INTERACTIVE CUSTOMIZING PRODUCT ORDERING SYSTEM

A thesis submitted to the Graduate School in partial fulfillment of the requirements for the degree of Masters of Science (Information Technology)

Universiti Utara Malaysia

by

Joanne Kua Swee Cheng

Sekolah Siswazah
(Graduate School)
Universiti Utara Malaysia

PERAKUAN KERJA KERTAS PROJEK
(Certification of Project Paper)

Saya, yang bertandatangan, memperakukan bahawa
(I, the undersigned, certify that)

JOANNE KUA SWEE CHENG

calon untuk Ijazah
(candidate for the degree of) Sarjana Sains(Teknologi Maklumat)

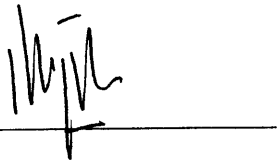
telah mengemukakan kertas projek yang bertajuk
(has presented his/her project paper of the following title)

INTERACTIVE CUSTOMIZING PRODUCT ORDERING SYSTEM

seperti yang tercatat di muka surat tajuk dan kulit kertas projek
(as it appears on the title page and front cover of project paper)

bahawa kertas projek tersebut boleh diterima dari segi bentuk serta kandungan, dan meliputi bidang ilmu dengan memuaskan.
(that the project paper acceptable in form and content, and that a satisfactory knowledge of the field is covered by the project paper).

Nama Penyelia
(Name of Supervisor) : Prof. Madya Nazib Nordin

Tandatangan
(Signature) : 

Tarikh
(Date) : 11th Oktober 2000

PERMISSION TO USE

In presenting this project in partial fulfilment of the requirements for a post graduate degree from the Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for copying of this project in any manner, in whole or in part, for scholarly purposes may be granted by my supervisor or, in their absence, by the Dean of the Graduate School. It is understood that any copying or publication or use of this project or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my project paper.

Requests for permission to copy or to make other use of material in this project in whole or in part should be addressed to:

Dean of Graduate School
Universiti Utara Malaysia
06010 UUM Sintok
Kedah Darul Aman

ABSTRACT (BAHASA MALAYSIA)

Projek ini bertujuan untuk membangunkan satu sistem tempahan interaktif yang membenarkan pengguna membuat pilihan untuk membantu industri kek tempatan. Satu teknologi imej Internet iaitu Metastream 3.0, telah digunakan untuk membangun bahagian grafik sistem tersebut. Tujuannya adalah untuk meningkatkan keyakinan pengguna terhadap imej komputer dalam menentukan kualiti sesuatu produk dan untuk membenarkan pengguna membuat pilihan. Sebagai usaha pertama di Malaysia, projek ini telah menunjukkan kebolehan pengaplikasian teknologi tersebut dalam bidang ini. Walau bagaimanapun, prestasi sistem ini dapat ditingkatkan di masa hadapan dengan meletakkan sistem ini di Internet.

ABSTRACT (ENGLISH)

This project is aimed at developing an interactive customizing product ordering system to aid the local bake goods industry. A new 3D Internet imaging technology, Metastream 3.0, was used to develop the graphical part of the system. The purpose was to increase consumer's confidence in relying on computer images to determine the quality of a product and to allow customization. Being the first attempt to use Metastream 3.0 in developing a system for bake goods in Malaysia, the project had succeeded in initiating an application in this area. Further works such as to upload the system to the Internet could be performed to improve the performance of the system.

ACKNOWLEDGEMENTS

First of all, I would like to thank my supervisor, Assoc. Prof. Nazib Nordin for all his help, encouragement and patience throughout the development of this project. I would also like to thank a friend who introduced me to Metastream technology, for this project wouldn't be as good without his advice. Last but not least I would like to thank my family members for their undying love and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Introduction

This chapter discusses the aim of the project followed by the problem statement section which will introduce the problem gradually. Several issues concerning the importance of consumer confidence in relying on computer images and customization will be covered in this section. At the end of this section, the project and its nature will then be briefly touched. This will then be followed by the objectives and significance of the project.

1.1 Problem Statement

An ordering system via the internet is an example of how E-commerce can be implemented. E-commerce is changing altogether the traditional way of doing business. Today, consumers can buy a wide variety of products and services through the Internet without having to leave their premises.

In addition to goods and services that can be delivered electronically, the Internet is also used to sell physical goods. Some popular physical goods sold on the Internet are apparel, books, food, flowers, music and toys (Margherio, et. al., 1999).

To make the most of the potential of the Internet, retailers will have to overcome a number of challenges. One of it is to increase consumer confidence in relying on computer images and information to determine the quality and fit of a product (Margherio, et. al., 1999).

Also, consumers like to think things are created and tailored especially for them. Technology has made it possible to comply with the customer's wishes and to have customization at different market prices. Even so, an interactive customizing graphical approach is seldom used. Normally electronic catalogs only include photographs with fixed pricing and some information of the product which is most often not sufficient enough to convince the customers to buy the product.

According to (Amor, 2000), many people browse the internet for information. Mostly they get textual information from it. The quality of pictures is poor compared to magazines. This is the main reason most users do not buy goods online that rely on high-quality pictures to show their design.

Goods that are bought on an emotional basis need good graphics on the internet to sell. These emotive goods are bought not only because of their price or feature, but also because the consumer like what they see. Business opportunities on the Internet for high quality imaging can be identified in two areas (Amor, 2000):

- E-Catalogues – which are high resolution images in an e-commerce storefront, especially for emotive goods.

- E-Business Communications – which are newsletters, publishing and banner advertising which will also profit from higher resolution of images.

According to (Amor, 2000), customers also want a better and more interactive experience while browsing their preferred online shop. As a conclusion to be made, something more than just high resolution images need to be of concern. Another concern of this project is to give people what they want, which is some freedom of choice for larger assortment through customization of products sold.

Analysts believe that Internet retailing will grow quickly. On the conservative end, it is expected to reach $7 billion by the year 2000. If mail order sales are used to determine the potential for Web retail sales, as some suggest, the figure could reach $115 billion in five to eight years. If online shopping provide customers with a larger assortment, better prices, and a greater choice than mail order companies, the figure may even exceed that projection (Margherio, et. al., 1999).

The advent of E-commerce and a strong economy have spawned a dramatic shift in power from sellers to customers. Today, customers are dictating not only what they will buy, but how, when and where they will buy the goods and services. Therefore the most competitive companies in the E-economy will be those that not only

understand their customer needs and expectations, but also align their entire organizations to deliver consistently differentiated and personalized customer experiences at all points of interaction (M2 Communications, 2000).

As conclusion convincing customers to buy a product is a very important factor in determining the success of a business. Therefore it would be a great advantage for any E-commerce businesses, if the imaging technology were to be enhanced at the same time giving customers the freedom to choose and customize their product to suit their personal needs.

This project is mainly interested in studying the application of Metastream 3.0 in E-commerce by designing and implementing a system that can dynamically customize a product to the wishes of the customers. The business case of this project is as follows.

Sweet Dreams, a local cake manufacturer wants to sell birthday cakes online. The important part is when the customer connects to the Internet site of the manufacturer, they are able to customize the appearance (color and some design) and price of a birthday cake. If the customer would like to order the birthday cake, they will need to fill out an electronic form stating some personal details and the time and date to pick up the cake from the store. All payments are done physically at the time the pickup is done. Other services include a calorie counter designed for the health conscious and an informative tour of how a cake is decorated.

In building an interactive customizing product ordering system for Sweet Dreams, which is a bakery ordering system, the problem statement of this project can be narrowed down to the below:

- This project will develop a system that will display different designs of birthday cakes according to a set of customer/user specifications.

- The a system shall calculate the price for the cake according to a set of customer/user specifications used in hope to aid the user in achieving an appropriate price.

- The system shall calculate the calories of the cake according to a set of customer/user specifications used in hope to aid health conscious users.

1.2 Objectives

The aim of this project is to design and implement an interactive customizing product ordering system for Sweet Dreams, which is a bakery ordering system, to increase consumer's confidence in relying on computer images and information to determine the quality of a product. The project also aims to increase customer satisfaction through customization.

In order to realize the full potential of an image-rich web site the business-to-consumer requirements, the Internet imaging technology needs to meet the following (Amor, 2000):

- High resolution – Viewing and printing of high resolution images should be feasible by as many users as possible, without having the need for more resources on the client, server and network side.

- Download speed – Download times for high-resolution images must be kept to a minimum.

- Universal format – A multi-functional, information-rich format, that allows a bandwidth efficient display on the web and printout. It should need neither special plug-ins nor additional software.

- Universal access – The users should be able to receive the pictures that they desire from any source without being restricted by their own location.

- Free choice of browser and image processing software – The users should be able to use any browser or image processing software to download, print or alter the images.

- Scalability – The Internet imaging solutions and technologies must enable companies and webmasters to manage the fast-growing numbers of images quickly and efficiently.

- Integration with existing solutions – Every new imaging technology must enable the users to include downloaded images in existing applications.

1.3 Significance of Project

The project is significant since it is the first project initiated to aid the bake goods industry in Malaysia. At the moment, no previous attempt has been carried out for this similar kind of project.

In addition, the graphical part of this project is built using Metastream 3.0 technology, which is a new technology for streaming hyper-realistic three dimensional (3D) images on the web (Metastream 3 technology was launched by Metastream, on March 29th 2000).

This project is hoped to increase consumer confidence in computer images and information thus making people more confidant in the quality of physical goods sold online. It is also hoped that this project will serve as an example to other E-commerce sites which are still using 2D graphics thus encouraging them to use Metastream 3.0 technology in the displaying and selling of their products online.

1.4 Summary

As a summary of this chapter, the project aims to aid the local bake goods industry. The project also aims to increase consumer's confidence in relying on computer images to determine the quality of a product and to increase customer satisfaction through customization. These issues are discussed in detail in the problem statement section. In the problem statement also, a description of the project, an interactive customizing product ordering system for Sweet Dreams bakery, was briefly touched. This is then followed by the objective of the system and ended with the significance of this project.

Chapter 2 Background: Internet Imaging Technologies

This chapter will cover five Internet imaging technologies suitable for the project that is, FlashPix, QuickTime VR, VRML, Cult3D and Metastream 3.0. For each of the technology an overview is covered. This is followed by the required software for each of the technology and a comparison of the technology and the business case of Sweet Dreams. This chapter will end with a comparison of the five Internet imaging technologies.

2.1 Review

Today, most web browsers support only static image format. Static means that the image itself cannot change its colors, perspective or resolution. All browsers understand the static file formats such as GIF, JPEG and PNG (Amor, 2000).

Static images are best choice for images that do not contain content, but are for web page design purpose only. However, static image formats are frequently used for delivering high-quality and interactive imaging, but this requires large overheads and limits its use. For example, clicking on a small thumb nail of an image on a web page will give a higher resolution on the image of the same picture. This may be a good solution. However, it becomes highly complicated with thousands of

images on the web site displaying them all in different resolutions. Using a static format requires some good conversion tools, a database and some other tools (CGI-scripts, content management systems, etc.) to manage the site. The use of interactive image formats will simplify the administrative procedure (Amor, 2000).

There is a whole range of different products on the market that support dynamic image format. Dynamic image formats are able to change their resolution, perspective, lights, colors and the like. In the context of Internet imaging, dynamic does not mean Java applications, Director movies or Active X controls that can also deliver dynamic and interactive formats that need additional programming. Dynamic image formats do not require programming but they do require design (Amor, 2000).

Dynamic images contain meta-information about the picture itself, allowing one to modify its appearance without having to create a new picture. As an example, an apple can viewed from all sides without having to reload the web page. However, looking at an apple in GIF format cannot allow the viewer to view the apple from different perspective. This is simply because the GIF format hasn't stored any information on what the other side of the apple looks like (Amor, 2000).

Most of the dynamic image formats use browser plug-ins or need special viewer programs. This may limit the use of the formats if the plug-ins are not available for all target platforms (Amor, 2000).

In this project five Internet imaging technologies are reviewed on the basis of the business requirements stated above. The five technologies are FlashPix, QuickTime VR, VRML, Cult3d and Metastream 3.0.

2.2 FlashPix

FlashPix is unlike any other imaging technology available today. This is because the key to FlashPix is a multi-resolution, tiled file format that allows images to be stored at different resolutions for different purposes, such as editing or printing. Each resolution is divided into 64 X 64 blocks, or tiles. Within a tile, pixels can be either uncompressed, JPEG compressed or single color compressed (Amor, 2000).

FlashPix objects are stored in structured storage container files and the image data is stored in defined color spaces called sRGB, which are close to most commonly used RGB color spaces, but which are calibrated to known colour values. By defining the colour space options and providing standard ICC colour management profiles, FlashPix delivers consistent color on standard systems as well as with color managed systems (Amor, 2000).

The FlashPix format is also rich in non-image data definitions. This metadata includes information such as content description, camera information and scan description. Finally, FlashPix has the ability to add numerous extensions (Amor, 2000).

Many companies are using FlashPix technology. Fat Face, a UK based retailer of fashion for the Generation X has re-launched its web site in November 1998 using FlashPix technology. More than 2,300 products are now available in high resolution on the Internet. Customers now have the possibility to choose, for example, a pair of trousers and are able to zoom in and see embroideries, stitching and other details (Amor, 2000).

2.2.1 FlashPix Features

Some of the highlights of the FlashPix format are the following (Amor, 2000):

- Resolution-independent co-ordinate system.
- Multiple image representations.
- Defines a standardized color space that keeps colors consistent when viewed across various displays and printers.
- Contains compression header information shared between all files at all resolution in all sub-images, reducing file size dramatically.
- Structured storage makes FlashPix easily accessible.

- Allows creation of new extensions such as an audio extension which allows you to attach audio data to an image in a single complete package.

- Scaleable and portable.

- Specifies operations to be applied to image data acquired from a source.

- Resulting image data can be cached independently of the source format.

2.2.2 Required Software

In order to view FlashPix images over the Internet, some software need to be installed on the web server. Among the server products there are "Net-graphica/IIP Server" by TrueSpectra Inc, Hewlett-Packard's "OpenPix ImageIgniter" and Live Picture's "Image Server". Newer releases of the server software allow the use of cgi-bin applications, Java applets, ActiveX components or even flat images. The most common way is to use Java applet because it works on all platforms and requires no additional software to be installed, but in some cases, this may not be the fastest way to view the pictures (Amor, 2000: 463).

2.2.3 FlashPix and Business Case

The Sweet Dreams interactive customizing online bakery do not profit much from this technology. The requirement here is for the user to view the cake and change some design or color. FlashPix can let the user zoom up to the cake for a closer view, but it doesn't allow a 3D type of experience. Also, FlashPix is unable to allow the customers to change the colours and design of the cake.

2.3 QuickTime VR

QuickTime was by Apple Computer and has become the most popular environment for multimedia CD-ROM and Internet productions. QuickTime VR is the virtual reality component that allows viewing 360 degree panoramic images. Having panoramic images means that one can either look around from a point of view or that one can walk around a certain object of display. Panoramic images allow the webmaster to use it in a showroom in which the users are standing in the middle. Depending on the quality of the image, one can zoom into details like curtains, pictures on the wall and the like. The same method of zooming in and out can be applied to objects that can be viewed from all sides, for example, a car (Amor, 2000).

One of the most prominent users of the QuickTime VR format was NASA in July 1997, when Pathfinder landed on Mars. A little rover called Sojourner drove

around the surface of Mars and took pictures in panoramic photographs. It was later convert to QuickTime VR. The QuickTime VR model enabled users to look around Mars. They could even zoom into details to enjoy the strange landscape (Amor, 2000).

2.3.1 QuickTime Elements

QuickTime consists of three distinct elements (Amor, 2000):

- The QuickTime Movie file format defining
  - means of storing digital media compositions
  - container format storing not only media assets but also the description of the media composition

- The QuickTime Media Abstraction Layer defining
  - access to software tools and applications to the media support service
  - hardware acceleration for critical portions
  - means for extending and enhancing the media services

- Rich set of built-in media services
  - comprehensive set of built-in capabilities to reduce development time

- interoperability between QuickTime application

2.3.2 Required Software

Software for developing QuickTime VR images can be obtained from Apple and third party vendors. QuickTime is not an open standard. Development and implementation are done entirely by Apple. The software comes in a form of a stand-alone player and browser plug-in, both free in standard version. On the web server side, no additional software needs to be installed. QuickTime files are just stored in the path of the web server documents in order to be retrieved by the browsers (Amor, 200).

2.3.3 QuickTime VR and Business Case

Even though QuickTime VR is more suitable to the requirements of Sweet Dreams compared to FlashPix technology, it is still not enough. Although QuickTime VR will be able to create a 3D like environment it is not able to allow the customers to change the colour and design of the cake.

2.4 VRML

The Virtual Reality Modeling Language (VRML) is the file format standard for 3D multimedia and shared virtual worlds on the Internet. It is an open specification for creating, viewing and manipulating 3D objects and spaces. VRML adds interaction, structured graphics, depth and time to the Internet. VRML has its own scripting language and interfaces to Java and JavaScript enabling the user to create 3D worlds and make it interactive. The approach to VRML is totally different to the approach of FlashPix and QuickTime VR because one cannot scan in an image and convert it to VRML format (Amor, 2000).

The VRML language describes the objects you want to see on screen. The language describes the objects you want to see on screen. The language contains contructs for displaying cubes, cylinders, spheres and many other simple and complex geometric figures. The structure is similar to the HTML language. Using these relatively simple elements one is able to build up complex scenes. Architects, who build houses, for example use this method (Amor, 2000).

VRML worlds offer advantages over "normal online shops". Activeworlds.com is a 3D virtual reality environment featuring an online shopping called @Mart. More than 100 online businesses like Amazon.com and Beyond.com have already opened shops in @Mart. Shoppers walk around the @Mart virtual mall by pointing their mouse or using the arrow keys on their keyboard to proceed. Although the

environment is 3D, the merchandise is displayed in 2D. A detailed description of the product is displayed when clicking on the graphics (Amor, 2000).

2.4.1 VRML Features

The main features of VRML are (Amor, 2000):

- Open Standard – VRML was recognized as an international standard (ISO/IEC-12772-1:1997) in December 1997 by the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).

- 3D Multimedia – Long before its standardization VRML became the de-facto standard for sharing and publishing data between CAD, animation, and 3D modeling programs.

- Shared Virtual Worlds – Being able to talk and work in a 3D shared virtual space was one of the earliest motivations of the VRML pioneers.

- On the Internet – Unlike previous 3D applications, using the Internet to share 3D objects and scenes was built into VRML from the very beginning.

2.4.2 Required Software

All CAD software is able to design VRML objects and scenes. However, there are also specialized VRML software products. As for the interactivity, Java or JavaScript is used as a programming interface. There is no software that needs to be installed on the web server, but in order to view VRML, the user needs to download a VRML plug-in (Amor, 2000).

