

E-COMMERCE WITHIN 3D ENVIRONMENT

Oleh

Khaw Kien Weng

**Disertasi ini dikemukakan kepada
UNIVERSITI SAINS MALAYSIA**

**Sebagai memenuhi sebahagian daripada syarat keperluan
untuk ijazah dengan kepujian**

SARJANA MUDA KEJURUTERAAN (KEJURUTERAAN ELEKTRONIK)

**Pusat Pengajian Kejuruteraan
Elektrik dan Elektronik
Universiti Sains Malaysia**

Mac 2005

ABSTRACT

With the advent of cheap and fast computer, the notion of 3D graphics has been very popular lately, from video games to weather simulations to movies that give us a glimpse of virtual reality. The World Wide Web has gained even more popularity. Virtual Reality Modeling Language (VRML) was born to solve this problem: how to put compelling 3D onto every PC connected to the Web. One of the effective applications of 3D on the Web is 3D E-Commerce. Most of the E-Commerce web sites currently available on the internet are presenting their products to their customers using 2D images. Due to the limitations of text and 2D graphics, the customers can just have a vague picture of the product. They could not really know how the product looks like in reality. With 3D environment or virtual reality, the customer of the E-Commerce web site may rotate a product to have a 360-degree view of that product. Rather than a 2D image, the customer can choose how to look at the model or world, the viewer can navigate the world and the most interesting is the file size can often be much less. 3D E-commerce presents the most realistic and detailed view of products to customers. This project is to investigate the feasibility of E-Commerce within 3D environment. In this project, e-commerce within 3D environment is developed. A case study of a furniture shop is considered in this project. VRML is used to develop the 3D objects and environment in the project. In order to provide text and navigation aids to help the user read about and interact with the VRML world, the web site is developed by integrating the VRML world with Hypertext Markup Language (HTML) text. Just like any other typical e-commerce web site, a server side application and database supports is a must. In this project, Java Server Pages (JSP) is used to build the server side program which is responsible to generate the web pages with the combination of static and dynamic contents and to communicate with the database. Tomcat is used as the container for running the JSP web application. MySQL, a DBMS, is used to build and manage the database for the project.

ACKNOWLEDGEMENT

I would like to express my gratitude to the many people who have made my undergraduate studies possible and who have made it such a rewarding experience. First and foremost, my greatest honour and appreciation go to Dr. Kamal Zuhairi Zamli, my supervisor cum lecturer for his tireless dedication and no-nonsense approach in guiding his students. It has been a great privilege to be a part of his group.

Many thanks to Chang Heng Tuck, Yap Chin Chuan, Lok Teng Kee, Chen Chun Ling and Chua Tiong Kin who had assisted me unconditionally.

Next, I would like to thank my family and friends who have supported me through the years. Thanks to my mother and father, who have always encouraged me to do my best.

My acknowledgments would not be complete without expressing my personal belief in and gratitude towards God, our creator and redeemer. I feel very fortunate to have had the opportunity to study this world from the vantage point of science and engineering, and am truly amazed at what a marvelous place it is.

TABLE OF CONTENTS

	Page
ABSTRACT	ii
ACKNOWLEDGEMENT	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	v
LIST OF TABLES	vi
CHAPTER 1 INTRODUCTION	1
1.1 Setting the Scene	1
1.2 Aims and Objectives of the Project	1
1.3 Overview of the Project	2
1.4 Structure of the Report	4
CHAPTER 2 LITERATURE REVIEW	5
2.1 Virtual Reality Modeling Language (VRML).....	5
2.1.1 The Advantages of VRML over Other Alternatives	5
2.1.2 Basic Concept of 3D Graphics	6
2.1.3 Building VRML Worlds.....	7
2.2 Java Server Pages (JSP).....	9
2.2.1 The Advantages of Java Server Pages over Other Alternatives	9
2.2.2 JSP Fundamentals.....	10
2.2.3 Apache Tomcat.....	12
2.3 Database - MySQL	12
2.3.1 Rationale of Using MySQL.....	13
2.3.2 Structured Query Language (SQL)	13
2.3.3 SQL Basics	14
2.4 Unified Modeling Language (UML).....	15
2.4.1 Use Case Diagram	16
2.4.2 Class Diagram	16
2.4.3 Component Diagram	16
CHAPTER 3 METHODOLOGY	17
3.1 Use Case Diagram	17
3.2 Class Diagram	20
3.3 Component Diagram	35
CHAPTER 4 IMPLEMENTATION	39
4.1 The Flow of Process - Buying Product.....	39
4.2 Administrator Login	44
4.3 The Flow of Process - Administrator Product Management	45
4.3.1 Add New Product	46
4.3.2 Edit Product	48
4.3.3 Delete Product	49
4.4 The Flow of Process – Administrator Order Management	50
4.4.1 Order Details	50
CHAPTER 5 CONCLUSION	51
5.1 Overview	51
5.2 Result.....	51

5.3	Discussion.....	52
5.3.1	Difficulties Encountered in the Project	52
5.3.2	Suggestion for Further Improvement	52
5.4	Closing Remark	53
	REFERENCES	54
	APPENDIX A: Installation of the Project.....	56
	APPENDIX B: Navigating in Cortona VRML Client	57
	APPENDIX C: Source Code	63
	APPENDIX D: CD.....	94

