incorporated in the web application. On each operation, the user is first verified. The desktop application is protected by a password to access the software.

7.3 Prevention of SQL Injection Attack

SQL injection is a code injection technique that exploits a security vulnerability occurring in the database layer of an application. The vulnerability is present when user input is either incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed. It is an instance of a more general class of vulnerabilities that can occur whenever one programming or scripting language is embedded inside another. SQL injection attacks are also known as SQL insertion attacks. [15]

7.3.1 Measures Taken to Prevent SQL Injection Attack

Using prepared statements instead of embedding parameters in SQL query strings prevents SQL injection attack. [15] Using ORM frameworks ensures

use of prepared statements. As the proposed project is implemented with the help of ORM frameworks, it is ensured to be free from this attack.

7.4 Prevention of XSS Attack

Cross-site scripting holes are web application vulnerabilities that allow attackers to bypass client-side security mechanisms normally imposed on web content by modern browsers. By finding ways of injecting malicious scripts into web pages, an attacker can gain elevated access privileges to sensitive page content, session cookies, and a variety of other information maintained by the browser on behalf of the user. Cross-site scripting attacks are therefore a special case of code injection. [16]

7.4.1 Measures Taken to Prevent XSS Attack

The primary defense mechanism to stop XSS is contextual output encoding/escaping. There are several different escaping schemes that must be used depending on where the untrusted string needs to be placed within an HTML document including HTML entity encoding, JavaScript escaping, CSS escaping, and URL (or percent) encoding. [16]

The web application part of the proposed project encodes all URLs and HTML entities; and escapes JavaScript.

7.5 Prevention of HTTP Response Splitting Attack

HTTP response splitting is a form of web application vulnerability, resulting from the failure of the application or its environment to properly sanitize input values. It can be used to perform cross-site scripting attacks, cross-user defacement, web cache poisoning, and similar exploits.

The attack consists of making the server print a carriage return (CR, ASCII 0x0D) line feed (LF, ASCII 0x0A) sequence followed by content supplied by the attacker in the header section of its response, typically by including them in input fields sent to the application. Per the HTTP standard (RFC 2616), headers are separated by one CRLF and the response's headers are separated from its body by two. Therefore, the failure to remove CRs and LFs allows the attacker to set arbitrary headers, take control of the body, or break the response into two or more separate responses — hence the name. [17]

7.5.1 Measures Taken to Prevent HTTP Response Splitting Attack

The generic solution to HTTP Response Splitting attack is to URL-encode strings before inclusion into HTTP headers such as *Location* or *Set-Cookie*. [17]

URL-encoding of strings into HTTP headers has been ensured in the web application part of the proposed project.

7.6 Prevention of CSRF/XSRF Attack

Cross-site request forgery, also known as a one-click attack or session riding and abbreviated as CSRF ("sea-surf") or XSRF, is a type of malicious exploit of a website whereby unauthorized commands are transmitted from a user that the website trusts. Unlike cross-site scripting (XSS), which exploits the trust a user has for a particular site, CSRF exploits the trust that a site has in a user's browser. [18]

7.6.1 Measures Taken to Prevent CSRF Attack

Web sites have various CSRF countermeasures available [18]:

1. Requiring authentication in GET and POST parameters, not only cookies.
2. Checking the HTTP Referer header.
3. Ensuring there's no crossdomain.xml file granting unintended access to Flash movies.
4. Limiting the lifetime of authentication cookies.
5. Requiring a secret, user-specific token in all form submissions and side-effect URLs prevents CSRF; the attacker's site can't put the right token in its submissions.

All the above measures have been taken in the web application of the proposed project.