2.4.3 VRML and Business Case

There are a few issues in VRML that is against Sweet Dream's requirements. One of it is that it requires a lot of processing time and that renders it unusable for low-end personal computers. More complex scenes require Pentium II or PowerPC G3 chips that run faster than 400MHz, supported by fast graphics cards (Amor, 2000).

Other than that, VRML is often use in building 3D scenarios and not focusing on one object. Also, VRML does not focus on the details of an image, which is important for the displaying and convincing the customers to buy a product (Gotfried, 1999).

2.5 Cult3D

Today individuals, corporations and designers can easily build and display high-quality, animated, interactive 3D objects for both the desktop and the Web using Cult3D from Cycore. Cult3D is a powerful 3D rendering technology that allows you to deliver real time interactive objects of high quality and speed to all Internet users. Its aim is to help improve shop-to-buy ratios on e-commerce sites, reduce Web site abandon rates, improve customer service and enhance interest and understanding of sales presentations (Cult3D).

Superior image quality is critical when displaying almost any object on the Internet, 2D or 3D. With support for multiple shading types, textures, and bilinear filtering in combination with advanced compression technology, Cult3D allows Internet users to view and interact with high quality objects without suffering from lengthy downloads or jerky movements. In addition, Cult3D allows complex animations, which create opportunities for adding interactivity to an object (Cult3D).

Although aimed at improving e-commerce, Cult3D also has the objective to help improve distance learning, entertainment, art and culture in the Internet and e-advertising. One of the many example usage of Cult3D is by Dressmart. Dressmart is a Swedish company selling clothes on the Internet. On the site, customer can select among a wide range of branded clothes for men. The customers can interactively try out combinations of clothes and have a look at it in 3D. This site

also allow the user to try out different colors, collars and cuffs on a shirt and then combine the shirt with a selection of ties. The choices available - shown in small squares on the web page (done with Java programming). - are applied by clicking with the left mouse button. With the Cult3D standard interface, the user can zoom in and spin around the chosen combination of shirt and tie (Cult3D).

2.5.1 Cult3D Features

Some important features of Cult3D 5.0 are (Cult3D):

- Apart from supporting 3D on the web it also supports ActiveX which allows 3D on Microsoft Office documents such as PowerPoint and Word.

- Support for Adobe Acrobat allows insertion of 3D images into PDF files.

- Motion Controller Objects: allow object movements (translate, rotate, scale) to be specified relative to different reference frames, local space, world space and camera space.

- Sounds, expressions and cursors can be inserted into the scene graph.

- There is a Java interface

- General features include:

  - Rendering colordepths: - 32/24-bit -16-bit/15-bit/8-bit (4x4 dithering).
  - Autoconverts to the colordepth the OS is using.
  - Optimized blitting for maximum performance.
  - Unlimited number of polygons in an object.
  - 3D Rendering support:

    - Bump mapping
    - Phong shading
    - Gouraud shading
    - Environment mapping
    - Transparency
    - Flatshading
    - Bilinear filtering
    - 32-bit Z-buffer
    - Stereoscopic rendering
    - All textures are perspective correct and use both sub-pixel and sub-texel accuracy.

2.5.2 Required Software

The Cult3D viewer works with Netscape Navigator and Microsoft Explorer. It consists of two components. Firstly is the Cult3D plug-in, which allows users to view Cult3D objects on the Web, in MS Office documents, and in Adobe Acrobat files. Secondly is the Cult3D Designer and Exporter, which imports 3D models, adds interactivity, compresses the file to a fraction of its original size, and exports it

to a Cult3D object file that can be embedded on a Web page or inserted into an MS Office or Acrobat document (Cult3D).

2.5.3 Cult3D and Business Case

Cult3D meets almost all of Sweet Dream's requirement of this project. Not only it enables the production of high quality 3D images on the web, it also allows enough interactivity for changing the colours of an object triggered by users.

2.6 Metastream 3.0

MetaStream 3 (MTS3) is an interactive, 3D web-graphics delivery technology. It incorporates crisp, resolution-independent, rotate-able, 3D images with animation and user-interaction functionality. It features progressive display and fast delivery of 3-D objects streamed to Web browsers with smaller file sizes, better scalability, and better image quality. The MTS3 files are incorporated into Web (HTML) pages much the same way as animated GIFs or QuickTime movies are packaged with Web pages (Angelo, 2000).

MTS3 allows one to virtually pick up MTS3 images and then rotate them for viewing from any angle. One can zoom in to move the image objects closer, zoom out to move the image objects away, or pan the objects to move them into different positions on the screen (Angelo, 2000).

Metastream focuses on E-commerce. One example usage of MTS3 is the demo at Metastream's official site. The demo, Heidi's Linen displays a bed where the customer can rotate, zoom or even shift it in any position. Other than that, the customer can change the colours of the sheets by clicking on little squares of pattern on the same page which is done using Java Script.

2.6.1 Metastream 3.0 Features

Here are some important features of Metastream 3.0 (full features are covered in the chapter 3):

- High visual fidelity.

- MTS3 provides 3D experience equivalent to 2D in terms of quality and download times.

- MTS 3 supports interactivity and animations which includes more navigation forms as well as support for keyframe and procedural animation of object scale, position, orientation and material information.

- Easy integration into existing design workflows.

2.6.2 Required Software

For viewing MTS3 images, MTS3 plug-in needs to be downloaded. As for developing MTS3 content, its can be done using Notepad or Metastream Scene Builder (a test software by Metastream) along with any other 3D software that supports Metastream file format (.mts) to create a more complex scene.

2.6.3 Metastream 3.0 and Business Case

Metastream 3.0 technology also meets Sweet Dream's requirements. Similar to Cult3D, it enables the creation of high quality 3D graphics on the web and is interactive enough to allow the changing of colours of an object triggered users.

2.7 Comparison of Imaging Technologies

Table 2.0 depicts how much each Internet imaging technology fulfills the Sweet Dream's requirement and the objectives of the system (Amor, 2000: 474; Cult3D; Metastream):

Table 2.1: Comparison of Internet Imaging Technologies

Business Requirements	FlashPix	QuickTime VR	VRML	Cult3D	Metastream 3.0
High Resolution	Yes	Yes	Yes, requires fast computer	Yes	Yes
Download Speed	Fast	Slow	Fast	Fast	Fast
Universal Format	Adapting itself to the browser	Requires plug-in	Requires plug-in	Requires plug-in	Requires plug-in
Universal Access	Yes	Yes	Yes	Yes	Yes
Free Choice of Browser	Yes	Yes, requires plug-in	Yes, requires plug-in	Netscape & Internet Explorer, requires plug-in	Netscape & Internet Explorer, requires plug-in
Scalability	Yes	Yes	Yes	Yes	Yes
Software Integration	Image processing	Multimedia	CAD	Microsoft Office documents & Adobe Acrobat files	(no information)
Open Standard	Yes	No	Yes	No	No
Open Source	Yes	No	Yes	No	No
Level of 3D Effect	Low	Medium	High	High	High
Level of User Interaction	Medium	Medium	High	High	High

As we can see Cult3D and Metastream 3 fullfills almost all the business requirement criterias. According to (Lindsay, 2000) two leading contenders in 3D technologies over the Internet are Cycore's Cult3D 5.0 and MetaCreation's Metastream 3.0. However, Metastream is chosen to explore its potential since it is the latest among the technologies.

2.8 Summary

This chapter reviewed and compared five Internet imaging technologies for their suitability in developing the project. The five technologies are, FlashPix, QuickTime VR, VRML, Cult3D and Metastream 3.0. FlashPix is more to a 2D oriented image technology while QuickTime VR is a movie file oriented image technology. Therefore, both technologies are not suitable for this system because of the lack of interactivity. VRML is a 3D oriented imaging technology but it focuses on 3D scenarios and not on a particular object. Also, VRML is not suitable for low end computers as it requires more processing time for complex scenes. From the comparison, two most suitable technologies are Cult3D and Metastream 3.0. Both technologies enable high-resolution images which are relatively small in size. Other than that, both technologies allow enough interaction needed by the project. However as a conclusion of the comparison, Metastream 3.0 was chosen over Cult3D since it is the latest among the technologies .

Chapter 3 Metastream 3.0 Technology

Chapter 3 will cover the user requirements, capabilities, features and benefits of Metastream 3.0 or MTS3 technology (Metastream; Sree Kotay, 2000). Other than that, the application of Metastream 3.0 will be touched at the end of the chapter, stating how far this technology has been practiced.

3.1 What is Metastream 3.0?

MetaStream 3.0 (MTS3), is an interactive, 3D web-graphics delivery technology. It incorporates crisp, resolution-independent, rotate-able, 3D images with animation and user-interaction functionality (Angelo, 2000).It also scales the resolution for optimal performance on any computer or connection speed.

The primary component of the technology is a browser plug-in that allows online consumers to interact with a virtual product on an e-commerce site, much as they might in a store.

Other features of MTS3 include hyper-realistic rendering, with physically correct lighting, reflections, and shadows to provide accurate product representations. It also features a comprehensive integration layer so that other web media types and

content can interact seamlessly with MTS3 content to provide significantly more informative and compelling product presentations.

MTS3 employs a combination of wavelet technology and procedural textures and materials, producing files that are dramatically smaller yet comparable in quality to 2D image formats. Smaller files equate to faster interaction, so the consumer's experience is quick and enjoyable.

3.2 System Requirements

As mentioned earlier, the primary component of this technology is a browser plug-in. The Metastream viewer plug-in is the first application based on the Metastream 3D file format (called MTS) – a new open PC file format announced jointly by Intel Corporation and Metacreations Corporation. The plug-in will then enable the creation, delivery and viewing of scalable 3D graphics.

The system requirements for Metastream 3.0 plug-in is:

- Windows 95, 98, 2000 or NT
- Netscape Navigator 4.07 or later
- Microsoft Internet Explorer 4.x or later

- 256 color display (24-bit recommended)
- 28.8Kbps modem
- 5 MB free disk space
- 32 MB RAM (64 recommended)
- Pentium 166 or faster (Pentium II)

3.3 Capabilities of Metastream 3.0

Metastream 3.0 is able to give customers on the web a hyper-realistic or true to life shopping experience. Metastream 3.0 allows five capabilities to touch, look, design, present and teach.

3.3.1 Touch

Touch allows the customer to walk down the virtual aisle of a store and pick up something that interest him or her. It allows the customers to virtually touch the product.

3.3.2 Look

Metastream 3.0 allows the creation and presentation of photo-realistic 3D models over the web. It allows the user to look at all sides of an object or product on the screen. The customer can zoom-up, rotate or shift the object for a closer look at what interest him or her.

3.3.3 Design

Metastream 3.0 also allows customizing in online shopping. With multiple object and texture map features of Metastream 3.0, customers can interchange parts, colors and patterns to see the products in various combinations. For example, Metastream can let customers change the colours of a desired vehicle.

3.3.4 Present

Metastream 3.0 can be used to present data in a way that's both meaningful and useful in an online environment. For example, financial services companies could use Metastream 3.0 to let online customers track multiple portfolios in one fully customizable 3D graph which can aid them in making better-informed decisions.

3.3.5 Teach

Customers often hesitate in buying a product for the fear of difficultly in assembling or using. The 'teach' features of Metastream 3.0 can be used to deliver

rich and interactive content. It can help shoppers be more informed about a product, facilitate online sales and reduces customer service calls.

3.4 Metastream 3.0 Features and Benefits

Here are the full features and some descriptions of the Metastream 3.0 technology as described on the Metastream official site (Metastream):

3.4.1 Real-Time Full Scene Antialiasing

Image resolution artifacts, such as jagged or blocky edges can result in a softer, clear image that is more realistic and easier on the eye. The MTS3 Player performs this process in real-time on the entire scene.

3.4.2 Progressive Accumulation Antialiasing

This is the process of performing multiple antialiasing passes while the scene is at rest to refine the image quality. Typically 2 – 32 passes will achieve a quality level high enough to meet a photo realistic standard.

3.4.3 Real-Time Projected Soft Drop Shadows

Soft drop shadows are a subtle way to add dimension to an object on the page and its presence suggests an object isn't flush against the background. Drop shadows

make objects -- geometry, text, or complex graphics -- seem to float above the background. The size and the color (darkness or lightness) of the shadow gives the viewer an idea of about how far from the background the object is floating. These attributes are calculated and adjusted in real-time by the MTS3 player to produce the same visual cues as real life objects.

3.4.4 Real-Time E-Lightmap ™ Phong Renderer

The MTS3 Phong renderer provides the content creator more freedom to achieve their creative vision, whether it's photo-realism or a cartoon look. And because lighting and reflective qualities are combined into one e-lightmap they can be calculated simultaneously. This increases the render speed of the MTS3 Phong renderer.

3.4.5 Bump Mapping

It's the application of a special texture (a bump map) to an object that instructs the MTS3 Player to add more detail to the object's surface when it renders the scene. Bump mapping is a visual illusion -- no new geometry is actually created.

3.4.6 MIP Mapping

MIP mapping removes aliasing artifacts that appear when texture-mapped objects are some distance from the camera viewpoint. The artifacts occur because the textures can contain more detail (pixels) than the display monitor is capable of displaying in that given space. MIP mapping helps alleviate this problem by filtering and removing some detail from the texture when it is a certain distance from the camera and is no longer necessary.

3.4.7 Bilinear Filtering

Sometimes when you get very close to an object in a 3D scene, the texture map doesn't have enough pixel information to determine the real color of each pixel on the screen. The result is a texture that looks chunky. The MTS3 player uses real-time bilinear filtering to result in a smoother, cleaner image in this case.

3.4.8 Perspective Correct and Sub Pixel/Texel Texture Mapping

All textures in MTS3 are extremely accurate in their placement on the object. They remain perspective correct no matter where they are in relation to the camera. This insures the same visual cues as real life objects.

3.4.9 32bit, 16bit and 8bit Color Depth Interactive Renderer

MTS3 renders high quality images in real-time with all video cards. Even with cards capable of only 256 colors MTS3 will render a high quality image. With cards capable of millions of colors and high resolution MTS3 will always display a real-time scene. The MTS renderer automatically senses the color depth of display adjusts to match it ensuring the highest quality render without user interaction.

3.4.10 8 Bit Alpha Channel Textures

An alpha channel controls (maps) the area of opacity/transparency for a MTS3 texture and up to 256 levels of opacity/transparency can be employed on a single texture. This adds to the photo-realism of the scene because MTS3 objects have varying levels of transparency like real word objects.

3.4.11 Realistic Procedural Materials

Materials built into MTS3 provide more realistic plastics, rubbers, metals, and the like, than picture textures can provide. Also they reduce download times by eliminating the need to download a bitmap texture in many cases.

3.4.12 Real-Time Photoshop Blend Mode Support for Textures

MTS3 composites textures in the same composite techniques Photoshop uses for compositing layers. This allows more complex and compelling textures to be generated and animated in real-time with a reduced file size.

3.4.13 DrawAnywhere Technology

The Metastream plug-in window is completely transparent and allows all content on the web page to show through. This means Metastream content visually blends with other web content seamlessly.

3.4.14 Bi-Directional Scripting Between the Browser and MTS3

Any media type on a web page can send a message to the MTS3 Player via JavaScript to cause a scene attribute to change. Conversely any object in the scene can send a message via JavaScript to the browser to cause a change in the web page content.

3.4.15 XML Based MTS3 Scene File Description

In order to ease the authoring learning curve, and to take advantage of an industry standard syntax and the authoring tools that support it, Metastream 3.0 uses XML as its native media description language (Sree Kotay, 2000).

The MTS3 ".mtx" file is an XML based description of the content in a MTS3 scene, how it should look and how it interacts with other content and the viewer. According to Sree Kotay (2000), Metastream XML contain all of the information relevant to content display (including camera setting and other preferences previously tied to the HTML in Metastream 2) and encapsulates them in XML.

All MTS3 geometry, textures, etc., are referenced from the ".mtx" file. This XML based scene description allows the MTS3 scene to be generated on-the-fly based on user input from a web page. Individual geometry, textures and other media types can then be loaded from a database based on the XML scene description. This provides flexibility while exponentially reducing the amount of MTS3 content that has to be produced to display every possible combination.

3.4.16 Media Hosting

MTS3 hosts media types other than 3D content. These include: Quick Time VR panoramas and object movies and IPIX panoramas. The combination of these

media types creates a more informative and compelling message while allowing existing content to be leveraged.

3.4.17 Text and Bitmap Rollovers

Text or images that materialize when the mouse moves over specified areas in the scene. When they appear they always face the viewer no matter where the camera is placed. And they are antialiased with drop shows to help them stand out from the rest of the scene. This allows more detail and information to be exposed while helping guide the user through the scene.

3.4.18 Streaming

Metastream uses a proprietary polygon reduction algorithm to stream data from the source to the MTS3 Player. The shape of the object is maintained with the least amount of polygons and delivered. More polygons are streamed in as fast as possible increasing the detail of the object. Objects always appear in their entirety and never stream in a piece at a time. This allows MTS3 content to be delivered over any speed connection with a positive user experience.

3.4.19 Scalable Multi-Resolution Playback

The MTS3 Player uses the same proprietary polygon reduction algorithm to scale objects in the scene to ensure real-time interaction. If playback performance drops below the real-time interaction level polygons are removed to speed it up. This allows MTS3 content to be viewed most machines with a positive user experience.

3.4.20 Trixels NT

Trixels NT is a custom wavelet compression codec that adds procedural noise to retain fine image details. Textures and other bitmaps are compressed to about half the size of JPEGs through MTS3.

3.4.21 Smart Cache

The smart cache retains MTS3 objects, textures, etc., that are viewed the most in a private MTS3 cache. This eliminates the need to download them each time they are viewed and speeds their delivery to the MTS3 Player.

3.4.22 Pre-Fetching

Pre-fetching allows the content author to specify the scene content download order. This allows the content author to guarantee the content necessary for immediate

scene interaction gets downloaded first. And ensure the viewer can interact with the MTS3 content immediately.

3.4.23 Back-Face Culling

Back-face culling instructs the MTS3 Player to ignore polygons that are not visible by the camera and only render polygons that are. This reduces render times and increases real-time interaction.

3.4.24 Animation

MTS3 has a keyframe animation system that animates any scene attribute including: object translation, object scaling, object rotation, object opacity, object pivot point, camera translation, camera scaling, camera rotation, camera pivot point, camera target, texture, material, alpha channel, e-lightmap properties, shadow color, shadow radius, shadow opacity.

3.4.25 Interaction

The MTS3 interaction system is a complete event handling system that has total access to the MTS3 scene graph and attributes. This means that any animation of a scene attribute can be triggered and the playback controlled by a variety of mouse, user and custom events.