LIST OF FIGURES

	Page
Figure 1.1: Overview of the project	3
Figure 2.1: XYZ Axes	6
Figure 2.2: Yaw, Pitch and Roll Rotations.....	7
Figure 3.1: Use Case Diagram of the Web Site.....	17
Figure 3.2: Class Diagram of the Web Site	21
Figure 3.3: Class Diagram of VRML	27
Figure 3.4: Class Diagram of Database	32
Figure 3.5: Component Diagram for product.jsp	35
Figure 3.6: Component Diagram for cart.jsp.....	35
Figure 3.7: Component Diagram for checkout.jsp	36
Figure 3.8: Component Diagram for admin.jsp.....	36
Figure 3.9: Component Diagram for addproduct.jsp.....	36
Figure 3.10: Component Diagram for editproduct.jsp	37
Figure 3.11: Component Diagram for deleteproduct.jsp.....	37
Figure 3.12: Component Diagram for productlist.jsp	38
Figure 3.13: Component Diagram for orderlist.jsp	38
Figure 3.14: Component Diagram for orderdetail.jsp	38
Figure 4.1: Snapshot of the homepage (index.jsp)	40
Figure 4.2: Snapshots of the products	41
Figure 4.3: Anchor.....	41
Figure 4.4: Snapshot of product.jsp.....	42
Figure 4.5: Snapshot of cart.jsp	43
Figure 4.6: Snapshot of checkout.jsp	44
Figure 4.7: Snapshot of checkout.jsp (after the form is submitted)	44
Figure 4.8: Snapshot of admin.jsp.....	45
Figure 4.9: Snapshot of admin.jsp (after the administrator is authorized)	45
Figure 4.10: Snapshot of productlist.jsp.....	46
Figure 4.11: Snapshot of addproduct.jsp.....	47
Figure 4.12: Snapshot of productlist.jsp (after a new product is added).....	47
Figure 4.13: Source code added to main.wrl for new product	48
Figure 4.14: Snapshot of editproduct.jsp.....	49
Figure 4.15: Snapshot of deleteproduct.jsp	49
Figure 4.16: Snapshot of orderlist.jsp.....	50
Figure 4.17: Snapshot of orderdetail.jsp.....	50

LIST OF TABLES

	Page
Table 2-1: JSP directives and scripting elements	10
Table 2-2: Example of Database Table - Persons	14
Table 2-3: Example of result from a SELECT query	15
Table 3-1: Types of files for different border's color in class diagram	20
Table 3-2: Product class	22
Table 3-3: Description of attributes and methods in Product class	22
Table 3-4: Order class	23
Table 3-5: Description of attributes and methods in Order class	24
Table 3-6: OrderedItem class	25
Table 3-7: Description of attributes and methods in OrderedItem class	25
Table 3-8: DBConnection class	26
Table 3-9: Description of attributes and methods in DBConnection class	26
Table 3-10: Admin class	26
Table 3-11: Description of attributes and methods in Admin class	26
Table 3-12: The structure of the database table - products	33
Table 3-13: The structure of the database table - orders	33
Table 3-14: The structure of the database table - ordereditems	34

CHAPTER 1

INTRODUCTION

1.1 Setting the Scene

With the advent of cheap and fast computer, the notion of 3D graphics has been very popular lately, from video games to weather simulations to movies that give us a glimpse of virtual reality. The World Wide Web has gained even more popularity. Virtual Reality Modeling Language (VRML) was born to solve this problem: how to put compelling 3D onto every PC connected to the Web (Chris and Bruce, 1997).

One of the effective applications of 3D on the Web is 3D E-Commerce. Most of the E-Commerce web sites currently available on the internet are presenting their products to their customers using 2D images. Due to the limitations of text and 2D graphics, the customers can just have a vague picture of the product. They could not really know how the product looks like in reality.

With 3D environment or virtual reality, the customer of the E-Commerce web site may rotate a product to have a 360-degree view of that product. Rather than a 2D image, the customer can choose how to look at the model or world, the viewer can navigate the world and the most interesting is the file size can often be much less. 3D E-commerce presents the most realistic and detailed view of products to customers.

1.2 Aims and Objectives of the Project

The aim of the project is to investigate the feasibility of E-Commerce within 3D environment. In order to achieve this aim, a number of objectives have been considered:

- a) To demonstrate the visualization of 3D environment
- b) To develop the website in low cost solution.
- c) To study and implement VRML for development of 3D web site.
- d) To study implementation of database for the project.
- e) To implement the server side programming using Java Server Pages

1.3 Overview of the Project

The web site in this project displays a 3D environment of a furniture shop to the customers or visitors. The website demonstrates the visualization of 3D environment by using VRML. The web site needs to be browsed using any internet browsers which has the plug-in to support VRML.

The home page of the web site contains the 3D environment of the shop. Every product in the web page has an anchor, that will redirect the visitor to the web page which shows the details of the product, when the visitor click on the product. The customers may add the product to their shopping carts if they are interested to buy the product. The shopping cart will display the total amount for the products. The customers may go back to the home page to buy more products or check out.

The customers will be requested to fill in some personal information and credit card information to pay for the order. The application will only handle the personal information submitted by the customers while the credit card information will not be stored into the database. In real E-Commerce application, the credit card information will be transferred to payment gateway. So, some parts of this project will need to be modified before deploy it on the web.