7.7 Summary

Most applications tend to be designed without bearing security issues in mind. As the proposed project is aimed at marketing and developed considering high volume of usage, security issues have been thoroughly studied and implemented successfully in the project. Having discussed all the issues related to the proposed project, the following chapter concludes this report with some final discussions.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Comparison to Existing Similar Products

The proposed product is a desktop software for inventory and sales management bundled with a web application to handle e-commerce. There are a very few products which provide both solutions as a bundle. Also, the proposed product is more suitable for mid to small-scale business. Here are a few points comparing the proposed product to some existing reputed products offering similar solution:

1. The products with both business management and e-commerce support (e.g. BizAutomation [1], NetSuite [2] etc.) provide all their services online using cloud-computing architecture. However, in third-world countries where there are no guaranteed continuous availability of electricity and Internet, businessmen might have to suspend their sales or temporarily revert to traditional paper-based system during these periods and again re-input those data to the software if they have to use such online products.
2. These products are too costly for small-scale businesses. And small-scale businesses do not require most of the functionalities provided by these products.
3. There are other products which has desktop application for business management (e.g. Databiz Inc. Highlights [3], inFlow Inventory [4]), but they lack the e-commerce package. Also, they are not flexible enough to customize products and categories as the proposed product is.

8.2 Future Enhancements

The implemented project is primarily a prototype-implementation of the proposed project. To develop it with the aim of marketing and high volume of usage, many more features need to be added to it. Most of the features that should be added would be for enhancement of look-and-feel of the software and website, and for increasing ease of use. However, some features providing useful functionalities also need to be added. A few examples of such features may be as follows:

- Support for multiple branches/houses of a company
- Generating visual graphs and charts depicting business progress.
- Data mining to provide with useful information needed to improve business considerations.
- Full featured Point of Sales (POS)

8.3 Final Remarks

The proposed project is a great initiative for improvement of the mid to small-scale business sector especially in the third-world countries. It is hoped that successful and complete implementation and marketing of the product would bring out a new horizon of business management and e-commerce world-wide.

REFERENCES

- [1] BizAutomation, <http://www.bizautomation.com> (Last accessed on 26-Dec-2010)
- [2] NetSuite, <http://www.netsuite.com/portal/home.shtml> (Last accessed on 26-Dec-2010)
- [3] Databiz Inc. Highlights, <http://www.databizsoftware.com/Highlights-Inventory-Accounting-Software.php> (Last accessed on 26-Dec-2010)
- [4] inFlow Inventory, <http://www.inflowinventory.com> (Last accessed on 26-Dec-2010)
- [5] Wikipedia article on *C Sharp (programming language)*, [http://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](http://en.wikipedia.org/wiki/C_Sharp_(programming_language)) (Last accessed on 27-Dec-2010)
- [6] Wikipedia article on *Microsoft SQL Server*, http://en.wikipedia.org/wiki/SQL_Server_2008 (Last accessed on 27-Dec-2010)
- [7] Wikipedia article on *Object-relational mapping*, http://en.wikipedia.org/wiki/Object-relational_mapping (Last accessed on 27-Dec-2010)
- [8] Wikipedia article on *ADO.NET Entity Framework*, http://en.wikipedia.org/wiki/ADO.NET_Entity_Framework (Last accessed on 27-Dec-2010)
- [9] Wikipedia article on *XHTML*, <http://en.wikipedia.org/wiki/Xhtml> (Last accessed on 27-Dec-2010)
- [10] Wikipedia article on *Cascading Style Sheets*, <http://en.wikipedia.org/wiki/Css> (Last accessed on 27-Dec-2010)
- [11] Wikipedia article on *PHP*, <http://en.wikipedia.org/wiki/Php> (Last accessed on 27-Dec-2010)
- [12] Wikipedia article on *JavaScript*, <http://en.wikipedia.org/wiki/JavaScript> (Last accessed on 27-Dec-2010)
- [13] Wikipedia article on *MySQL*, <http://en.wikipedia.org/wiki/MySQL> (Last accessed on 27-Dec-2010)

- [14] Wikipedia article on *XML-RPC*, <http://en.wikipedia.org/wiki/XML-RPC> (Last accessed on 27-Dec-2010)
- [15] Wikipedia article on *SQL injection*, http://en.wikipedia.org/wiki/Sql_injection (Last accessed on 27-Dec-2010)
- [16] Wikipedia article on *Cross-site scripting*, http://en.wikipedia.org/wiki/Cross-site_scripting (Last accessed on 27-Dec-2010)
- [17] Wikipedia article on *HTTP response splitting*, http://en.wikipedia.org/wiki/HTTP_response_splitting (Last accessed on 27-Dec-2010)
- [18] Wikipedia article on *Cross-site request forgery*, http://en.wikipedia.org/wiki/Cross-site_request_forgery (Last accessed on 27-Dec-2010)