3.4.26 Self Updating Component Architecture

A significant obstacle to widespread adoption of most browser plug-ins is the difficulty end-users encounter when a new version of the plug-in is released. All effort that the author made to initially distribute the plug-in is wasted when new features are added, since everyone who has the plug-in is required to download it again. Because the plug-in as a whole are large in size by nature this interrupts the user's web experience and becomes a deterrent to using and upgrading plug-ins.

Metastream has solved this problem by making MTS3 a highly componentized system that is managed by a smart auto-updating mechanism called the MTS3 Hub. In its most basic form, the MTS3 Hub is a small component that automatically downloads other components only as they are needed. This allows Metastream and approved third party developers to invent new features and make them immediately available for use, with no interruption to the user's experience. The result is that MTS3 becomes a dynamic system that seamlessly and securely expands, updates, and repairs itself in small segments.

Of equal importance is that the component nature of the MTS3 Hub makes MTS3 truly platform independent. The Hub provides low-level browser and OS communication facilities, so new components are written as MTS3 Hub Plug-ins, not plug-ins to a particular browser or application. The only new piece required for MTS3 to run on in a new host is a new version of the Hub itself.

3.5 Metastream 3.0 and Sales Benefits

As we can see from the above features and benefits, Metastream 3.0 has a number of advantages. Here are several sales benefits that for employing Metastream 3.0:

3.5.1 Increased Sales

Greenfield Online Research has shown that consumers are far more likely to purchase products that are presented in detailed 3D over those displayed as 2D photographs. This is because, there is greater information offered in 3D version, and requires less time and effort on the consumer's part to find that information (Metastream).

3.5.2 Increased Traffic

The same research has also shown that users are much more likely to visit and remain at an e-commerce site that contains high-quality 3D product presentations. In addition to consumer education that Metastream 3.0 offers, research indicates that people just simply find it fun (Metastream).

3.5.3 Reduced Product Returns

Product return rates from online purchases soar in 30% range, almost three times that of catalog purchases. Research from Greenfield Online demonstrates that consumers are 54% less likely to return a product if they have the ability to inspect it in 3D using MTS3 (Metastream).

3.6 Metastream 3.0 Application

There are still not many sites using Metastream 3.0, since it is a new technology. Currently, Nike and Eddie Bauer are using Metastream 3.0 to present their products. Other than that, the Metastream official site has demonstrations of several products on its web pages. However, further application of this technology are still unknown.

3.7 Summary

As a summary, the application of Metastream 3.0 technology is more centered around the field of E-commerce. This is due to its features in enabling the viewing and creation of high-resolution, interactive and scalable 3D images on the web. Also, Metastream 3.0 creates a true to life experience for online shoppers which will help E-commerce sites increase their sales and lessen the rate of product return.

Chapter 4 Project Methodology

Chapter 4 will cover the methodology used in the project that is rapid evolutionary prototyping. Other topics covered include the developer's system requirements, the technologies used to develop the system and the workflow to create Metastream 3.0 content. In the developer's system requirements section, both hardware and software requirements will be touched.

4.1 What is Prototyping?

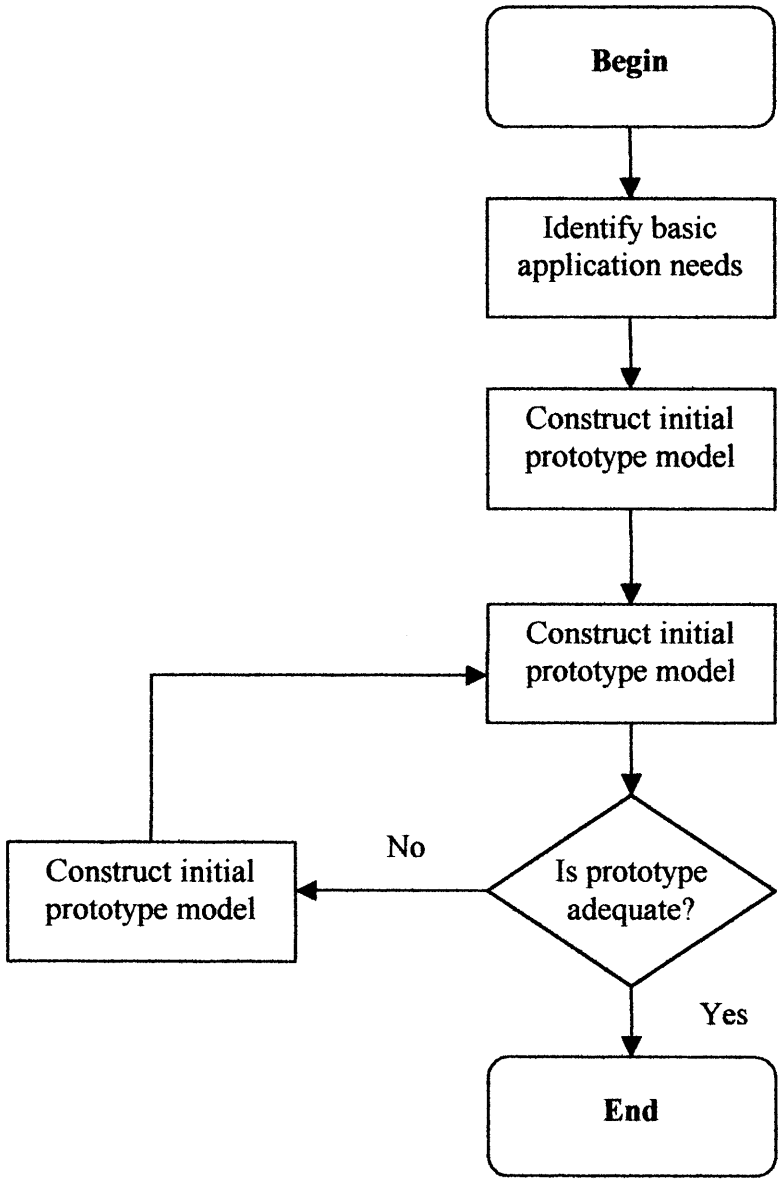
This project adapts the rapid evolutionary prototyping approach. According to Fournier (1999), the rapid evolutionary approach is particularly recommended for small sized projects, which is why the methodology was used in the project.

Prototyping consist of the construction, in context, of a working model of a system, in a short period of time. This model is generally quickly developed, but can be either thrown away once the system requirements are well understood or gradually expanded to evolve into a full-blown operational system (Fournier, 1999).

This project is more of a functional type of prototype. With functional or evolutionary prototype, a working model of the application windows, frames,

menus, reports, and forms is created (Fournier, 1999). The prototype can gradually evolve into a fully functional system. A generic view of prototyping process appears in figure 4.2 (Fournier, 1999).

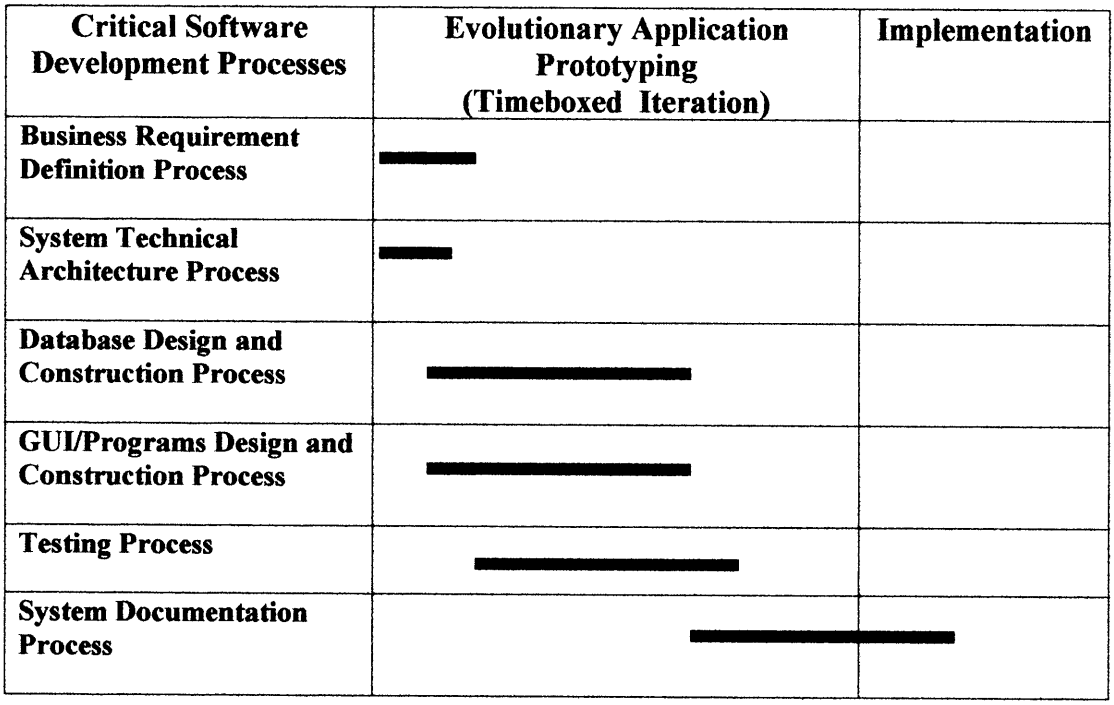
Figure 4.1: Generic View of Prototyping Process



4.2 Rapid Evolutionary Prototyping Approach

This project adapts the rapid evolutionary prototyping approach. Figure 4.1 shows the steps involved in this methodology.

Figure 4.2: Adapted Version of Rapid Evolutionary Prototyping Approach



According to Fournier (1999), there are only two development phases. The major objectives of the development phases are as in the table 4.1:

Table 4.1: Rapid Evolutionary Prototyping Approach Development Phases

Phase	Description
Evolutionary Application Prototyping	The purpose of this approach is to directly construct the system solution with an evolutionary prototyping method. The prototype gradually evolves into a fully functional system through a pre-determined number of development iterations. Each iteration is executed within a rigid timebox to control the possibility of creeping functionality and to ensure that the system is delivered in time.
Implementation	The purpose of this phase is to transfer the new system into its production environment.

Some advantages in this approach are, a more limited set of intermediate set of documents are produced. Other than that, the rapid evolutionary prototyping approach is very attractive for users since they can visually see what their system looks like as its actual construction is taking place (Fournier, 1999).

4.3 Critical Software Development Processes

Figure 4.1 listed out the most critical software development processes for rapid evolutionary prototyping. Table 4.2 is an explanation of each of the processes involved (Fournier, 1999):

Table 4.2: Critical Software Development Processes

Process	Description
Business Requirements Definition Process	This process entails the definition of the business requirements for the application. A business data and process model is usually constructed to capture data and functional needs of the application.
System Technical Architecture Process	This process entails developing an initial technical architecture for the system. The major components of the technical architecture will be included here.
Database Design and Construction Process	This process entails the design and construction of the databases that are required to support the data needs of the system.
GUI/Programs Design and Construction Process	This process entails the design and construction of all programs that are required to automate the application processes. It starts with defining the graphical user interface for the application and building a quick prototype to

	demonstrate the graphical user interface in real life mode.
Testing Process	The testing process entails verifying that the new system does meet the stated functional specifications and requirements. Evaluation is also done in this process (Mauldin, 1996)
System Documentation Process	The system documentation process entails the creation of all the user and system manuals that are required to use and support the system.

4.4 Developer's System Requirement

To develop the project both hardware and software requirements are as below.

4.4.1 Hardware Requirements

The hardware used to develop the project is as below:

- Microsoft Windows 98
- Microsoft Internet Explorer Version 5.0
- 256 color display (24-bit recommended)
- 36.6Kbps modem

- 5 GB free disk space

- 128 MB RAM

- Pentium 667 MHz or Pentium III

4.4.2 Software Requirements

The software requirements for the project as divided into the area of their usage are as below:

4.4.2.1 Graphical Software Requirements

To develop the 3D graphical representation of a cake, the software used and their functions are as below:

Table 4.3: Graphical Software Requirements

Software	Function
Metastream 3.0 plug-in	To view the Metastream 3D graphics.
Metastream Scene Builder	To compose the 3D graphics to be published in Metastream XML or .mtx and Metastream binary or .mts file format (this software is to be

	downloaded from Metastream official site).
3D Studio Max Release 2.5	For 3D graphics rendering (this software must support Metastream or .mts file extension).
Adobe Photoshop Version 5.5	For creating 2D textures for the 3D graphics.
Notepad	For the editing of Metastream XML or .mts file format.

4.4.2.2 Web Page Software Requirements

To develop the web pages and its other components, the software used and their functions are as below:

Table 4.4: Web Page Software Requirements

Software	Function
Microsoft FrontPage 2000	To develop other components of the web page. The purpose of this is to enable speedy creation editing of HTML and JavaScript pages.

4.5 Technologies Used in Web Page Components

There are four major components in this prototype. The technologies used to fulfill the requirements of each of these components are stated in table 4.5 for clarity purposes:

Table 4.5: Technologies Used in Web Page Components

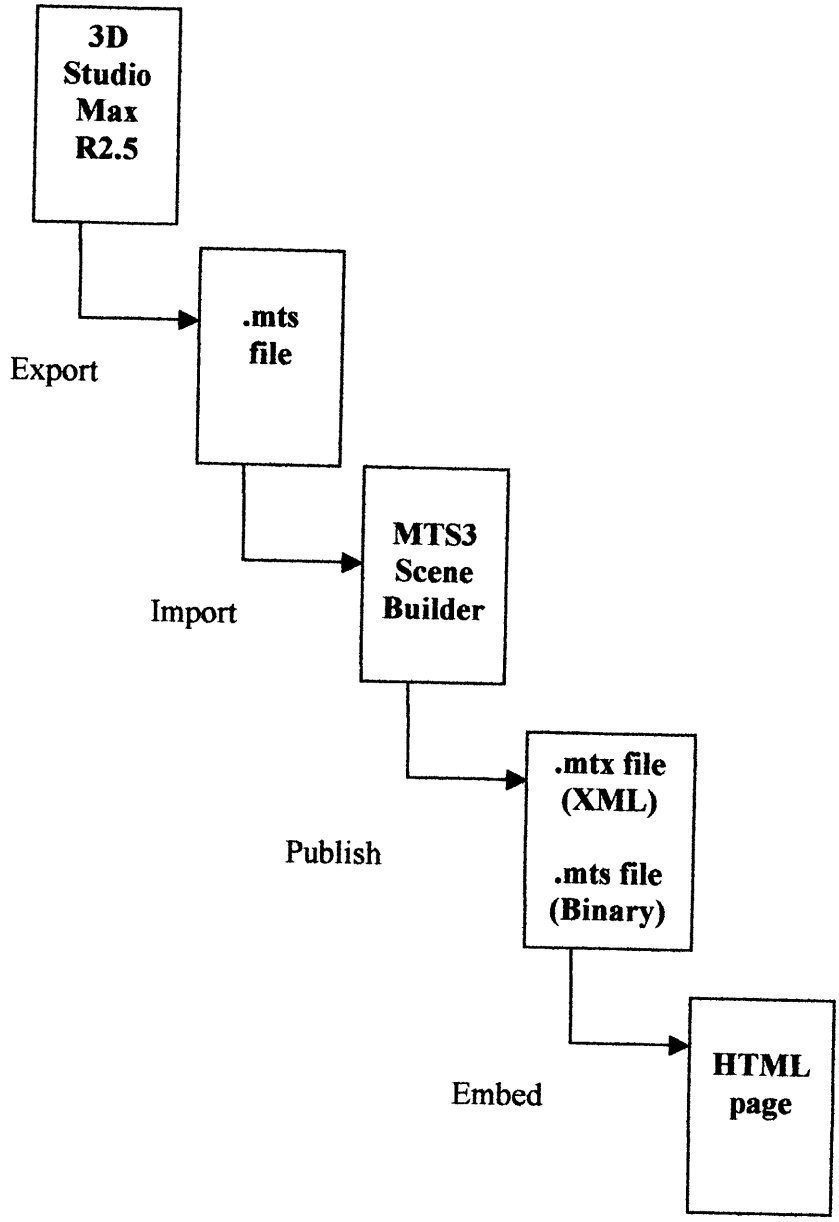
Component	Technology
3D Graphic	Metastream 3.0
Price Counter	JavaScript & HTML Language
Calorie Counter	JavaScript & HTML Language
Order Taker	JavaScript & HTML Language

The most important component to tackle in the prototype is the 3D graphic component. The general workflow for the creation of this component will later be explained in Metastream 3.0 content creation workflow. As for the other three components, knowledge of JavaScript and HTML will suffice.

4.6 Metastream 3.0 Content Creation Workflow

To develop a Metastream 3.0 content, the user require the software requirements as listed in table 4.3. The general workflow is as shown in figure 4.3.

Figure 4.3: General Workflow for Metastream 3.0 Content Creation



Firstly, 3D Studio Max Release 2.5 is used to create the 3D graphics or scene. The graphics is then exported as a .mts file. Next, Metastream Scene Builder is used to import the .mts file. Any alteration to the 3D graphics can still be made to the scene in the Metastream Scene Builder. When complete, Metastream Scene Builder is used to publish the scene into two types of files that is the .mtx and .mts files. The .mtx files is then embedded into HTML page while the .mts file is used as a reference by the system.

4.7 Summary

As a summary, the project adapts the rapid evolutionary prototyping approach. This is mostly due to the reason that the methodology is suitable for small projects. This chapter also covered the developer's system requirement in both hardware and software requirements. Apart from this, there is also a section covering the technologies used for each part of the developed system. This chapter is then ended with a general workflow of developing Metastream 3.0 content.

Chapter 5 Requirement Analysis

This chapter recaps the objectives for the interactive customizing product ordering system with the purpose of evaluating its applicability of Metastream 3.0. Later, it identifies the system actors and explains their tasks through a Use Case model. In the last section, the chapter states out and explain the crucial requirements of the system.

5.1 Objectives

The primary objective of this project is to evaluate the applicability of Metastream 3.0 technology by constructing an interactive customizing product ordering system. The system must be able fulfill the following objectives below:

- High resolution – Viewing and printing of high resolution images should be feasible by as many users as possible, without having the need for more resources on the client, server and network side.

- Download speed – Download times for high-resolution images must be kept to a minimum.

- Universal format – A multi-functional, information-rich format, that allows a bandwidth efficient display on the web and printout. If possible it should need neither special plug-ins nor additional software.

- Universal access – The users should be able to receive the pictures that they desire from any source without being restricted by their own location.

- Free choice of browser and image processing software – The users should be able to use any browser or image processing software to download, print or alter the images.