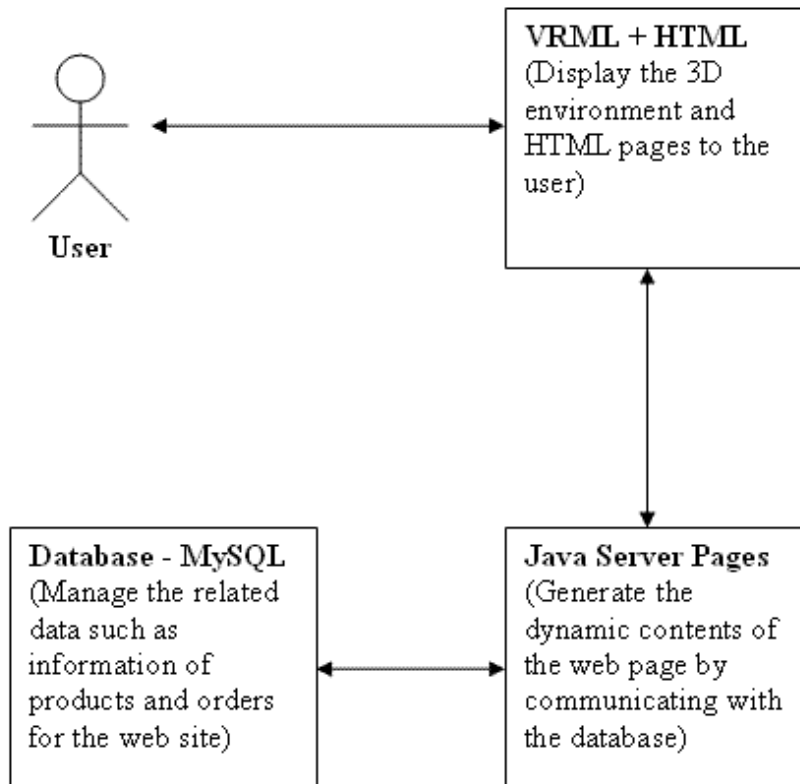


Figure 1.1: Overview of the project

For the administration of the web site, the administrator has the permission to manage products and orders. The administrator has to log in before he or she wants to carry out the tasks. After login, the administrator can view the product list and order list. The administrator may add a new product and edit or delete an existing product. The administrator has the permission to view the order list and ordered items for each order.

In this project, VRML is used to create the 3D objects and environment. JSP is used to build the server side program which is responsible to generate the web pages corresponding to the user requests while MySQL is used to build and manage the database for the project.

1.4 Structure of the Report

Chapter 1 is an introduction chapter that introduces the virtual reality on the internet and the application of 3D E-Commerce. It also lists the aims and objectives of the project to clarify the purpose of this thesis, provides a brief overview of the project which including the sequence of the flowing of the web site and the components that is required to build the web site.

Chapter 2 is a literature review chapter that covers Unified Modeling Language, Virtual Reality Modeling Language, Java Server Pages (JSP) and Database – MySQL. The chapter will gives introduction and fundamentals for each of them and their advantages over other alternatives.

Chapter 3 discusses about methodology of the project. It will describe how Unified Modeling Language is used in the design of the project. This chapter explains the design of this project by presenting use case diagram, class diagram and component diagram.

Chapter 4 provides information about the implementation of the project. It describes the flow of the process of buying products by customer and the process of products and orders management by administrator.

Chapter 5 draws the final conclusion of this work. It also discusses the difficulties faced during the project and suggests the future work as the continuation of this work.

CHAPTER 2

LITERATURE REVIEW

2.1 Virtual Reality Modeling Language (VRML)

Virtual Reality Modeling Language (VRML) is an open, industry-standard scene description language for 3D scenes, or worlds, on the internet. With VRML, one can author and view distributed interactive 3D worlds that are rich with text, images, animation, sound, music and even video.

Though VRML is a computer language, it is not a programming language. VRML files are not compiled, but are simple ASCII text files which can be parsed by a VRML interpreter. These interpreter programs (or parsers) are often called VRML browsers. The process of authoring basic VRML scenes is same like writing HTML documents: writing code with ASCII text editor, save the file, and view the VRML scene (or world) with a VRML browser.

2.1.1 The Advantages of VRML over Other Alternatives

There are several options for publishing 3D information on the internet. Here only lists the most popular ones among the available technologies. They are VRML, movies, Java3D and Morfit.

Movie is a common technology nowadays. The user is required to download the movie files and then use a movie player on the client side to play the movie. There is no flexibility of viewing positions. Besides, it requires large files and causes the problem of heavy load on networks.

Java 3D is a low level 3D scene-graph based graphics programming API for the Java language. Java 3D requires the user to download the data and the whole program for viewing the data. In addition, Java 3D uses the Java engine on the client, as well as require the developer to write the whole program for viewing the data. Typically, Java 3D may also requires large files for download.

Morfit is also a technology that can be used to build 3D web sites. Morfit 3D engine is based on Microsoft ActiveX technology. This means it is a platform-specific technology but not a standard that is available for all the platforms.

Compared to the technologies mentioned above, VRML has quite a number of advantages over them. VRML has the flexibility of viewing positions, interactive animations and sounds. VRML typically requires smaller file sizes, and thus faster to transfer in internet. The fact that VRML has become a standard makes the language not specific to any platform.

2.1.2 Basic Concept of 3D Graphics

3D graphics defines 3D space with a system of three axes:

- X axis, which is analogous to Width or left (-) / right (+)
- Y axis, which is analogous to Height or down (-) / up (+)
- Z axis, which is analogous to Depth or far (-) / close (+)

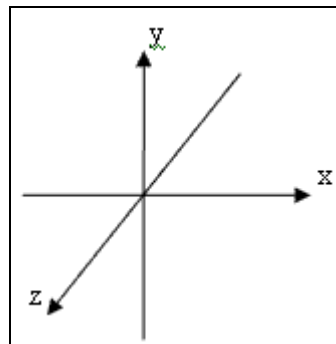


Figure 2.1: XYZ Axes

Assuming a center point, (through which the X, Y and Z axes all pass), the description of objects in 3D space can be expressed as coordinates representing absolute locations in a 3-dimensional coordinate system, relative to the center point. Each axis has a positive and a negative direction, extending from the center point of the scene. VRML assumes that, by default, the positive Y axis points up, the positive X axis points right and the positive Z axis

points toward the user. Objects within a scene (and the scene itself) may be rotated, causing the orientation of their axes to change accordingly.

Once an object's location has been determined, relative to the absolute center, it can then be oriented in one or more of three ways:

- Yaw, rotation about the Y axis
- Pitch, rotation about the X axis
- Roll, rotation about the Z axis

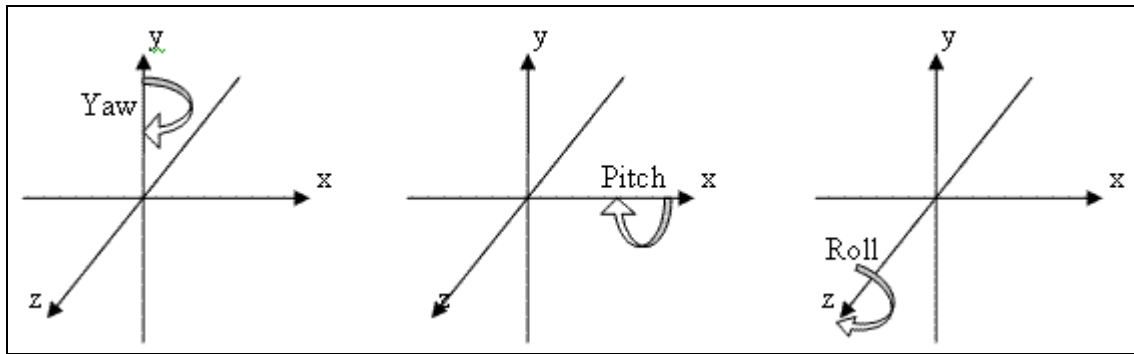


Figure 2.2: Yaw, Pitch and Roll Rotations

The three axes (X, Y and Z) and the three rotations (yaw, pitch and roll), together, is referred to as the six degrees of freedom. The location and orientation of objects in 3D space are determined by these six pieces of information.

2.1.3 Building VRML Worlds

The structure of a WRL file: VRML (*.wrl) files have 3 basic elements:

1. Header: A mandatory header line at the first line of the file which tells the browser that the file is VRML and which version it is. The pound sign (#) must be the first character on the header line:

```
#VRML V2.0 utf8
```

2. Comments: Comments are also preceded by a pound sign (#). Whenever a VRML browser sees a pound sign, it ignores the rest of that line.

3. Nodes: The objects in VRML file are called nodes. Each node contains a list of fields, which hold values that define parameters for its function. For instance, the fields of the Cone node define the height of the cone and the radius at its base. These two values are all that is needed to fully define a cone.

Here is a simple VRML file with a single node:

```
#VRML V2.0 utf8
# Draw a cone
Shape {
  geometry Cone {
    bottomRadius 0.5
    height 4
  }
}
```

Below is the list of the major node types and the features of some important nodes (Andrea et. al, 1997).

- a. Grouping nodes: Grouping nodes are used to create hierarchical transformation graphs.
E.g. Transform: Transform node have a children field that can hold a Shape node.
- b. Special groups:
E.g. Inline: A grouping node that reads its children data from a location specified in the url field.
- c. Common nodes: Used for things like light, sound, scripts, etc.
E.g. DirectionalLight: Defines a directional light source that illuminates along rays parallel to a given three-dimensional vector.
- d. Sensors: Sensor nodes generate events.
E.g. ProximitySensor: Generates events when the user enters, exits, and moves within the defined region in space.
E.g. TimeSensor: Generates events as time passes.

- e. Geometry nodes: Specifies the geometry of an object.
- f. Appearance nodes: Specifies the visual properties of geometry by defining the material and texture nodes.
- g. Interpolators: Interpolator nodes are designed for linear key framed animation.

2.2 Java Server Pages (JSP)

To build an interactive and dynamic web site, a server side scripting or web application will be needed. The server side support is responsible to generate dynamic contents of the web pages, handle the data in the forms sent by users and access database for retrieving or storing data for the web site. Java Server Pages is one of the popular server side programming technologies in web development. Java Server pages enables developer to mix regular, static HTML with dynamically generated content from servlets.

2.2.1 The Advantages of Java Server Pages over Other Alternatives

Java Server Pages has a number of advantages over many of its alternatives (Marty Hall, 2000).

Java Server Pages versus Active Server Pages (ASP):

ASP is a competing technology from Microsoft. The advantages of Java Server Pages are twofold. First, the dynamic part is written in Java, not VBScript or another ASP-specific language, so it is more powerful and better suited to complex applications that require reusable components. Second, JSP is portable to other operating systems and Web servers; the developer is not locked into Windows NT/2000 and IIS.