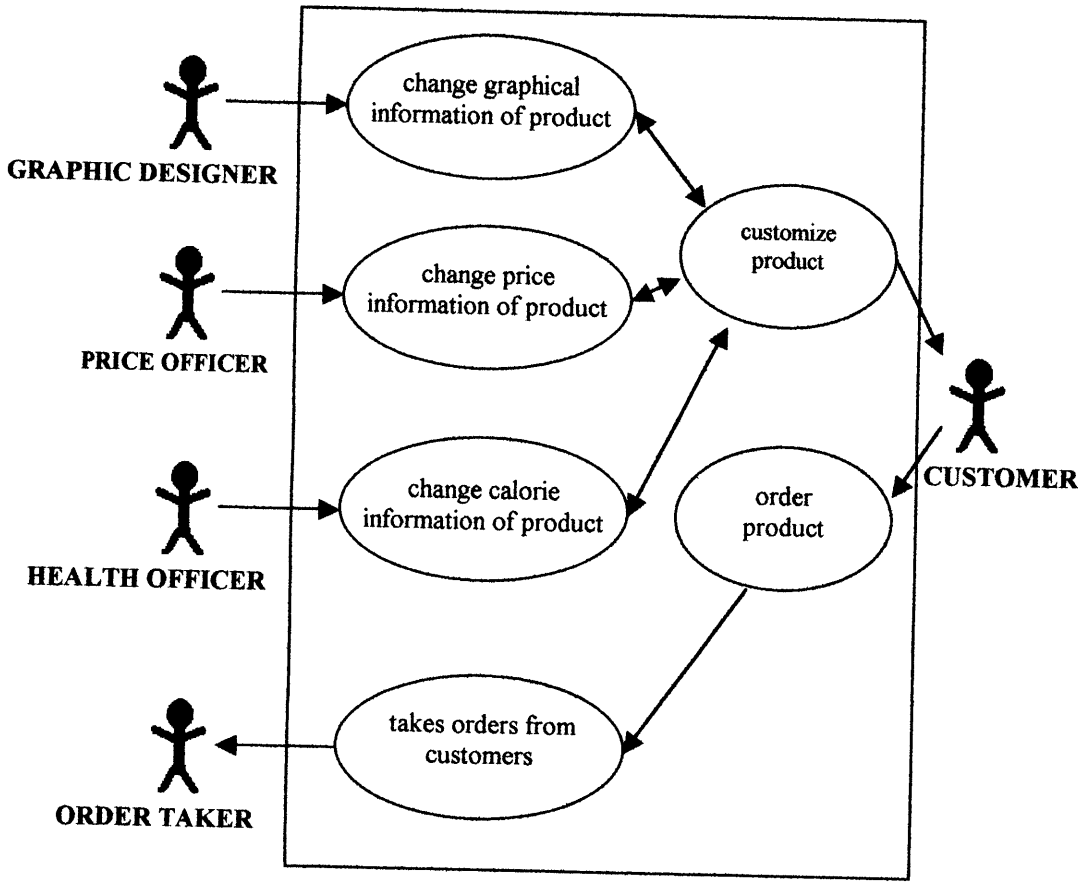
- Scalability – The Internet imaging solutions and technologies must enable the managing of fast-growing numbers of images quickly and efficiently.

- Integration with existing solutions – Every new imaging technology must enable the users to include downloaded images in existing applications.

5.2 Use Case Diagram

Uses Cases are used to document the requirements of the system. The actors of the system and their roles are shown in figure 5.1.

Figure 5.1: Use Case Diagram for the Design of Sweet Dreams System



5.2.1 Definition of Actors

The following are the actors for the system:

- Graphic Designer – Person responsible of maintaining the graphical information of the products in the system.

- Price Officer – Person responsible of maintaining the price information products in the system.

- Health Officer – Person responsible of maintaining the calorie information products in the system.

- Order Taker – Person responsible of taking the orders from customers.

- Customer – Person using the system with the intention to buy the product.

5.2.2 Definition of Use Cases

5.2.2.1 Graphic Designer

The main task of the graphic designer in the system is to maintain the graphical information of the products in the system. Tasks include:

- Adding or removing products to the system.

- Adding or removing design information of the products.

5.2.2.2 Price Officer

The main task of the price officer in the system is to maintain the price information of the products in the system. Task include:

- Adding or removing price information of the products.

5.2.2.3 Health Officer

The main task of the health officer in the system is to maintain the calorie information of the products in the system. Task include:

- Adding or removing calorie information of the products.

5.2.2.4 Order Taker

The main task of the order taker in the system is to take orders from customer. Tasks include:

- Taking orders from customers via e-mail.

- Sending customer order confirmation back to customers.

5.2.2.5 Customer

Customers browse the web page with the intention to buy products or just for fun. Customers that have the intention to buy products will do the following:

- Customize the design of the product.
- Customize the calorie of the product
- Customize the price of the product.
- Order the product.

5.3 Interactive Customizing Product Ordering System Requirements

Based on the objectives and the definition of the Use Cases, the following are the requirements for this system. The requirements for this system are organized according to different aspects of the system (Dong Pan, 1998) that is, system performance and functionality.

5.3.1 System Performance

The requirements for this system organized according to the system performance is as below:

- Requirement 1 – The system must support high-resolution images.

  The system must enable the viewing of high-resolution images, without having the need for more resources on the client, server and network side.

- Requirement 2 – The high-resolution images shall have minimal download times.

  The download times for the high-resolution images in the system must be kept to a minimum.

- Requirement 3 – The high-resolution images must be in universal format.

  The high-resolution images in the system must allow bandwidth efficient display on the web, if possible, without additional plug-in or software.

- Requirement 4 – The system must have universal access.

  The system and its images must be able to be received by users from any source without being restricted by their own location.

- Requirement 5 – The system shall be able to run on most browsers and image processing softwares.

  The system should be flexible enough to run on most browsers and image processing software.

- Requirement 6 – The images in the system must be scalable.

  The system must enable the managing of growing number of images quickly and efficiently.

5.3.2 Functionality

The requirements for this system organized according to functionality is as below:

- Requirement 7 – The system shall enable the adding and removing of product images.

  The system should be able to allow the user to add or remove product images when change is necessary.

- Requirement 8 – The system shall enable the adding and removing of design information of products.

  The system should be able to allow the user to add or remove product design information when change is necessary.

- Requirement 9 – The system shall enable the adding and removing of price information of products.

  The system should be able to allow the user to add or remove product price information when change is necessary.

- Requirement 10 – The system shall enable the adding and removing of calorie information of products.

  The system should be able to allow the user to add or remove product calorie information when change is necessary.

- Requirement 11 – The system shall enable order taking from customers.

  The system should be able to allow the user to receive orders from customers via e-mail.

- Requirement 12 – The system shall enable customizing of product design.

  The system should enable the user to customize the product design.

- Requirement 13 – The system shall enable customizing of product price.

  The system should enable the user to customize the product price.

- Requirement 14 – The system shall enable customizing of product calorie.

  The system should enable the user to customize the product calorie.

- Requirement 15 – The system shall enable product ordering.

  The system should enable the user to enter order particulars to place an order.

5.4 Summary

This chapter discusses the objectives and requirements of Sweet Dreams interactive customizing product ordering system. The complete list of requirements is summarized in table 5.1.

Table 5.1: The Requirements for Sweet Dreams Interactive Customizing Product Ordering System

Requirement 1	The system must support high-resolution images
Requirement 2	The high-resolution images shall have minimal download times
Requirement 3	The high-resolution images must be in universal format
Requirement 4	The system must have universal access
Requirement 5	The system shall be able to run on most browsers and image processing softwares
Requirement 6	The images in the system must be scalable
Requirement 7	The system shall enable the adding and removing of product images

Requirement 8	The system shall enable the adding and removing of design information of products
Requirement 9	The system shall enable the adding and removing of price information of products
Requirement 10	The system shall enable the adding and removing of calorie information of products
Requirement 11	The system shall enable order taking from customers
Requirement 12	The system shall enable customizing of product design
Requirement 13	The system shall enable customizing of product calorie
Requirement 14	The system shall enable customizing of product price
Requirement 15	The system shall enable product ordering

Chapter 6 Design and Implementation

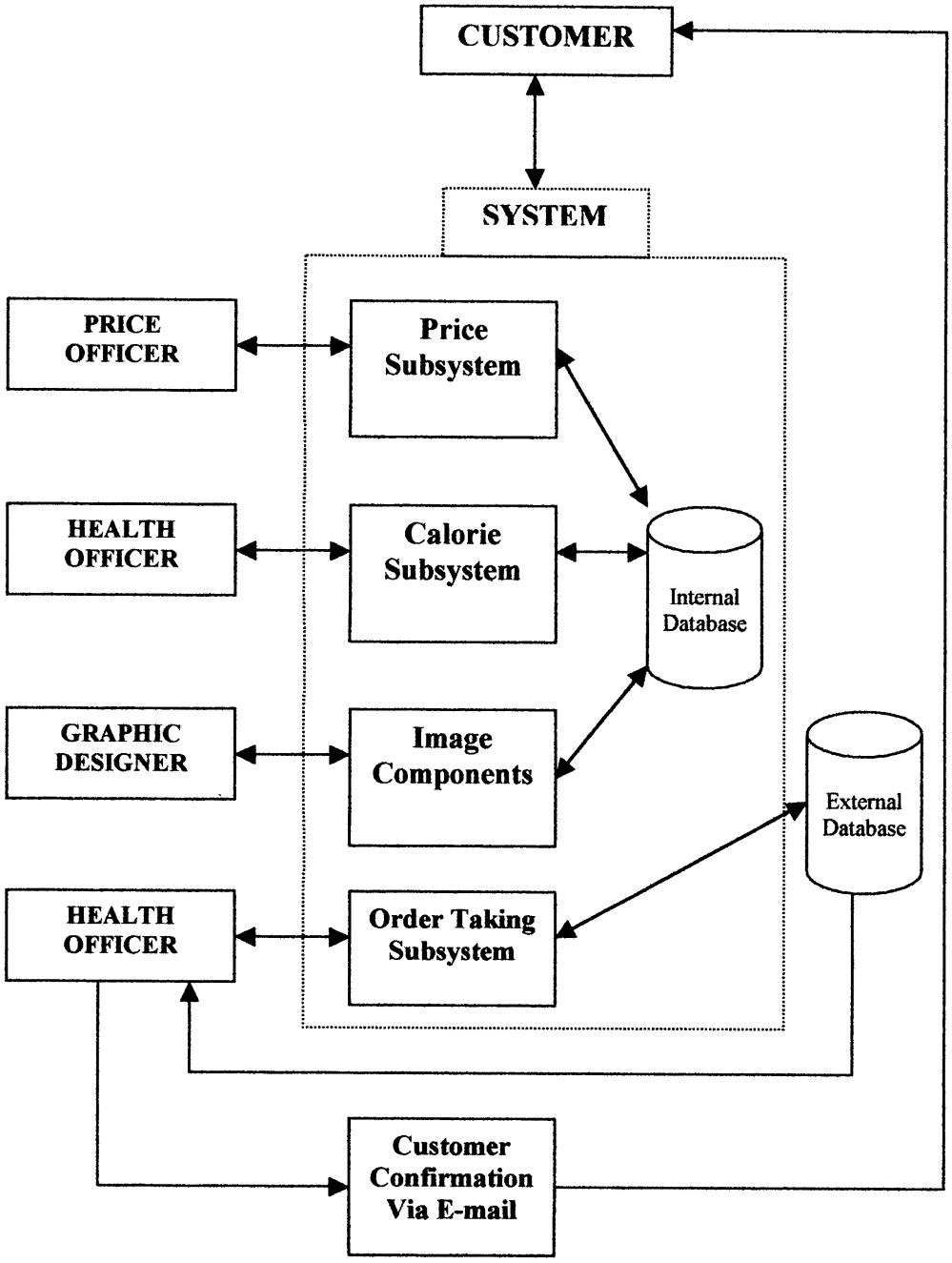
This chapter will cover the design and implementation of the prototype system. This chapter will touch on the system architecture followed by the designing of the graphical user interface of the prototype system. Finally, this chapter will end with the results from the testing of this system.

6.1 System Architecture

Figure 6.1 shows the interactive customizing product ordering system architecture. There are five actors which uses the system namely, price officer, health officer, graphic designer, order taker and customer (details have been covered in chapter 5).

The system architecture consists of four main parts and two databases. The four main parts include the price subsystem, the calorie subsystem, the graphical subsystem and the order taking subsystem. The two databases include an internal and an external database. The internal database is used to store of price, calorie and graphical information, while the external database is used to store information of customer orders.

Figure 6.1: Conceptual System Architecture



The price subsystem functions to calculate the price of the product when the user enters a set of price information while the calorie subsystem functions to calculate the calorie of the product when the user enters a set of calorie information.

The function of the graphical subsystem is to display different design combinations of a product when a set of image information is entered by the user. The order taking subsystem functions to take orders when the user enters a set of customer order information.

From figure 6.1 we can see that the price, calorie and graphical subsystem will interact with the internal database when storing or retrieving information. The order taking subsystem however, will interact with the external database when storing or retrieving its information.

Also, from figure 6.1, we can see how the actors interact with the system and its architecture. The customer interacts with the system and its four main parts. The customer may use the price, calorie, graphical or order taking subsystem. When the customer uses the price, calorie and graphical subsystem, all information and processing are done by the internal database. If an order is placed, the information is sent to the external database.

The price officer only interacts with the price subsystem. The price officer may want to add or remove price information of a product. Changes made to the price subsystem will be updated in the internal database.

The health officer only interacts with the calorie subsystem. The health officer may want to add or remove calorie information of a product. Therefore changes made to the calorie subsystem will be updated in the internal database.

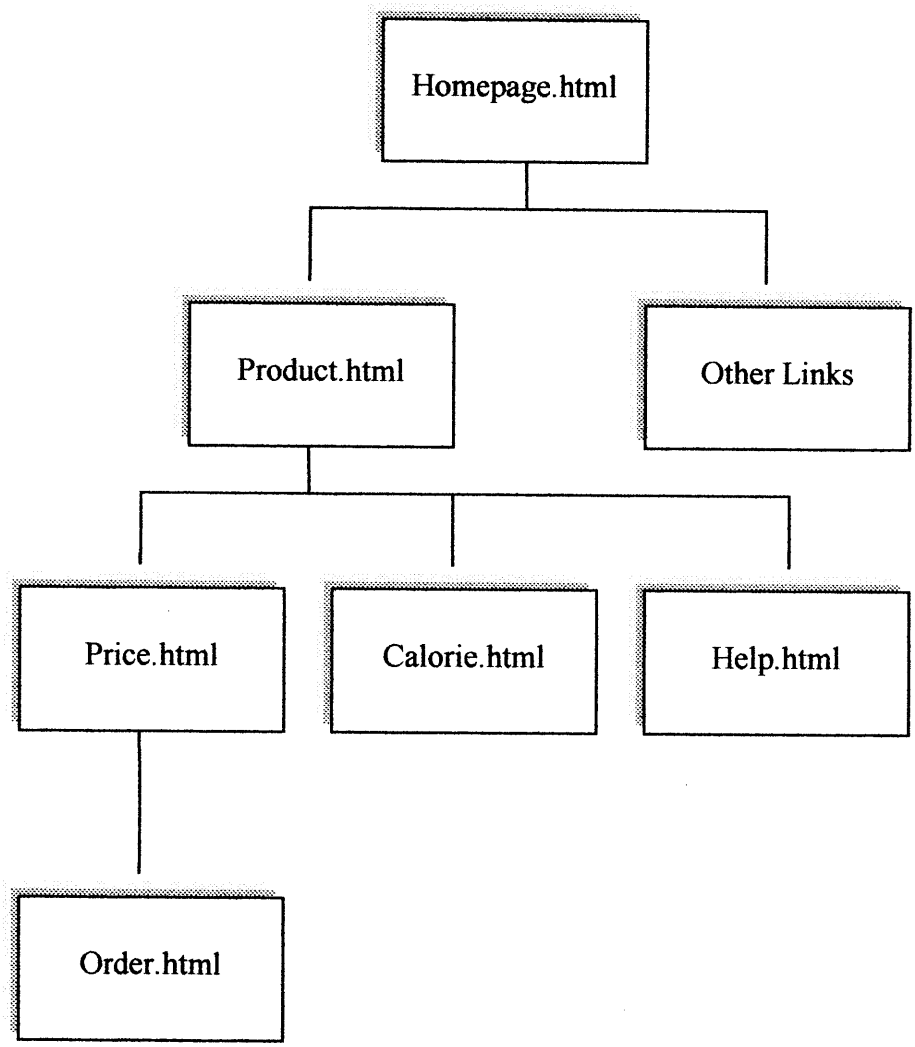
The graphic designer only interacts with the graphical subsystem. The graphic designer may want to add or remove design information of a product. Therefore changes made to the graphical subsystem will be updated in the internal database.

The order taker only interacts with the order taking subsystem. The order taker may want to take orders from customers. Customer orders are kept in the external database are then retrieved by the order taker. From here, the order taker will send the customer an order confirmation via e-mail for notification purposes.

6.2 Graphical User Interface Design

The graphical user interfaces is constructed into a working model and tested according to the design layout of figure 6.2. Figure 6.2 gives an overall illustration of the model's graphical user interface.

Figure 6.2: Graphical User Interface Design



Homepage.html is the main page of the system. The main page will introduce the company and provide links. Figure 6.2 illustrates two links from Homepage.html. Other Links are links to other information on the page, while Product.html is the link to the product page.

Product.html is the page where the product is displayed. This will is link to Price.html, Calorie.html and Help.html which are the price counter, calorie counter and help for users respectively. The Price.html page will link to Order.html. The Order.html is the ordering form for customers to fill out particulars of their order.

6.3 Testing and Results

After the system has been developed, it is tested based on the list in table 5.1 that is the requirements for the system. The aim is to see the level of functionality and operability of the prototype system. The testing and its results can be seen as in table 6.1.

Table 6.1: System Performance

Test	Description	Performance
Requirement 1	The system must support high-resolution images	The system supports high-resolution images through Metastream 3.0 technology.
Requirement 2	The high-resolution images shall have minimal download times	The high-resolution images have small download times. This is because the files are relatively small in size.
Requirement 3	The high-resolution images must be in universal format	A plug-in is required to view the images.
Requirement 4	The system must have universal access	The system is yet to be uploaded to the Internet.

Requirement 5	The system shall be able to run on most browsers and image processing softwares	The system runs perfectly on Microsoft Internet Explorer browser.
Requirement 6	The images in the system must be scalable	The images are manageable.
Requirement 7	The system shall enable the adding and removing of product images	Images are added and removed through HTML.
Requirement 8	The system shall enable the adding and removing of design information of products	Design information are added and removed through HTML, JavaScript and Metastream XML.
Requirement 9	The system shall enable the adding and removing of price information of products	Price information are added and removed through HTML and JavaScript.
Requirement 10	The system shall enable the adding and removing of calorie information of products	Calorie information are added and removed through HTML and JavaScript.
Requirement 11	The system shall enable order taking from customers	The system supports order taking from customers via e-mail.
Requirement 12	The system shall enable customizing of product design	The system allows customizing of product design.
Requirement 13	The system shall enable customizing of product calorie	The system allows customizing of product calorie.
Requirement 14	The system shall enable customizing of product price	The system allows customizing of product price.