Java Server Pages versus PHP:

PHP is a free, open-source HTML-embedded scripting language that is somewhat similar to both ASP and JSP. The advantage of JSP is that the dynamic part is written in Java, which

already has an extensive API for networking, database access, distributed objects, whereas PHP requires learning an entirely new language. (Laura, 1996)

Java Server Pages versus Static HTML:

HTML cannot contain dynamic information, so static HTML pages cannot be based upon user input or server-side data sources. JSP is so easy and convenient that it is quite reasonable to augment HTML pages that only benefit slightly by the insertion of dynamic data. Previously, the difficulty of using dynamic data precluded its use in all but the most valuable instances.

2.2.2 JSP Fundamentals

There are five types of JSP directives and scripting elements (John Zukowski, 14 Feb 2005).

Table 2-1: JSP directives and scripting elements

JSP directives and scripting elements (and their corresponding syntax)	
Directives	<% @ directive %>
Declarations	<% ! Declaration %>
Expressions	<% = expression %>
Code Fragment of Scriptlet	<% code fragment %>
Comments	<% -- comment --%>

Directives

JSP directives are for the JSP engine. They do not directly produce any visible output but instead instruct the engine how to handle the rest of the JSP page. They are always enclosed within the <% @ ... %> tag. The two primary directives are “page” and “include”.

The page directive is the one that located at the top of almost all JSP pages. Although not required, it lets the developer to specify things like where to find supporting Java classes:

```
<% @ page import="java.util.Date" %>
```

where to send the surfer in the event of a runtime Java problem:

```
<% @ page errorPage="errorPage.jsp" %>
```


and whether information need to be manage at the session level for the user, possibly across multiple web pages:

```
<% @ page session="true" %>
```

The “include” directive is used to separate content into more manageable elements, such as those for including a common page header or footer. The page included could be fixed HTML page or more JSP content:

```
<% @ include file="filename.jsp" %>
```

Declarations

JSP declarations are used to define page-level variables to save information or define supporting methods that the rest of a JSP page may need. If too much code is included in the page, it is usually better off in a separate Java class. Declarations are found within the `<%! ... %>` tag. Always end variable declarations with a semicolon, because any content must be valid Java statements:

```
<%! int i=0; %>.
```

Expressions

With expressions in JSP, the results of evaluating the expression are converted to a string and directly included within the output page. JSP expressions belong within `<%= ... %>` tags and do not include semicolons, unless part of a quoted string:

```
<%= i %>
```

```
<%= "Hello" %>
```

Code Fragments or Scriptlets

JSP code fragments or scriptlets are embedded within `<% ... %>` tags. This Java code is then run when the request is serviced by the Web server. Around the scriptlets would be raw HTML or XML, where the code fragments can be used to create conditionally executing code, or just something that uses another piece of code. For example, the following displays the string “Hello” within H1, H2, H3 and H4 tags, combining the use of expressions and scriptlets. Scriptlets are not limited to one line of source code:

```
<% for (int i=1; i<=4; i++) { %>
```

```
<H<%=i%>>Hello</H<%=i%>>
```

```
<% } %>
```

Comments

The last of the key JSP elements is for embedding comments. Although one can always include HTML comments in the files, users can view the HTML comments if they view the page source. If the developers do not want users to be able to see the comments, they would embed them within the `<%-- ... --%>` tag:

```
<%-- comment for server side only --%>
```

2.2.3 Apache Tomcat

For the implementation of Java Server Pages, a web server such as Apache Tomcat, JavaServer Web Development Kit (JSWDK), Sun's Java Web Server and etc. In this project, Tomcat is used for the development and testing for the web application. Tomcat is an open source, Java-based Web application container that runs servlet and JSP Web applications (Tom Osborne, 14 Feb 2005). Tomcat is supported and maintained under the Apache-Jakarta subproject by volunteers from the open source Java community. Tomcat Server is the reference implementation for the servlet and JSP specifications, so it is probably safe to say that Tomcat Server implements the Apache-Jakarta specifications as well or better than most commercial servers. It can be used as a small stand-alone for testing servlets and JSP pages.

2.3 Database - MySQL

A database is an organized collection of information or data, adopted from (Jennifer Serra, 15 Feb 2005). The term database is often erroneously referred to as a synonym for a "database management system (DBMS)". They are not equivalent. A database is a store of data that describe entities and the relationships between the entities. A database management system is the software mechanism for managing that data.

Although there are different ways to organize data in a database, relational databases are one of the most effective. Relational database systems are an application of mathematical set theory to problem of effectively organizing data. In a relational database, data is collected into tables (also called relations).

A table represents some class of objects that are important to an organization. For example, a company may have a database with a table for employees, another table for customers, and another for stores. Each table is built of columns and rows. Each column (also called field) represents some attribute of the object represented by the table. For example, an Employee table would typically have columns for attributes such as first name, last name, employee ID, department, pay grade, and job title. Each row (also called record) represents an instance of the object represented by the table. For example, one row in the Employee table represents the employee who has employee ID 12345. There is another key concept for relational database, which is called primary key. A Primary Key is one or more columns whose values uniquely identify a row in a table.

2.3.1 Rationale of Using MySQL

MySQL is one of the most popular relational database management systems currently available. According to the MySQL Benchmarks, the MySQL server is presented as having the overall best performance and scalability along with Oracle9i. Also, the MySQL server is excelled in stability, ease of tuning and connectivity. In spite of its high performance, MySQL is simple to download and install. The installation procedure is straightforward and well documented, and since the information is readily available online, one should have no problems installing it. MySQL is licensed under the GNU General Public License (GPL), which means that it is free for learning and the deployment of database-enabled web site. The MySQL is fully networked, and therefore it can be accessed from anywhere on the Internet. Because MySQL is available on so many platforms, it is easy to develop an application or work with a database that can be moved to a different platform.

2.3.2 Structured Query Language (SQL)

The SQL part of MySQL stands for “Structured Query Language”, which is the most common language used to access databases. SQL is a standard interactive and programming language for getting information and updating a database. A fundamental difference between SQL and standard programming languages is that SQL is declarative while standard programming languages are procedural, they use programming elements such as conditional statements (if-then-else, do-while etc.). SQL has none of these types of statements.

SQL works with database management systems like DB2, Informix, MS SQL, Oracle, Sybase, etc. Unfortunately, there are many different versions of the SQL language, but to be in compliance with the ANSI standard, they must support the same major keywords in a similar manner (such as SELECT, UPDATE, DELETE, INSERT, WHERE, and others). (W.J. Gilmore, 16 Feb 2005)

2.3.3 SQL Basics

SQL Database Tables

A database most often contains one or more tables. Each table is identified by a name (e.g. “Customers” or “Orders”). Tables contain records (rows) with data.

Below is an example of a table called “Persons”:

Table 2-2: Example of Database Table - Persons

LastName	FirstName	City	State
Ahmad	Rahim	Bukit Mertajam	Penang
Jordan	Michael	Ipoh	Perak
Kurosaki	Ichigo	Seremban	Negeri Sembilan

The table above contains three records (one for each person) and four columns (LastName, FirstName, Address, and City).

SQL Queries

There are four basic commands which are the workhorses for SQL and figure in almost all queries to a database.

- i. INSERT INTO – inserts new data into a database table
- ii. DELETE – deletes data from a database table
- iii. SELECT – extracts data from a database table
- iv. UPDATE – updates data in a database table

With SQL, a database can be queried and have a result set returned (based on the example above).

A query like this:

“SELECT LastName FROM Persons”

Gives a result set like this:

Table 2-3: Example of result from a SELECT query

LastName
Ahmad
Jordan
Kurosaki

2.4 Unified Modeling Language (UML)

In this project, Unified Modeling Language (UML) is used to design the framework of the website. UML is a visual language that provides a way for people who analyze and design object-oriented systems to visualize, construct and document the artifacts of software systems and to model the business organizations that use such systems (Simon et. al, 2000).

In short, UML is a visual language that can be used in developing software systems. The term ‘language’ confuses some people. It is not a language like a human language, nor is it a programming language. Programming languages consist of a set of elements and a set of rules that define how to combine those elements to make valid programs. Formal specification languages like UML also consist of a set of rules that determine how to combine the elements.

Most of the elements of UML are graphical: they consist of lines, rectangles, ovals and other shapes. However, graphical elements of UML are only a graphical representation of whatever is being modeled. The graphical representation that helps people to understand the model or parts of the model, and it is the graphical representation that makes UML a visual specification language rather than a textual one. UML consists of 9 types of diagrams: use case, class, object, sequence, collaboration, component, deployment, state chart and activity. In this project, only use case, class diagram and component diagram will be presented.

2.4.1 Use Case Diagram

Use Case is a technique used for organizing and presenting requirements. Use cases are created during the early stages of a project. Producing use case diagrams and the associated documents is an analysis technique rather than a design technique. Use case diagrams show use cases and actors and the associations among them. Uses cases represent sequences of actions carried out by the system and actors represent the people or other systems that interact with the system being modeled.

2.4.2 Class Diagram

Class diagrams show the static structure of the classes that make up a system or subsystem. The static structure of classes includes the classes of interest themselves and the features of those classes – that is, their attributes and operations, and the relationships among them. Class diagrams do not show how different components of a class model interact with each other. The main purposes of producing class diagrams are to document the classes that constitute a system or subsystem and to describe the relationships among those classes.

2.4.3 Component Diagram

In a large project there will be many files that make up the system. These files will have dependencies on one another. The nature of these dependencies will depend on the language used for the development. There are also dependencies between source code files and the executable files or bytecode files that are derived from them by compilation. Component diagrams show these dependencies between software components in the system (Simon et. al, 2002).

CHAPTER 3

METHODOLOGY

UML is used to design the web application of this project. This chapter shows how the project is designed by using UML. So, use case diagram, class diagram and component diagram of the web application will be presented.

3.1 Use Case Diagram

Use cases modeling for this web site are presented as below:

Primary actor: Customer, Administrator

Secondary actor: Database

Use cases: i.) for customer – Buy Products

ii.) for administrator – Manage Products, Manage Orders

Figure 3.1 is a use case diagram for the web site.