Requirement 15	The system shall enable product ordering	The system allows customizing of product ordering.

6.4 Evaluation

From the testing, the system fulfills requirement 1, 2, 6, 7, 8, 9, 10, 11, 12, 13, 14 and 15. Requirement 3 was partly fulfilled because the system still requires a plug-in in order to achieve a universal format. As for requirement 4, to have universal access of the system, it has yet to be uploaded to the Internet for further testing. Requirement 5 wasn't achieved due to the reason that the system could not run properly on other browsers other than Microsoft Internet Explorer 4.x.

The testing proved that the system especially fulfills the customer side of requirements (requirement 11-15). However, improvements has to be made for the system to be more user friendly, to improve the price officer, health officer and graphic designer side of requirements fulfillment (requirement 7-10). This is because these system users have to update information through JavaScript databases which will be time consuming for those without much programming language skill.

6.5 Summary

This chapter covered the dynamic and procedural aspects of the system architecture. The actors and their interactions with the system and its architecture were also touched in this section. The chapter later proceeded with the design of the system's user interface, testing and results. As a summary of the result, the test is able to prove that the system supports most of the its requirements. However, the system must be made more user friendly for the non customer users. Also, the system still needs to be uploaded to the Internet for further testing and improvements.

Chapter 7 Conclusion

The conclusion chapter will review the project's overall progression. This includes problems and limitations encountered during the development of this project. Finally, this chapter will be ended with possible directions for future work related to the project.

7.1 Problems and Limitations

This project focuses on the graphical concerns of the product customizing system. Other concerns include the price and calorie calculating system. However, some problems and limitations discovered during and prior to the development of the project are:

- Although this is a web based system, it has yet to be uploaded to the web for the final testing due to time constraints

- This system is built for and tested on Internet Explorer 5 browser. It might encounter some problems on Netscape Navigator and other browsers.

- The graphical part of the system (using Metastream 3), can yet to be refined and improved for better appearance.

- The Metastream XML coding of the graphical part of the system can yet be refined and improved to maximize its performance.

- The system can be made more user friendly to aid the non customer users who need to operate the system but lack programming skills.

7.2 Future Design and Development Considerations

Throughout the development of this model system, several issues regarding design and development were discovered. Future design and development of related projects could be done based on these considerations:

- Develop and test the prototype system on the web using various browsers.

- Develop an educational or entertainment system using Metastream 3.0 technology.

- Develop an e-commerce site teaching customers how to assemble or use a product using Metastream 3.0 technology.

- Detailed study on various imaging technologies.

7.3 Summary

In the future, this prototype may be further developed into a fully functional system. In addition, this project can also serve as a throwaway prototype for large companies who might use it as an example system to develop a better and improved version of the system.

Bibliography

Amor, D. (2000). *The E-Business (R)evolution: Living and Working in an Interconnected World.* United States: Prentice Hall.

Angelo, M. (2000). *Metastream: 3D Gets A New Dimension. WebTools.* [www document]. URL. http://www.webtools.com/story/graphics/TLS20000407S0001

Cult3D official site. [www document]. URL. http://www.cult3d.com

Dong Pan. (1998). *The Application of Design Patterns in Knowledge Inference Engine.* Managing the Software Engineering Research Project.

Fournier, R. (1999). *A Methodology for Client/Server and Web Application Development.* United States: Prentice Hall.

Gotfried, C. (1999). *A Tentative Guide to Metastream.* 25 Sept. 2000. [www document]. URL. http://www.lookoutnow.com/3d/lookmeta.htm

Lindsay, G. (2000). *Selling in 3D.* Fortune. [www document]. URL. http://www.fortune.com/fortune/technology/2000/04/17/eco3.html

Mauldin, M. (1996). *The Development of Computer-Based Multimedia.* TechTrends: 15-19.

Metastream official site. [www document]. URL. http://www.metastream.com

Margherio, L., Henry, D., Cooke, S., et. al. (1999). *The Emerging Digital Economy.* Washington: Secretariat on Electronic Commerce.

Sree Kotay (2000). *Metastream 3 White Paper: Visual Fidelity.* [www document]. URL. http://www.metastream.com

Sree Kotay (2000). *Metastream 3 White Paper: Streaming and Scalability.* [www document]. URL. http://www.metastream.com

Sree Kotay (2000). *Metastream 3 White Paper: Interactivity and Animation.* [www document]. URL. http://www.metastream.com

Sree Kotay (2000). *Metastream 3 White Paper: Web Media Integration.* [www document]. URL. http://www.metastream.com

Sree Kotay (2000). *Metastream 3 White Paper: Texture Compression.* [www document]. URL. http://www.metastream.com

Sree Kotay (Feb, 2000). *Metastream 3 White Paper: The Metastream Hub.* [www document]. URL. http://www.metastream.com

___.(2000) . *ANDERSON CONSULTING: Anderson Consulting and Vignette form strategic alliance to help clients build successful customer-focused Internet businesses.* M2 Communications.

APPENDIX

USER MANUAL

Topic 1 System Requirements

The system requirements for Sweet Dreams Interactive Customizing Bakery Ordering System is as listed below:

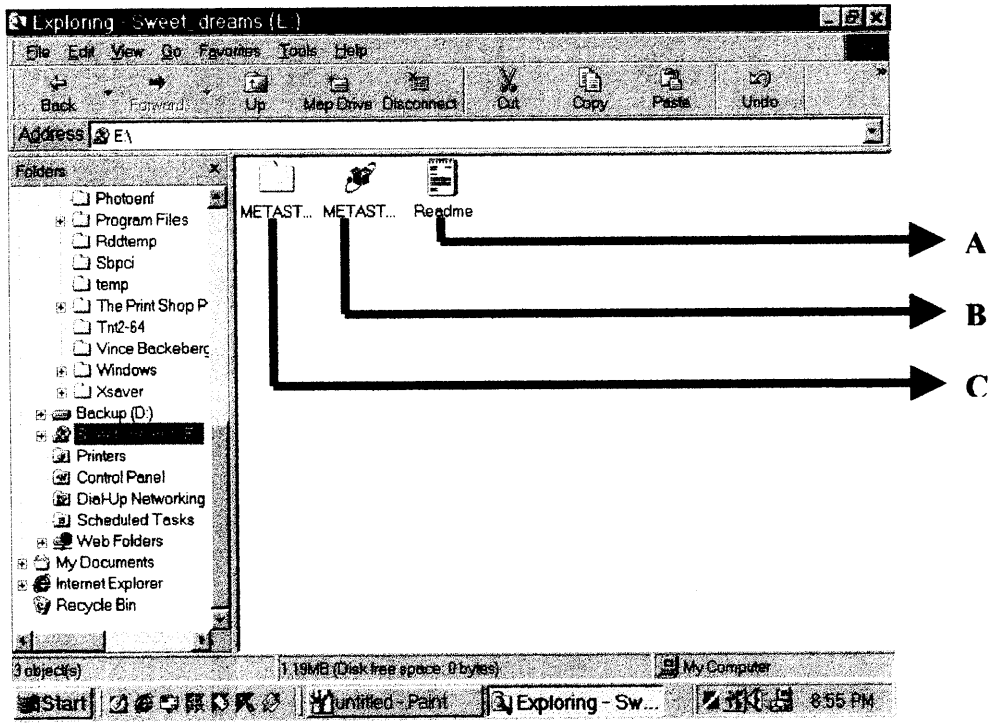
- Windows 95, 98, 2000 or NT
- Microsoft Internet Explorer 4.x or later
- 256 color display (24-bit recommended)
- 5 MB free disk space
- 32 MB RAM (64 recommended)
- Pentium 166 or faster (Pentium II)

Topic 2 Installing the System

Here are the steps to follow in order to install the system:

1. Start the computer.
2. Insert Sweet Dreams Interactive Customizing Bakery Ordering System CD into D drive (CD drive).
3. Open Windows Explorer and click on folder (D:).

Figure 2.1: Items in CD

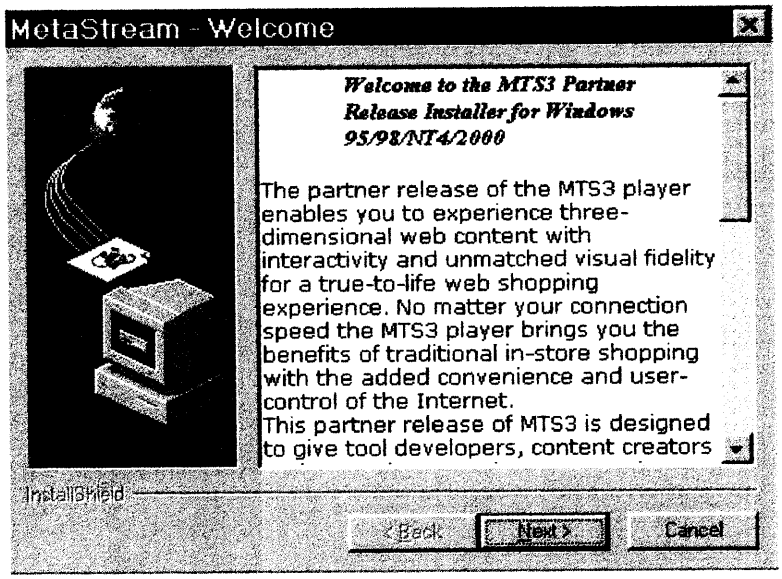


There are 3 items (shown in figure 2.1) in the folder:

Label	Description	Explanation
A	Readme File	Brief instruction for installing the system.
B	Metastream Full Installation	Metastream 3.0 plug-in full installation.
C	Metastream Folder	Folder containing files of the system.

4. Click on Metastream Full Installation. The installation window (figure 2.2) will appear. Follow the guided steps to install the full installation of the plug-in.

Figure 2.2: Metastream 3.0 Plug-in Installation Window



5. After installing the plug-in, copy the Metastream Folder to drive (C:) by clicking and dragging the folder to drive (C:).

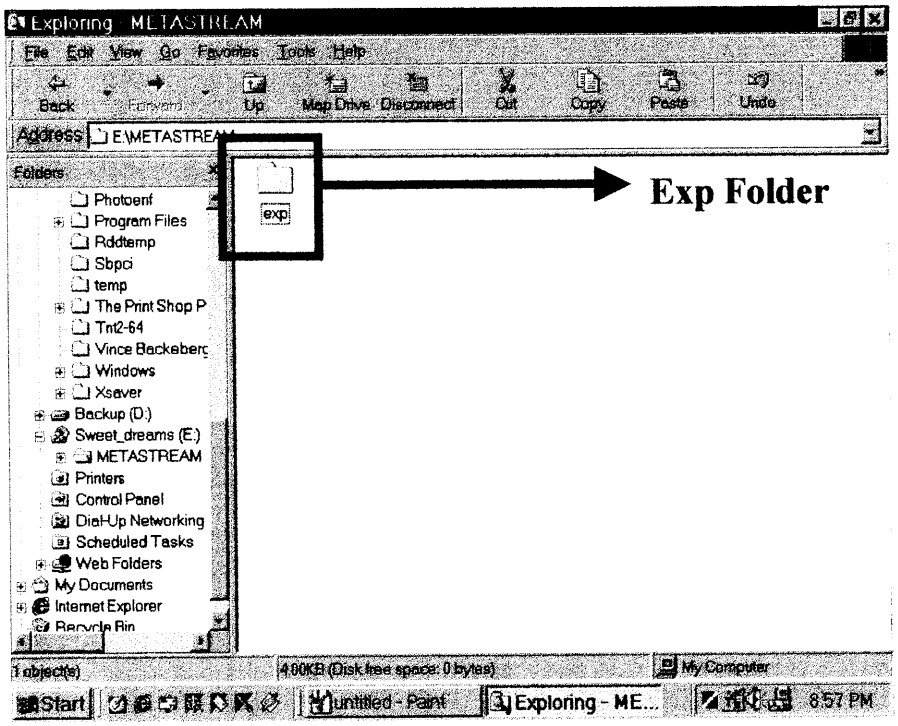
6. After following these steps the installation of the system is done.

Topic 3 Opening the System

Here are the steps to open and run the system:

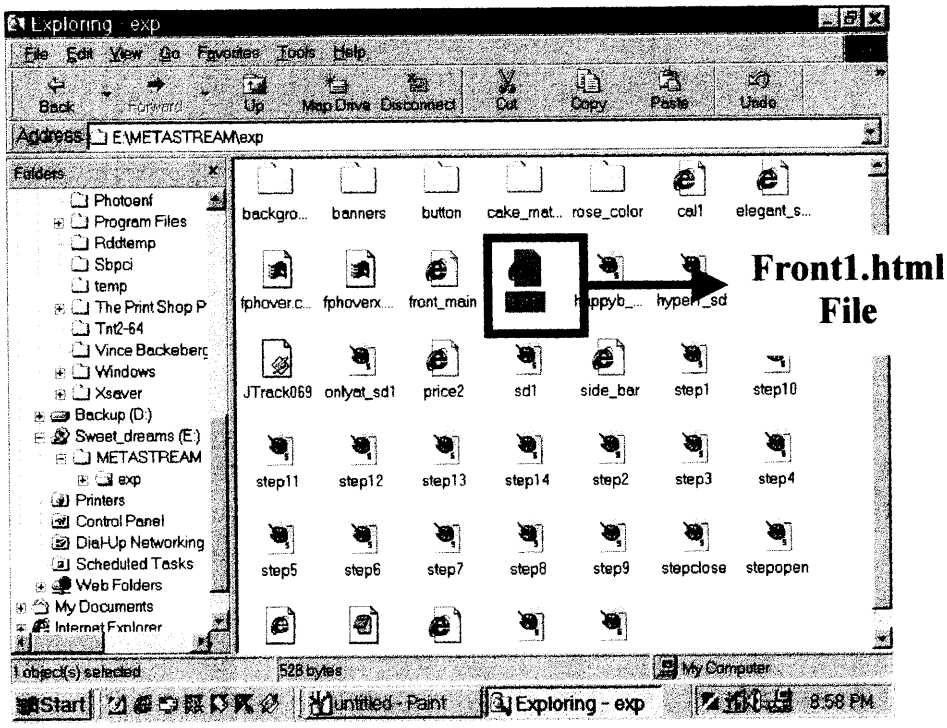
1. Click on the Metastream Folder. A folder by the name of Exp will appear. Click on the Exp Folder (figure 3.1).

Figure 3.1: Exp Folder Within Metastream Folder



2. There are several files and folders in the Exp Folder. Click on Front1.html file (figure 3.2). The main page of the system will appear.

Figure 3.2: Files and Folders Within Exp Folder

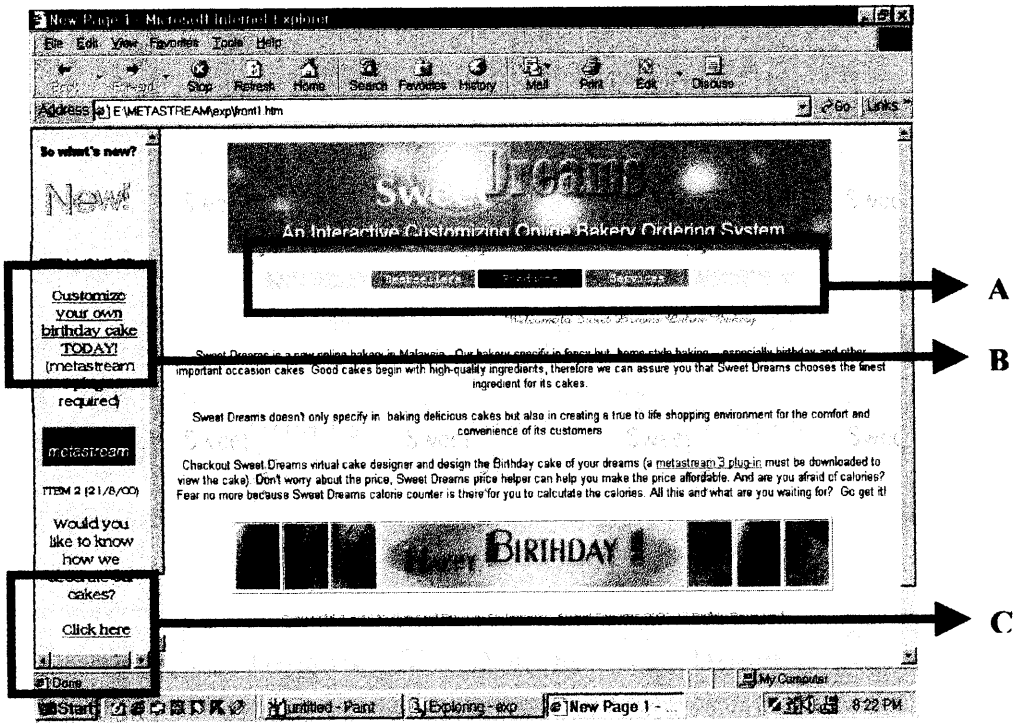


Topic 4 Using the System

4.1 Main Page

There are several important links on the Main Page (figure 4.1) of the system. While browsing the main page, the user can choose one of the following links.

Figure 4.1: System Main Page

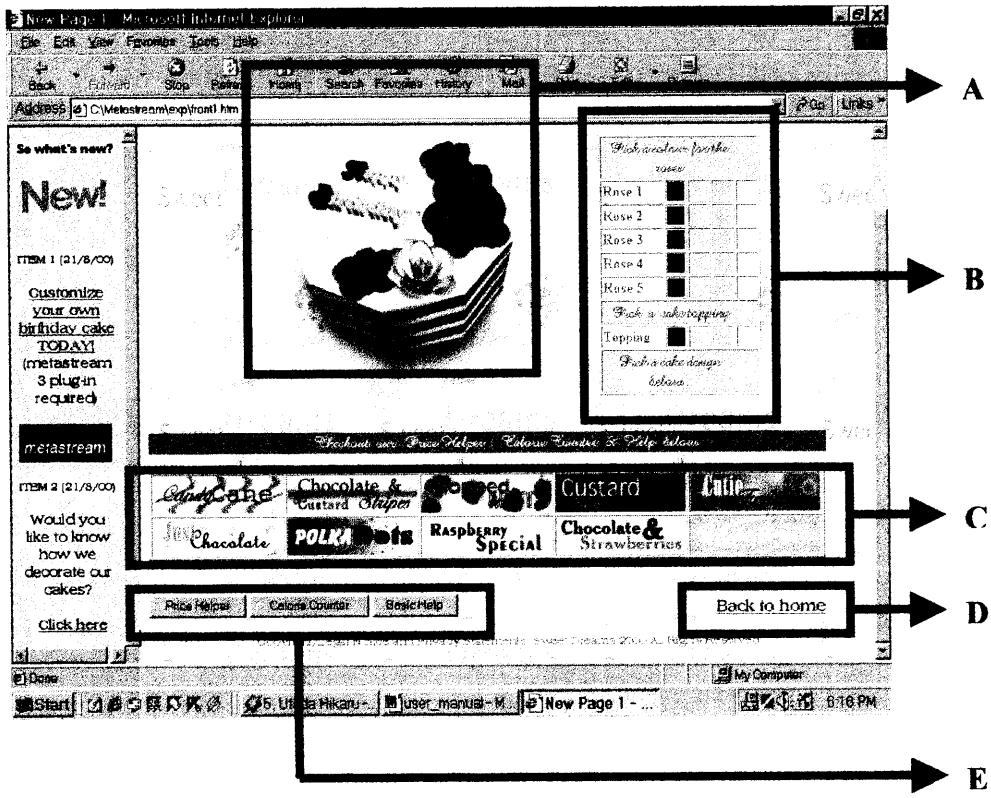


Label	Description	Explanation
A	Company and Product Information	Links to company and product information.
B	Product Customizing	Link to product customizing page.
C	Special Information	Link to product making tour.