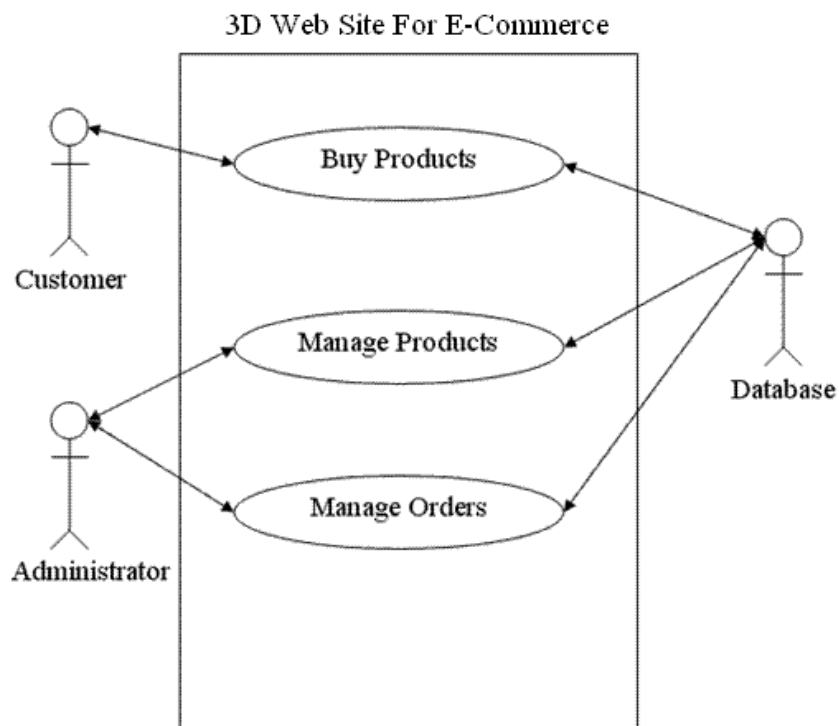


Figure 3.1: Use Case Diagram of the Web Site

Use Case for Customer

Use case name: Buy Products

Use case descriptions:

The use case starts when

1. The customer clicks a product in the 3D environment of the home page.
2. The system will extract the information of the product from the database and show the details of the product to the customer. The customer may drag the product on the page so that he or she can view the product from different angles.
3. The customer select the quantity of the product that he or she wishes to buy and then add the product into the shopping cart.
4. The page of the shopping cart will show the quantity and total for the product and the total amount for the shopping cart. The customer may go back to the homepage to buy more products.
5. After the customer is satisfied, the customer may check out.
6. In the “check out” page, the customer will be requested to enter his or her personal details such as name, address, email, phone and also the credit card information in the form.
7. After the customer clicks the button to submit the form, the page will show the order ID to the customer.
8. The information of the customer will be stored in the database. The system takes no action to the credit card information. (In the real-word application, when the customer submits the form, the credit card data will be transmitted to payment gateway).

The use case ends.

Use Cases for Administrator

Use case name: Manage Products

Use case descriptions:

The use case starts when

1. The administrator clicks the “admin” hyperlink to go to the administrator login page.
2. The administrator key in the username and password.

3. The system validates the username and password.
4. The permission is given to the administrator to view the page for products management.
5. The page will show a product list to the administrator. Here the administrator may add a new product and edit or delete an existing product.
6. If the administrator wishes to add a new product, he or she may click the “add new product” hyperlink. Then, system will show a form to the administrator to fill in the information of the product such as product name, product description, price and the URL of the VRML file for the product. After filling in the information, the administrator clicks the button to submit the information. The system will store the information of the new product into the database.
7. If the administrator wishes to edit an existing product, he or she may click the “edit” hyperlink of the corresponding product. The system will show a form with the information of the product in each field. The administrator edits the information and clicks the button to submit the information. The system will update the information in the database.
8. In the case of deleting a product, the administrator clicks the “delete” hyperlink of the corresponding product, a dialog box will pop-up to confirm the deletion of the product. The administrator clicks “yes”, and the system will delete the information of the product from the database.

The use case ends.

Use case name: Manage Orders

Use case descriptions:

The use case starts when

1. The administrator clicks the “admin” hyperlink to go to the administrator login page.
2. The administrator key in the username and password.
3. The system validates the username and password.
4. The permission is given to the administrator to view the page for orders management.
5. The page will show an order list to the administrator. Here the administrator may view the customer’s information, total amount and date for particular order.






6. The administrator may click the hyperlink for the particular order to view the details of the order.
7. After that, the system will show a list of the ordered products with the information such as product ID, product name, unit price and quantity.

The use case ends.

3.2 Class Diagram

Figure 3.2 shows the class diagram of the web site. In the class diagram, the list compartments for attributes and operations in class are omitted.