4.2 Customizing the Product

Figure 4.2, depicts the product which is a cake, on the Product Customizing Page.

Figure 4.2: Product Customizing Page



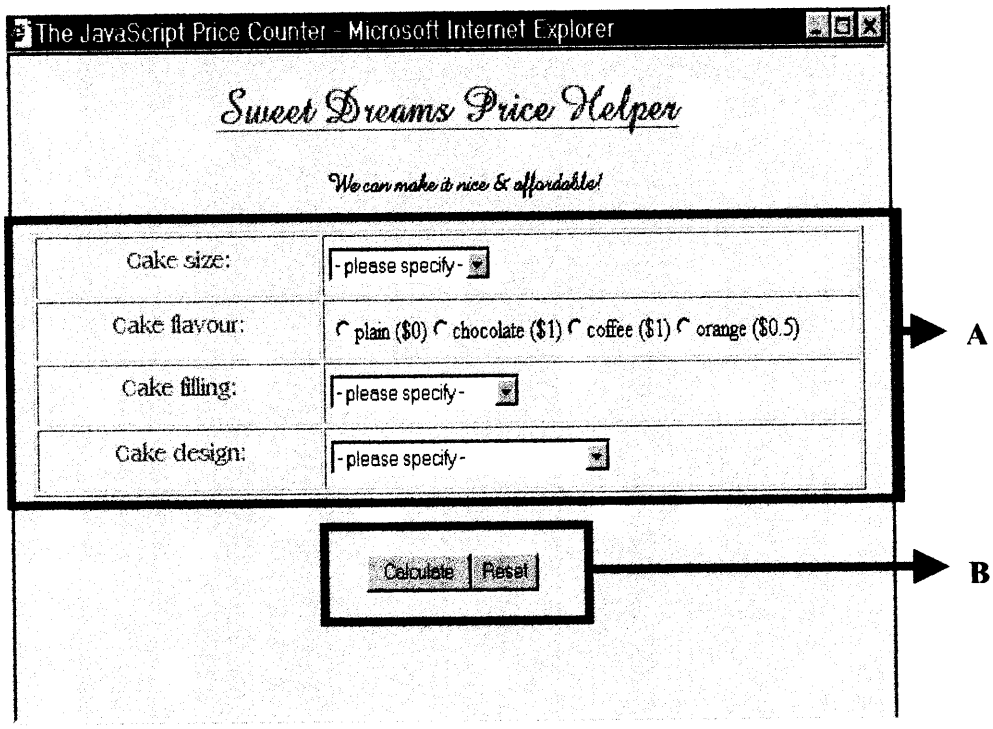
Label	Description	Explanation
A	Product Image	Image of a cake.
B	Topping Controls	Buttons to change the colour of the roses and topping of the cake image.
C	Side Design Controls	Buttons to change the side design of the cake image.
D	Home Page Link	Link to Main Page.
E	User Facilities	User facilities include: -Price Helper -Calorie Counter -Basic Help

On the Product Customizing Page, there will be a product image. The user may use the Topping Controls and the Side Design Controls to customize the product to satisfaction. Then the user may use the User Facilities, that is, the Price Helper, the Calorie Counter or Basic Help according to needs.

4.3 Price Helper

Price Helper is a user facility to aid the users in customizing an affordable price of the product. Figure 4.3, depicts the Price Helper.

Figure 4.3: Price Helper



Label	Description	Explanation
A	Price Information	Price information selections.
B	Calculate and Reset Button	Calculate and reset selections respectively.

To use the Price Helper, the user must select price information and click the Calculate button. A confirmation window (figure 4.4) will appear, confirming user selection and stating the total price of the product. If the user wishes to order, he or she may click the OK button, fill in the ordering form (figure 4.5) and click the Submit button. If not, the user may click the Cancel button on the confirmation window.

The user may then click the Reset button on the Price Helper window to reset the selections and make a new selection, or close the Price Helper window by clicking the Close control on the top right of the window.

Figure 4.4: Price Helper Confirmation

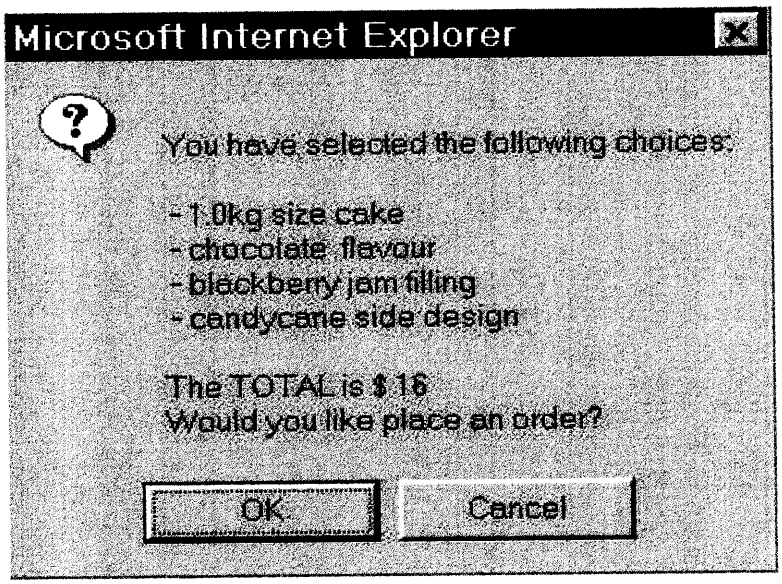
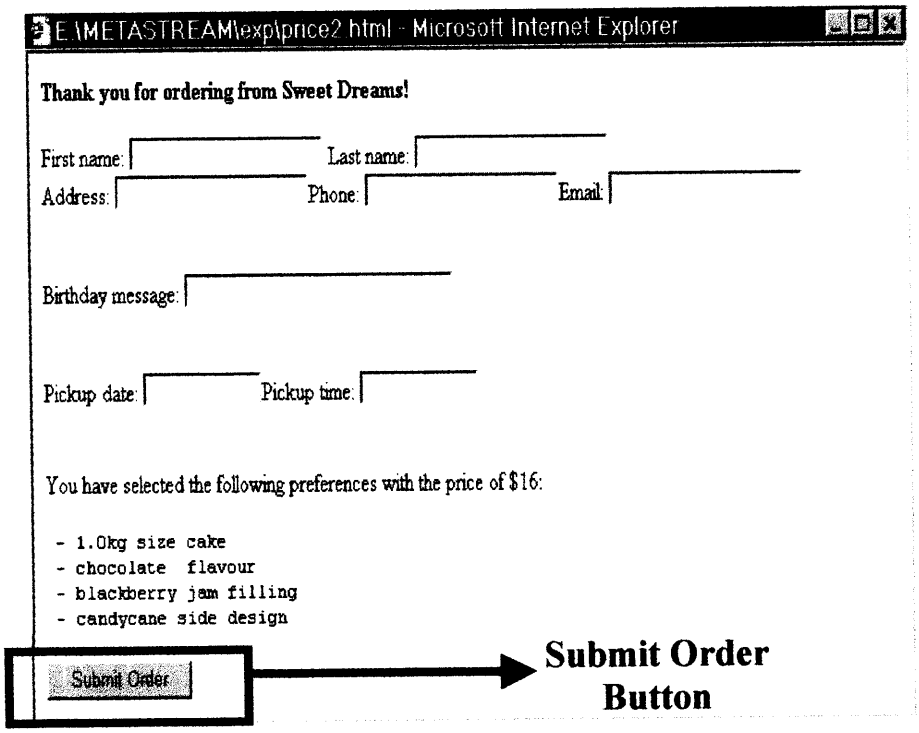


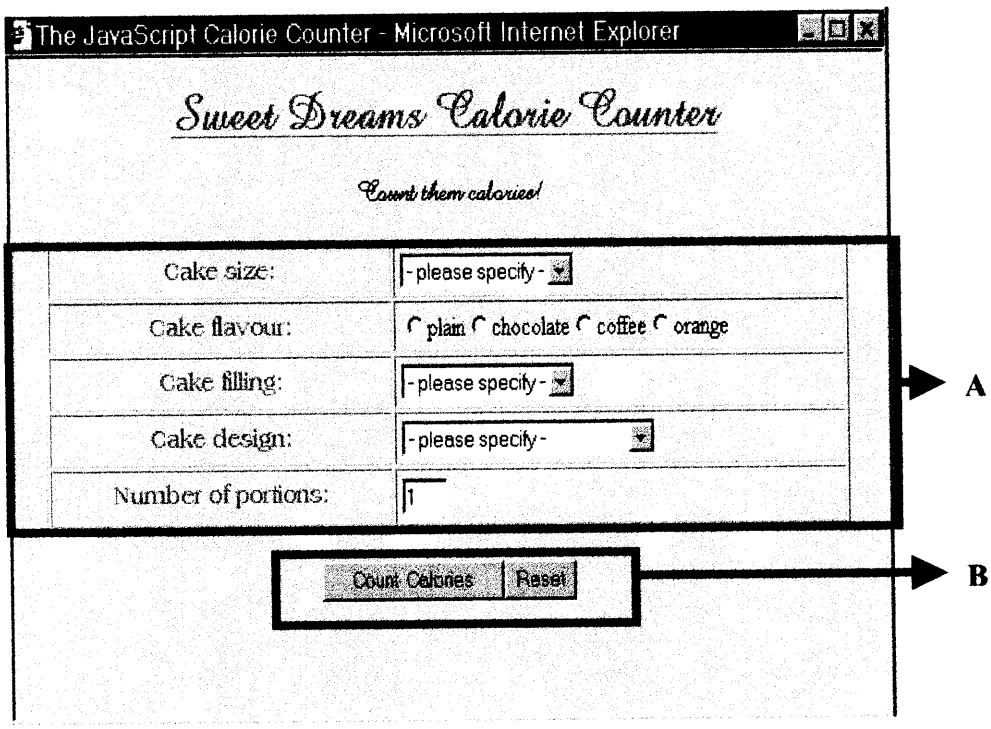
Figure 4.5: Order Form



4.4 Calorie Counter

Calorie Counter is a user facility to aid the users in counting the amount of calories of the product. Figure 4.6, depicts the Calorie Counter.

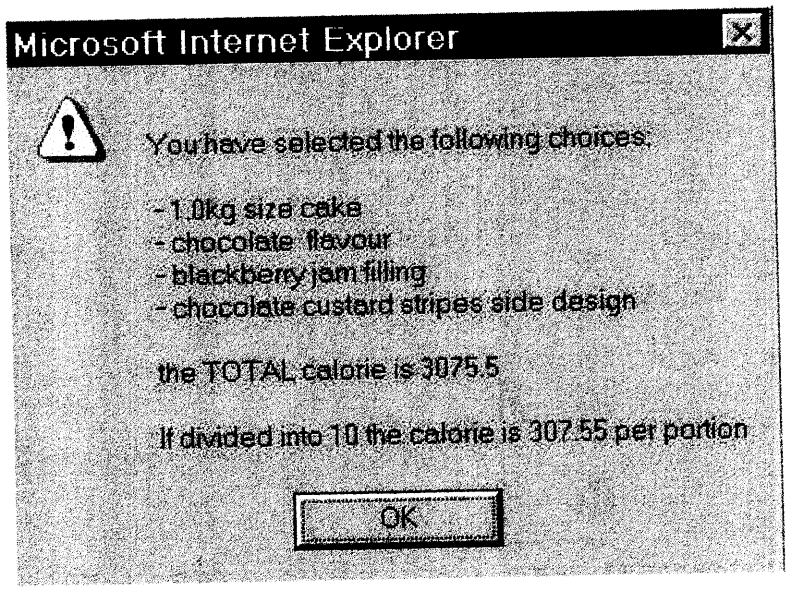
Figure 4.6: Calorie Counter



Label	Description	Explanation
A	Calorie Information	Calorie information selections.
B	Count Calories and Reset Button	Calculate calories and reset selections respectively.

To use the Calorie Counter, the user must select calorie information and click the Count Calories button for total calories (figure 4.7). The user may then click the Reset button to reset the selections and make new selections or close the Calorie Counter by clicking the Close control on the top right of the window.

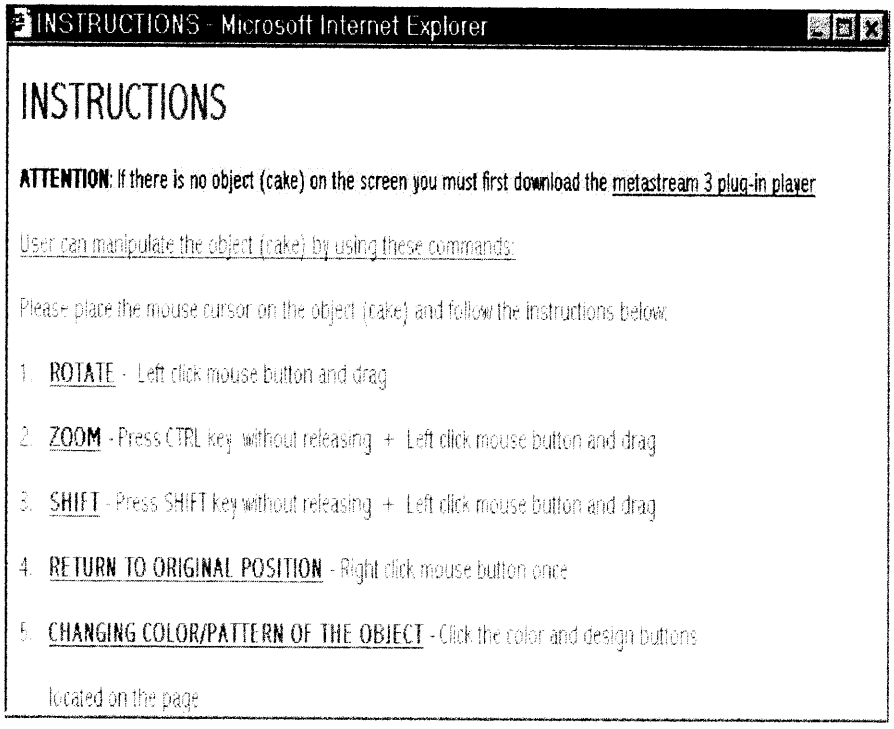
Figure 4.7: Total Calorie



4.5 Help

Basic Help is intended to give the user some basic guidance and instructions about the system. Figure 4.8 depicts the Basic Help window.

Figure 4.8: Basic Help



APPENDIX

SOURCE CODE

Main Page

```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>New Page 2</title>
<base target="_self">

<script language=javascript>
<!-- Hide script from old browsers
        var adImages = new
Array("happyb_sd2.jpg","onlyat_sd1.jpg","wecan_sd1.jpg","hyper1_sd.jpg","onlyat_sd1.j
pg","wecan_sd2.jpg");
        var thisAd = 0;
        var imgCt = 6;

        function cycle() {
                if(document.images) {
                        thisAd++;
                        if(thisAd==imgCt)       {
                                thisAd = 0;
                        }
                        document.adBanner.src=adImages[thisAd];
                        setTimeout("cycle()", 3*1000);
                }
        }
//End hiding script from old browsers -->
</script>

</head>

<body background="backgrounds/sweetdreams2.jpg" bgproperties="fixed"
onload="cycle()">

<p align="center"><img border="0" src="banners/sweetdreamslogo3.jpg" width="700"
height="120"></p>
<p align="center">
<applet code="fphover.class" codebase="./" width="120" height="20">
 <param name="textcolor" value="#FFFFFF">
```

```
  <param name="text" value="Store Info">
  <param name="fontstyle" value="regular">
  <param name="fontsize" value="14">
  <param name="font" value="Courier">
  <param name="color" value="#C8D6E6">
  <param name="effect" value="lightGlow">
 </applet>
 <applet code="fphover.class" codebase="./" width="120" height="20">
  <param name="textcolor" value="#FFFFFF">
  <param name="text" value="Bestsellers">
  <param name="color" value="#88A4D7">
  <param name="fontstyle" value="regular">
  <param name="fontsize" value="14">
  <param name="font" value="Courier">
  <param name="effect" value="lightGlow">
 </applet>
 <applet code="fphover.class" codebase="./" width="120" height="20">
  <param name="textcolor" value="#FFFFFF">
  <param name="text" value="Products">
  <param name="color" value="#02317D">
  <param name="effect" value="lightGlow">
  <param name="font" value="Courier">
  <param name="fontstyle" value="regular">
  <param name="fontsize" value="14">
 </applet>
 <applet code="fphover.class" codebase="./" width="120" height="20">
  <param name="textcolor" value="#FFFFFF">
  <param name="text" value="Services">
  <param name="font" value="Courier">
  <param name="fontstyle" value="regular">
  <param name="fontsize" value="14">
  <param name="color" value="#88A4D7">
  <param name="effect" value="lightGlow">
 </applet>
 <applet code="fphover.class" codebase="./" width="120" height="20">
  <param name="textcolor" value="#FFFFFF">
  <param name="text" value="Contact Us">
  <param name="font" value="Courier">
  <param name="fontstyle" value="regular">
  <param name="fontsize" value="14">
  <param name="color" value="#C8D6E6">
  <param name="hovercolor" value="#FFFFFF">
  <param name="effect" value="lightGlow">
 </applet>
 </p>
 <p align="center"><font color="#808080" face="Amazone BT" size="4">
 <marquee behavior="alternate" direction="right">Welcome to Sweet Dreams Online
 Bakery</marquee>
 </font></p>
 <p align="center"><font size="2" face="arial, helvetica">Sweet Dreams is a new online
 bakery in Malaysia.&amp;nbsp; Our bakery specify in fancy but&amp;nbsp; home-style
```

```
baking -- especially birthday and other important occasion cakes. Good cakes
begin with high-quality ingredients, therefore we can assure you that Sweet
Dreams chooses the finest ingredient for its cakes.</font></p>
<p align="center"><font size="2" face="arial, helvetica">Sweet Dreams doesn't
only specify in  baking delicious cakes but also in creating a true to life
shopping environment for the comfort and convenience of its customers.</font></p>
<p align="center"><font size="2" face="arial, helvetica">Checkout Sweet Dreams
virtual cake designer and design the Birthday cake of your dreams (a <a
href="http://www.metastream.com">metastream
3 plug-in</a> must be downloaded to view the cake). Don't worry about the price,
Sweet Dreams price helper can help you make the price affordable. And are you
afraid of calories? Fear no more because Sweet Dreams calorie counter is there
for you to calculate the calories. All this and what are you waiting for? 
Go get it! </font></p>
<center>
<table border="1" width="59%" height="3">
 <tr>
   <td width="11%" height="1" align="center"><img border="0"
src="banners/cake_decor1.jpg" width="50" height="70"></td>
   <td width="11%" height="1" align="center"><img border="0"
src="banners/cake_decor2.jpg" width="50" height="70"></td>
   <td width="11%" height="1" align="center"><img border="0"
src="banners/cake_decor5.jpg" width="50" height="70"></td>
   <td width="12%" height="1" align="center">
    <p align="left"><img border="0" src="sd1.jpg" width="350" height="70"
name="adBanner" ></td>
   <td width="12%" height="1" align="center"><img border="0"
src="banners/cake_decor4.jpg" width="50" height="70"></td>
   <td width="13%" height="1" align="center"><img border="0"
src="banners/cake_decor1.jpg" width="50" height="70"></td>
   <td width="35%" height="1" align="center"><img border="0"
src="banners/cake_decor3.jpg" width="50" height="70"></td>
 </tr>
</table>
<center>
<p align="center"><font face="Americana BT" color="#808080"
size="2"><b>Copyright/Legal
Notice and Privacy Statements  Sweet Dreams 2000 All Rights
Reserved</b></font></p>
<p align="center"><font size="2" face="arial, helvetica"> </font></p>
<p align="center"> </p>
<p align="center"> </p>

</body>

</html>
```