Table 3-1: Types of files for different border's color in class diagram

Border's Color	Type of Files
	Database
	JSP
	Java Bean
	VRML
	HTML

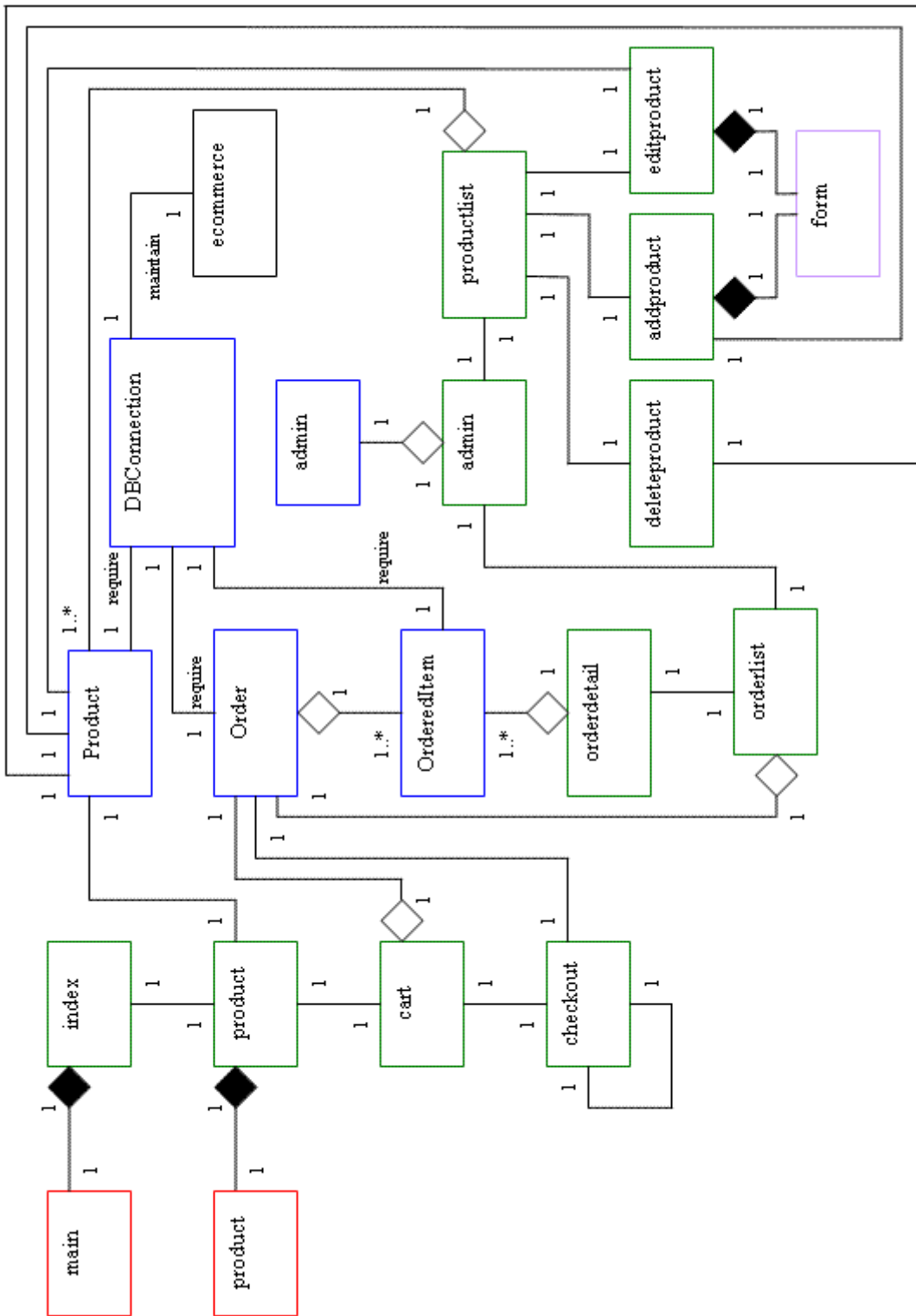


Figure 3.2: Class Diagram of the Web Site

Classes of Java Bean (.java)

Class Name: Product

Description: An entity class that holds the information for particular product and manages product information with the database.

Table 3-2: Product class

Product
+ nID : integer + txtName : String + txtDescription : String + txtVrmlUrl : String + dblPrice : double # insertSQL : String # editSQL : String
+ insert(DBConnection) : String + edit(integer, DBConnection) : Boolean + delete(integer, DBConnection) : Boolean

Table 3-3: Description of attributes and methods in Product class

Attributes	Descriptions
nID	the ID of the product
txtName	the name of the product
txtDescription	the description of the product
txtVrmlUrl	the URL of the VRML file of the product
dblPrice	the price of the product
insertSQL	the SQL statement to insert product information into database
editSQL	the SQL statement to update the product information in the database
Methods	Descriptions
insert	insert (add) the information of a new product into the database
edit	edit (update) the information of an existing product in the database
delete	delete the information of an existing product in the database

Class Name: Order

Description: An entity class that holds the information for an order and manages orders information with the database.

Table 3-4: Order class

Order
+ nID : integer + txtName : String + txtAddress : String + txtPhone : String + txtEmail : String + tsDate : Timestamp + dblTotal : double + vecOrderedItems : Vector # insertSQL : String - nMaxID : integer
+ add(integer, integer, double, String) + modify(integer, integer) + delete(integer) + calculateCart() + size() : integer + clear() + insert(DBConnection) : String - loadLastMaxID(DBConnection) - getNextOrderID() : integer

Table 3-5: Description of attributes and methods in Order class

Attributes	Descriptions
nID	the ID of the order
txtName	the name of the customer of the order
txtAddress	the address of the customer of the order
txtPhone	the phone number of the customer of the order
txtEmail	the email of the customer of the order
tsDate	the date when the customer make the order
dblTotal	the total amount for the order
vecOrderedItems	the array of object OrderedItems for the order
insertSQL	the SQL statement to insert order information into database
nMaxID	the last order ID before the order
Methods	Descriptions
add	add item into the order (shopping cart)
modify	change the number of item in the order (shopping cart)
delete	remove the item from the order (shopping cart)
calculateCart	calculate the total amount of all items in the shopping cart
size	return the size of the vector of vecOrderedItems
clear	clear the order (shopping cart)
insert	insert (add) the information of an order into the database
loadLastMaxID	get the highest nID in the table of Orders in the database
getNextOrderID	get the order ID for this coming order