Side Page

```
<html>

<head>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>New Page 3</title>
<base target="main">
<bgsound src="file:///Metastream/exp/JTrack069.mp3" loop="-1">
</head>

<body background="backgrounds/sweetdreams_side.jpg" bgproperties="fixed">

<p align="center"><font face="Arial Black" size="2"><span style="background-color:
#FFFF00">So
what's new?</span></font><img border="0" src="button/new_but.gif" width="117"
height="84"></p>
<p align="center"><font face="Americana BT" size="2"><span style="background-color:
#FFFFFF">ITEM
1 (21/8/00)</span></font></p>

<p align="center"><font face="Americana BT"><a
href="elegant_sweet.htm"><b>Customize
your own birthday cake TODAY!</b></a> (metastream 3 plug-in required)</font></p>

<p><a href="http://www.metastream.com"><img border="0" src="button/mts3_but.jpg"
width="107" height="41"></a></p>

<p align="center"><font face="Americana BT" size="2"><span style="background-color:
#FFFFFF">ITEM
2 (21/8/00)</span></font></p>

<p align="center"><font face="Americana BT">Would you like to know how we
decorate our cakes?</font></p>

<p align="center"><font face="Americana BT">  <b><a href="tour1.htm">Click
here</a></b></font></p>
<p align="center"> </p>

<p align="center"><b><font color="#808080" size="4" face="Americana BT"><a
href="front_main.htm"><span style="background-color:
#FFFF00">HOME</span></a></font></b></p>

</body>

</html>
```

Product Page

```
<html>

<head>
<meta http-equiv="Content-Language" content="en-us">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>Elegant</title>

<script language="JavaScript" SRC="trigger.js">
</script>
<script language="JavaScript">
<!--
var isIE4       = navigator.appName == "Microsoft Internet Explorer" &&
parseInt(navigator.appVersion.substring(0,1)) >= 4;
function mtspluginfunction()
{
        var argstr;
        argstr          = mtspluginfunction.arguments[0] + "(" + arguments[1] + ")";
        if (isIE4)      document.all.MetaStreamCtl.PluginCommand(argstr, 0, 0);
        else            document.MetaStreamCtl.DoCommand(argstr);
}
function triggeranimation()     {mtspluginfunction ("TriggerAnimation", arguments[0]);}
function reverseanimation ()    {mtspluginfunction ("ReverseAnimation",
arguments[0]);}
function stopanimation ()       {mtspluginfunction ("StopAnimation", arguments[0]);}
function startanimation ()      {mtspluginfunction ("StartAnimation", arguments[0]);}
function togglecollapsed()      {mtspluginfunction ("ToggleCollapsed", arguments[0]);}
//-->
</script>

</head>

<body background="backgrounds/sweetdreams3.jpg">

<p align="center" style="line-height: 100%"><b><font face="Americana BT"
color="#FFFFFF" size="6"><span style="background-color: #808000"> An
Elegant
&amp; Sweet Collection </span></font></b></p>

<p align="center" style="line-height: 100%"><font size="5" face="Amazone BT"
color="#808080">Customize
an elegant &amp; sweet birthday cake for your loved ones today</font></p>


<table border="0" width="90%" height="319" cellspacing="0" cellpadding="0">
```

```
  <tr>
   <td width="70%"
height="313">          &
nbsp;           &nb
sp;
   <object id="MetaStreamCtl"
                  classid="CLSID:03F998B2-0E00-11D3-A498-00104B6EB52E"
                  width=387
                  height=319

        codebase="http://components.metastream.com/MTSInstallers/MetaStream3.cab#V
ersion=3,0,2,47">
        <param name="Source" value="sw_el_4.mtx">
        <param name="Component" value="ISceneComponent">
        <param name="ComponentFileName" value="SceneComponent.mtc">
                <param name="BroadcastKeyFileURL" value="">
                <embed component="ISceneComponent"
componentfilename="SceneComponent.mtc" src="sw_el_4.mtx" type="application/x-mtx"
width="400" height="400" border="true" script="true" name="MetaStreamCtl"
BroadcastKeyFileURL>
        </object>
   </td>
   <td width="30%" height="313">
    <table border="1" width="188">
     <tr>
      <td width="157" colspan="5" background="backgrounds/sweetdreams_side.jpg">
       <p align="center"><font face="Amazone BT" size="4" color="#808080">Pick
       a colour for the roses</font></td>
     </tr>
     <tr>
      <td width="124" bgcolor="#FFFFFF"><font color="#808080"><b>Rose
1</b></font></td>
      <td width="27" height="20" bgcolor="#FFFFFF"><img border="0"
src="button/color_button/button_red.jpg" width="20" height="20"
onClick="triggeranimation('txtani_rose5_red')"></td>
      <td width="24" height="20" bgcolor="#FFFFFF"><img border="0"
src="button/color_button/button_pink.jpg" width="20" height="20"
onClick="triggeranimation('txtani_rose5_pink')"></td>
      <td width="29" height="20" bgcolor="#FFFFFF"><img border="0"
src="button/color_button/button_yellow.jpg" width="20" height="20"
onClick="triggeranimation('txtani_rose5_yellow')"></td>
      <td width="27" height="20" bgcolor="#FFFFFF"><img border="0"
src="button/color_button/button_white.jpg" width="20" height="20"
onClick="triggeranimation('txtani_rose5_white')"></td>
     </tr>
     <tr>
      <td width="124" align="left" bgcolor="#FFFFFF"><font
color="#808080"><b>Rose
       2</b></font></td>
```

```
        <td width="27" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_red.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose2_red')"></td>
        <td width="24" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_pink.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose2_pink')"></td>
        <td width="29" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_yellow.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose2_yellow')"></td>
        <td width="27" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_white.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose2_white')"></td>
      </tr>
      <tr>
        <td width="124" align="left" bgcolor="#FFFFFF"><font
 color="#808080"><b>Rose
         3</b></font></td>
        <td width="27" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_red.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose1_red')"></td>
        <td width="24" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_pink.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose1_pink')"></td>
        <td width="29" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_yellow.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose1_yellow')"></td>
        <td width="27" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_white.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose1_white')"></td>
      </tr>
      <tr>
        <td width="124" align="left" bgcolor="#FFFFFF"><font
 color="#808080"><b>Rose
         4</b></font></td>
        <td width="27" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_red.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose4_red')"></td>
        <td width="24" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_pink.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose4_pink')"></td>
        <td width="29" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_yellow.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose4_yellow')"></td>
        <td width="27" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_white.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose4_white')"></td>
      </tr>
      <tr>
        <td width="124" align="left" bgcolor="#FFFFFF"><font
 color="#808080"><b>Rose
         5</b></font></td>
```

```
        <td width="27" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_red.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose3_red')"></td>
        <td width="24" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_pink.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose3_pink')"></td>
        <td width="29" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_yellow.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose3_yellow')"></td>
        <td width="27" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_white.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_rose3_white')"></td>
      </tr>
      <tr>
       <td width="157" colspan="5" background="backgrounds/sweetdreams_side.jpg">
        <p align="center"><font size="4" face="Amazone BT"
 color="#808080">Pick 
        a  cake topping </font></td>
      </tr>
      <tr>
       <td width="124" bgcolor="#FFFFFF"><font
 color="#808080"><b>Topping</b></font></td>
        <td width="27" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_choc.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_top_choco')"></td>
        <td width="24" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_pink.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_top_straw')"></td>
        <td width="29" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_yellow.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_top')"></td>
        <td width="27" height="20" bgcolor="#FFFFFF"><img border="0"
 src="button/color_button/button_white.jpg" width="20" height="20"
 onClick="triggeranimation('txtani_top_whipcream')"></td>
      </tr>
      <tr>
       <td width="157" colspan="5" background="backgrounds/sweetdreams_side.jpg">
        <p align="center"><font color="#808080" face="Amazone BT" size="4">Pick
        a cake design  below...</font></td>
      </tr>
    </table>
   </td>
  </tr>
</table>
<p><font face="Amazone BT" color="#FFFFFF" size="4">
<marquee direction="right" behavior="alternate" bgcolor="#808000" width="786"
height="23">Checkout 
our   Price Helper  ,  Calorie Counter  &amp; 
Help  below...</marquee>
</font></p>
<table border="1" width="83%">
```

```
  <tr>
    <td width="18%"><img border="0" src="button/javabuttons/but_candy.jpg"
width="150" height="40" onClick="triggeranimation('txtani_candycane')"></td>
    <td width="19%"><img border="0" src="button/javabuttons/but_chocstripes.jpg"
width="150" height="40" onClick="triggeranimation('txtani_chococustripe')"></td>
    <td width="19%"><img border="0" src="button/javabuttons/but_chopped.jpg"
width="150" height="40" onClick="triggeranimation('txtani_choppednuts')"></td>
    <td width="19%"><img border="0" src="button/javabuttons/but_cust.jpg" width="150"
height="40" onClick="triggeranimation('txtani_custstrawkiwi')"></td>
    <td width="25%"><img border="0" src="button/javabuttons/but_cutie.jpg"
width="150" height="40" onClick="triggeranimation('txtani_cutiefrutie')"></td>
  </tr>
  <tr>
    <td width="18%"><img border="0" src="button/javabuttons/but_justchoc.jpg"
width="150" height="40" onClick="triggeranimation('txtani_justchoc')"></td>
    <td width="19%"><img border="0" src="button/javabuttons/but_polka.jpg"
width="150" height="40" onClick="triggeranimation('txtani_polkadot')"></td>
    <td width="19%"><img border="0" src="button/javabuttons/but_rasp.jpg" width="150"
height="40" onClick="triggeranimation('txtani_raspspecial')"></td>
    <td width="19%"><img border="0" src="button/javabuttons/but_strawchoc.jpg"
width="150" height="40" onClick="triggeranimation('txtani_chocostraw')"></td>
    <td width="25%"><img border="0" src="button/javabuttons/but_whip.jpg"
width="150" height="40" onClick="triggeranimation('txtani_whipcream')"></td>
  </tr>
</table>

<p align="left">

<br>
<input type="button" name="Price Helper" value="Price Helper"
onClick="window.open('price2.html', 'PriceWindow', 'width=700, height=420')">

<input type="button" name="Calorie Counter" value="Calorie Counter"
onClick="window.open('cal1.html', 'CalWindow', 'width=650, height=400')">

<input type="button" name="Instructions" value="Basic Help"
onClick="window.open('instruct1.htm', 'InstructWindow', 'width=700, height=400')">

<font color="#808080" size="4" face="Americana BT"><a href="front_main.htm"><span
style="background-color: #FFFF00">Back to Home</span></a>

<p align="center"><font face="Americana BT" color="#808080"
size="2"><b>Copyright/Legal
Notice and Privacy Statements  Sweet Dreams 2000 All Rights
Reserved</b></font></p>

</body>
```

Price Helper

```
<HTML>
<HEAD>
<TITLE>The JavaScript Price Counter</TITLE>

<SCRIPT LANGUAGE="JavaScript">
<!-- hide from non-JavaScript browsers
function MakeArray(size) {
  this.length = size;

  for(i = 1; i <= size; i++)
    this[i] = null;

  return this;
}

function Choice(c_Name, c_Price ) {
  this.name  = c_Name;
  this.price  = c_Price;
}


////////////////////////////<--CAKE CHOICE ITEM & PRICES-->////////////////////////

//ADD or EDIT choice items can be done in part 1 to 4
//each part consist of name and prices of choice of desired cake
//to add a choice item, change the MakeArray number and follow below sample code
//weightList[--type appropriate index--] = new Choice("--type choice item--", "--price--");


//part 1: size
weightList      = new MakeArray(4);
weightList[1]   = new Choice("1.0kg", 8.00);
weightList[2]   = new Choice("1.5kg", 12.00);
weightList[3]   = new Choice("2.0kg", 16.00);
weightList[4]   = new Choice("2.5kg", 20.00);
weightList[5]   = new Choice("3.0kg", 24.00);

//part 2: flavour
flavourList     = new MakeArray(4);
flavourList[1]  = new Choice("plain", 0);
flavourList[2]  = new Choice("chocolate", 1.00);
flavourList[3]  = new Choice("coffee", 1.00);
flavourList[4]  = new Choice("orange", 0.50);

//part 3: filling
fillingList     = new MakeArray(5);
fillingList[1]  = new Choice("blackberry jam", 2.00);
fillingList[2]  = new Choice("orange jam", 1.00);
fillingList[3]  = new Choice("plum jam", 2.00);
fillingList[4]  = new Choice("raspberry jam", 1.00);
```

```
  fillingList[5]    = new Choice("strawberry jam", 1.00);

  //part 4: design
  designList        = new MakeArray(10);
  designList[1]     = new Choice("candycane", 5.00);
  designList[2]     = new Choice("chocolate custard stripes", 6.00);
  designList[3]     = new Choice("chocolate and strawberries", 7.00);
  designList[4]     = new Choice("chopped nuts", 5.00);
  designList[5]     = new Choice("custard strawberry and kiwi", 5.00);
  designList[6]     = new Choice("cutie frutie", 5.00);
  designList[7]     = new Choice("just chocolate", 6.00);
  designList[8]     = new Choice("polkadot", 4.00);
  designList[9]     = new Choice("raspberry special", 5.00);
  designList[10]    = new Choice("whipcream", 4.00);

  ////////////////////////<--END OF CAKE CHOICE ITEM & PRICES-->//////////////////////


 totalPrice    = 0;
 strDesc       = "";

 function createSelect(strName, objList) {
   var tStr = "<SELECT NAME=\"" + strName + "\">"
         + "<OPTION SELECTED>- please specify -";

   for(var i=1; i<=objList.length; i++)
     tStr += "<OPTION>" + objList[i].name + ".....$ " + objList[i].price;

   tStr += "</SELECT>";

   return tStr;
 }

 function createRadio(strName, objList) {
   var tStr = "";

   for(var i=1; i<=objList.length; i++) {
     tStr += "<INPUT NAME=\"" + strName + "\" "
         + "TYPE=radio VALUE=\"" + objList[i].name + "\">"
         + objList[i].name + " ($" + objList[i].price + ")";
   }

   return tStr;
 }

 function calculate() {
   totalPrice  = 0;
   strDesc     = "";

   if(!calcSize())
     return false;
```

```
  if(!calcFlavour())
    return false;

  if(!calcFilling())
    return false;

  if(!calcDesign())
    return false;


  return confirm("\nYou have selected the following choices:\n\n" +
          strDesc + "\nThe TOTAL is $ " +
          totalPrice + "\nWould you like place an order?");
}


function calcSize() {
  var which = document.order1.c_size.selectedIndex;

  if(which == 0) {
    alert("\nYou must select a cake size.\n");
    return false;
  }

  totalPrice += eval(weightList[which].price);
  strDesc  += " - " + weightList[which].name + " size cake\n";

  return true;
}

function calcFlavour() {
  var which = 0, j = -1;

  for(var i=0; i<document.order1.c_flavour.length; i++) {
    if(document.order1.c_flavour[i].checked) {
      j = i;
      break;
    }
  }

  if(j == -1) {
    alert("\nYou must select a cake flavour.\n");
    return false;
  }

  for(var i=1; i<=flavourList.length; i++) {
    if(document.order1.c_flavour[j].value == flavourList[i].name)
      which = i;
  }
```

```
    totalPrice += eval(flavourList[which].price);
    strDesc  += " - " + flavourList[which].name + "  flavour\n";

    return true;
  }

  function calcFilling() {
    var which = document.order1.c_filling.selectedIndex;

    if(which == 0) {
      alert("\nYou must select a monitor.\n");
      return false;
    }

    totalPrice += eval(fillingList[which].price);
    strDesc  += " - " + fillingList[which].name + " filling\n";

    return true;
  }

  function calcDesign() {
    var which = document.order1.c_design.selectedIndex;

    if(which == 0) {
      alert("\nYou must select a cake design.\n");
      return false;
    }

    totalPrice += eval(designList[which].price);
    strDesc  += " - " + designList[which].name + " side design\n";

    return true;
  }

  function order() {
    document.write("<body bgcolor=#ffffff>");
    document.write("<h4>Thank you for ordering from Sweet Dreams!</h4>");
    document.write("<FORM name='order2' METHOD=POST
ACTION='mailto:joanne@yahoo.com'>");
    document.write("First name: <INPUT NAME='first' TYPE=TEXT> ");
    document.write("Last name: <INPUT NAME='last' TYPE=TEXT><br>");
    document.write("Address: <INPUT NAME='address' TYPE=TEXT" + "size=50>");
    document.write("Phone: <INPUT NAME='phone' TYPE=TEXT>");
    document.write("Email: <INPUT NAME='email' TYPE=TEXT><br><br><br>");
    document.write("Birthday message: <INPUT NAME='message' TYPE=TEXT
size=30><br><br><br>");
    document.write("Pickup date: <INPUT NAME='pickdate' TYPE=TEXT size=10>");
    document.write("Pickup time: <INPUT NAME='picktime' TYPE=TEXT
size=10><br><br><br>");
    document.write("You have selected the following preferences "
            + "with the price of $" + totalPrice + ":"
```

```
            + "<PRE>" + strDesc + "</PRE>");
    document.write("<INPUT value='Submit Order' TYPE=SUBMIT><br>");
    document.write("<INPUT NAME='description' TYPE=HIDDEN>");
    document.write("<INPUT NAME='cost' TYPE=HIDDEN></form></body>");

    document.order2.description.value = strDesc;
    document.order2.cost.value=totalPrice;
  }

  // end hide -->
  </SCRIPT>
  </HEAD>

  <BODY BGCOLOR=#FFFFFF background="backgrounds/sweetdreams_side.jpg">
  <FORM NAME="order1" METHOD=POST >

  <H3 align="center"><u><font face="Amazone BT" size="6" color="#808080">Sweet
  Dreams Price Helper</font></u></H3>

  <p align="center">
  <font face="Amazone BT" size="4" color="#000000">
  <marquee behavior="alternate" direction="right">We can make it nice &
  affordable!</marquee>
  </font>
  </p>

  <TABLE WIDTH=665 CELLPADDING=4 BORDER="1" height="167"
  ALIGN="center">

    <TR>
      <TH VALIGN=TOP width="217" height="19">
        <p align="center"><font color="#808080" face="Americana BT">Cake
  size:</font></p>
      </TH>

  <CENTER>
      <TD width="420" height="19">
        <SCRIPT LANGUAGE="JavaScript">
        <!-- begin hide
          document.write(createSelect("c_size", weightList) + "<BR>");
        // end hide -->
        </SCRIPT>
      </TD>
    </TR>

    </center>

    <TR>
      <TH VALIGN=TOP width="217" height="31">
        <p align="center"><font color="#808080" face="Americana BT">Cake
  flavour:</font></p>
```

```
      </TH>

  <CENTER>
      <TD width="420" height="19">
        <SCRIPT LANGUAGE="JavaScript">
        <!-- begin hide
          document.write(createRadio("c_flavour", flavourList) + "<BR>");
        // end hide -->
        </SCRIPT>
      </TD>
    </TR>

    </center>

    <TR>
      <TH VALIGN=TOP width="217" height="19">
       <p align="center"><font color="#808080" face="Americana BT">Cake
filling:</font></p>
      </TH>

  <CENTER>
      <TD width="420" height="19">
        <SCRIPT LANGUAGE="JavaScript">
        <!-- begin hide
          document.write(createSelect("c_filling", fillingList) + "<BR>");
        // end hide -->
        </SCRIPT>
      </TD>
    </tr>

    </center>

    <TR>
      <TH VALIGN=TOP width="217" height="19">
       <p align="center"><font color="#808080" face="Americana BT">Cake
design:</font></p>
      </TH>

  <CENTER>
      <TD width="420" height="19">
        <SCRIPT LANGUAGE="JavaScript">
        <!-- begin hide
          document.write(createSelect("c_design", designList) + "<BR>");
        // end hide -->
        </SCRIPT>
      </TD>
    </TR>

    </center>

  </TABLE>
```

```
<p align="center">

<BR>

<INPUT TYPE=button NAME="compute" VALUE="Calculate"  onClick="if(calculate())
order();"><input type="reset" value="Reset" name="reset_price">

</FORM>
</BODY>
</HTML>
```

Metastream XML

```
<?xml version="1.0"?>
<!-- MTS3 scene description file.                                    -->
<!-- Created: Mon Aug 14 17:33:26 2000                               -->
<!-- Creator: MTS Scene Builder (Build 3.0.2.58) -->

<MTSScene Version="3" >
        <MTSSceneParms EdgeBias="1" >
                <Transform Type="current" >
                        <Scale x="0.17679" y="0.177" z="0.16837" />
                        <Shear xy="0" yz="0" xz="0" />
                        <Rotate x="-2" y="0" z="0" />
                        <Position x="0.364" y="-0.18606" z="0.0955" />
                </Transform>
        </MTSSceneParms>
        <MTSCamera OrbitDist="4" >
                <Rotate x="60.77914" y="25.7831" z="0" />
                <Translate x="-0.16" y="-0.68" z="0" />
                <ViewLocation x="0" y="0" z="4" />
        </MTSCamera>
        <MTSInstance Name="sw_el_1" >
                <MTSInstance Name="sw_el_1" >
                        <MTSInstance Name="sw_el_3_MESH_0" />
                        <MTSInstance Name="sw_el_3_MESH_1" >
                                <Transform>
                                        <Scale x="1" y="1" z="1" />
                                        <Shear xy="0" yz="0" xz="-0" />
                                        <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                                        <Position x="-2.5625" y="1.3125" z="-0.0625"
/>
                                </Transform>
                                <MTSMaterial Name="Mat_0" ID="0">
                                        <MTSTextureMap Type="Diffuse"
Name="text_rose1"

        Path="C:\Metastream\exp\rose_color\rose_red.jpg" Width="200" Height="300">
                                        </MTSTextureMap>
                                <MTSColor Name="Diffuse" r="0.49804" g="0" b="0"
Opacity="0">
                                </MTSColor>
                                </MTSMaterial>
                        </MTSInstance>
                        <MTSInstance Name="sw_el_3_MESH_2" >
                                <Transform>
                                        <Scale x="1" y="1" z="1" />
                                        <Shear xy="0" yz="0" xz="-0" />
                                        <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                                        <Position x="-3.0625" y="1.25" z="0" />
```

```
                        </Transform>
                        <MTSMaterial Name="Mat_1" ID="0">
                                <MTSTextureMap Type="Diffuse"
Name="text_rose2"

        Path="C:\Metastream\exp\rose_color\rose_red.jpg" Width="200" Height="300">
                                </MTSTextureMap>
                        <MTSColor Name="Diffuse" r="0.49804" g="0" b="0"
Opacity="0">
                        </MTSColor>
                        </MTSMaterial>
                </MTSInstance>
                <MTSInstance Name="sw_el_3_MESH_3" >
                        <Transform>
                                <Scale x="1" y="1" z="1" />
                                <Shear xy="0" yz="0" xz="-0" />
                                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                                <Position x="-3.0625" y="1.25" z="0" />
                        </Transform>
                        <MTSMaterial Name="Mat_2" ID="0">
                                <MTSTextureMap Type="Diffuse"
Name="text_rose3"

        Path="C:\Metastream\exp\rose_color\rose_red.jpg" Width="200" Height="300">
                                </MTSTextureMap>
                        <MTSColor Name="Diffuse" r="0.49804" g="0" b="0"
Opacity="0">
                        </MTSColor>
                        </MTSMaterial>
                </MTSInstance>
                <MTSInstance Name="sw_el_3_MESH_4" >
                        <Transform>
                                <Scale x="1" y="1" z="1" />
                                <Shear xy="0" yz="0" xz="-0" />
                                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                                <Position x="-3.0625" y="1.25" z="0" />
                        </Transform>
                        <MTSMaterial Name="Mat_3" ID="0">
                                <MTSTextureMap Type="Diffuse"
Name="text_rose4"

        Path="C:\Metastream\exp\rose_color\rose_red.jpg" Width="200" Height="300">
                                </MTSTextureMap>
                        <MTSColor Name="Diffuse" r="0.49804" g="0" b="0"
Opacity="0">
                        </MTSColor>
                        </MTSMaterial>
                </MTSInstance>
                <MTSInstance Name="sw_el_3_MESH_5" >
```

```
                        <Transform>
                                <Scale x="1" y="1" z="1" />
                                <Shear xy="0" yz="0" xz="-0" />
                                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                                <Position x="-2.875" y="0.8125" z="0" />
                        </Transform>
                        <MTSMaterial Name="Mat_4" ID="0">
                                <MTSTextureMap Type="Diffuse"
Name="text_rose5"

        Path="C:\Metastream\exp\rose_color\rose_red.jpg" Width="200" Height="300">
                                </MTSTextureMap>
                        <MTSColor Name="Diffuse" r="0.49804" g="0" b="0"
Opacity="0">
                        </MTSColor>
                        </MTSMaterial>
                </MTSInstance>
                <MTSInstance Name="sw_el_3_MESH_6" >
                        <Transform>
                                <Scale x="1" y="1" z="1" />
                                <Shear xy="0" yz="0" xz="-0" />
                                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                                <Position x="-3.0625" y="1.25" z="0" />
                        </Transform>
                        <MTSMaterial Name="Mat_5" ID="0">
                                <MTSTextureMap Type="Diffuse"
Name="text_h"

        Path="C:\Metastream\exp\rose_color\rose_red.jpg" Width="200" Height="300">
                                </MTSTextureMap>
                        <MTSColor Name="Diffuse" r="0.49804" g="0" b="0"
Opacity="0">
                        </MTSColor>
                        </MTSMaterial>
                </MTSInstance>
                <MTSInstance Name="sw_el_3_MESH_7" >
                        <Transform>
                                <Scale x="1" y="1" z="1" />
                                <Shear xy="0" yz="0" xz="-0" />
                                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                                <Position x="-3.0625" y="1.25" z="0" />
                        </Transform>
                </MTSInstance>
                <MTSInstance Name="sw_el_3_MESH_8" >
                        <Transform>
                                <Scale x="1" y="1" z="1" />
                                <Shear xy="0" yz="0" xz="-0" />
```

```
                                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                                <Position x="-3.0625" y="1.25" z="0" />
                        </Transform>
                </MTSInstance>
                <MTSInstance Name="sw_el_3_MESH_9" >
                        <Transform>
                                <Scale x="1" y="1" z="1" />
                                <Shear xy="0" yz="0" xz="-0" />
                                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                                <Position x="-3.0625" y="1.25" z="0" />
                        </Transform>
                </MTSInstance>
                <MTSInstance Name="sw_el_3_MESH_10" >
                        <Transform>
                                <Scale x="1" y="1" z="1" />
                                <Shear xy="0" yz="0" xz="-0" />
                                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                                <Position x="-3.0625" y="1.25" z="0" />
                        </Transform>
                        <MTSMaterial Name="Mat_6" ID="0">
                                <MTSTextureMap Type="Diffuse"
Name="text_b"

        Path="C:\Metastream\exp\rose_color\rose_red.jpg" Width="200" Height="300">
                                </MTSTextureMap>
                        <MTSColor Name="Diffuse" r="0.49804" g="0" b="0"
Opacity="0">
                        </MTSColor>
                        </MTSMaterial>
                </MTSInstance>
                <MTSInstance Name="sw_el_3_MESH_11" >
                        <Transform>
                                <Scale x="1" y="1" z="1" />
                                <Shear xy="0" yz="0" xz="-0" />
                                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                                <Position x="-3.0625" y="1.25" z="0" />
                        </Transform>
                </MTSInstance>
                <MTSInstance Name="sw_el_3_MESH_12" >
                        <Transform>
                                <Scale x="1" y="1" z="1" />
                                <Shear xy="0" yz="0" xz="-0" />
                                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                                <Position x="-3.0625" y="1.25" z="0" />
                        </Transform>
                </MTSInstance>
```

```
<MTSInstance Name="sw_el_3_MESH_13" >
        <Transform>
                <Scale x="1" y="1" z="1" />
                <Shear xy="0" yz="0" xz="-0" />
                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                <Position x="-3.0625" y="1.25" z="0" />
        </Transform>
</MTSInstance>
<MTSInstance Name="sw_el_3_MESH_14" >
        <Transform>
                <Scale x="1" y="1" z="1" />
                <Shear xy="0" yz="0" xz="-0" />
                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                <Position x="-3.0625" y="1.25" z="0" />
        </Transform>
</MTSInstance>
<MTSInstance Name="sw_el_3_MESH_15" >
        <Transform>
                <Scale x="1" y="1" z="1" />
                <Shear xy="0" yz="0" xz="-0" />
                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                <Position x="-3.0625" y="1.25" z="0" />
        </Transform>
</MTSInstance>
<MTSInstance Name="sw_el_3_MESH_16" >
        <Transform>
                <Scale x="1" y="1" z="1" />
                <Shear xy="0" yz="0" xz="-0" />
                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                <Position x="-3.0625" y="1.25" z="0" />
        </Transform>
</MTSInstance>
<MTSInstance Name="sw_el_3_MESH_17" >
        <Transform>
                <Scale x="1" y="1" z="1" />
                <Shear xy="0" yz="0" xz="-0" />
                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
                <Position x="-3.0625" y="1.25" z="0" />
        </Transform>
</MTSInstance>
<MTSInstance Name="sw_el_3_MESH_18" >
        <Transform>
                <Scale x="1" y="1" z="1" />
                <Shear xy="0" yz="0" xz="-0" />
                <Rotate x="-94.94799" y="8.46704" z="-
172.53311" />
```

```
                                        <Position x="-3.0625" y="1.25" z="0" />
                                </Transform>
                        </MTSInstance>
                </MTSInstance>
                <MTSInstance Name="Simple_0" >
                        <Transform>
                                <Scale x="7.72622" y="4.82905" z="7.74871" />
                                <Shear xy="0" yz="-0" xz="-0" />
                                <Rotate x="7.0302" y="-1.10005" z="10.26211" />
                                <Position x="0.05667" y="-3.99971" z="-0.98282" />
                        </Transform>
                        <MTSGeometry Name="Cylinder" Type="MTSCylinder"
Resolution="10" Segments="10" StartDegrees="0" EndDegrees="360" />
                        <MTSMaterial Name="Mat_7" ID="0">
                        <MTSTextureMap Type="Diffuse" Name="text_side"

        Path="C:\Metastream\exp\cake_material\mat_candycane.jpg" Width="200"
Height="300">
                        </MTSTextureMap>
                        <MTSColor Name="Diffuse" r="0.49804" g="0" b="0"
Opacity="0">
                        </MTSColor>
                        </MTSMaterial>
                        <MTSMaterial Name="Mat_8" ID="1">
                        <MTSTextureMap Type="Diffuse" Name="text_top"
                                Path="C:\Metastream\exp\cake_material\mat_top.jpg"
Width="200" Height="300">
                        </MTSTextureMap>
                        <MTSColor Name="Diffuse" r="0.49804" g="0" b="0"
Opacity="0">
                        </MTSColor>
                        </MTSMaterial>
                </MTSInstance>
        </MTSInstance>
        <MTSTimeElem Type="MTSStream" Name="sw_el_3" On="1"
Path="sw_el_3.mts" >
                <Target Name="MTSInstance.sw_el_1" />
        </MTSTimeElem>

        <!-- Preloading materials for animation -->
        <!-- Preloading materials for rose1 animation -->

        <MTSTextureMap Name="rose_red"
Path="C:\Metastream\exp\rose_color\rose_red.jpg"/>
        <MTSTimeElem Name="txtani_rose1_red" Type="Texture" On="0">
                <Target Name="text_rose1" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_red
                </Timeline>
        </MTSTimeElem>
```

```
        <MTSTextureMap Name="rose_pink"
Path="C:\Metastream\exp\rose_color\rose_pink.jpg"/>
        <MTSTimeElem Name="txtani_rose1_pink" Type="Texture" On="0">
                <Target Name="text_rose1" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_pink
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_yellow"
Path="C:\Metastream\exp\rose_color\rose_yellow.jpg"/>
        <MTSTimeElem Name="txtani_rose1_yellow" Type="Texture" On="0">
                <Target Name="text_rose1" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_yellow
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_white"
Path="C:\Metastream\exp\rose_color\rose_white.jpg"/>
        <MTSTimeElem Name="txtani_rose1_white" Type="Texture" On="0">
                <Target Name="text_rose1" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_white
                </Timeline>
        </MTSTimeElem>

        <!-- Preloading materials for rose2 animation -->

        <MTSTextureMap Name="rose_red"
Path="C:\Metastream\exp\rose_color\rose_red.jpg"/>
        <MTSTimeElem Name="txtani_rose2_red" Type="Texture" On="0">
                <Target Name="text_rose2" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_red
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_pink"
Path="C:\Metastream\exp\rose_color\rose_pink.jpg"/>
        <MTSTimeElem Name="txtani_rose2_pink" Type="Texture" On="0">
                <Target Name="text_rose2" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_pink
                </Timeline>
```

```
        </MTSTimeElem>

        <MTSTextureMap Name="rose_yellow"
Path="C:\Metastream\exp\rose_color\rose_yellow.jpg"/>
        <MTSTimeElem Name="txtani_rose2_yellow" Type="Texture" On="0">
                <Target Name="text_rose2" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_yellow
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_white"
Path="C:\Metastream\exp\rose_color\rose_white.jpg"/>
        <MTSTimeElem Name="txtani_rose2_white" Type="Texture" On="0">
                <Target Name="text_rose2" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_white
                </Timeline>
        </MTSTimeElem>

        <!-- Preloading materials for rose3 animation -->

        <MTSTextureMap Name="rose_red"
Path="C:\Metastream\exp\rose_color\rose_red.jpg"/>
        <MTSTimeElem Name="txtani_rose3_red" Type="Texture" On="0">
                <Target Name="text_rose3" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_red
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_pink"
Path="C:\Metastream\exp\rose_color\rose_pink.jpg"/>
        <MTSTimeElem Name="txtani_rose3_pink" Type="Texture" On="0">
                <Target Name="text_rose3" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_pink
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_yellow"
Path="C:\Metastream\exp\rose_color\rose_yellow.jpg"/>
        <MTSTimeElem Name="txtani_rose3_yellow" Type="Texture" On="0">
                <Target Name="text_rose3" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_yellow
```

```
            </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_white"
Path="C:\Metastream\exp\rose_color\rose_white.jpg"/>
        <MTSTimeElem Name="txtani_rose3_white" Type="Texture" On="0">
            <Target Name="text_rose3" Property="pixl"/>
            <Timeline>
                0 *
                1 MTSTexture.rose_white
            </Timeline>
        </MTSTimeElem>


        <!-- Preloading materials for rose4 animation -->

        <MTSTextureMap Name="rose_red"
Path="C:\Metastream\exp\rose_color\rose_red.jpg"/>
        <MTSTimeElem Name="txtani_rose4_red" Type="Texture" On="0">
            <Target Name="text_rose4" Property="pixl"/>
            <Timeline>
                0 *
                1 MTSTexture.rose_red
            </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_pink"
Path="C:\Metastream\exp\rose_color\rose_pink.jpg"/>
        <MTSTimeElem Name="txtani_rose4_pink" Type="Texture" On="0">
            <Target Name="text_rose4" Property="pixl"/>
            <Timeline>
                0 *
                1 MTSTexture.rose_pink
            </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_yellow"
Path="C:\Metastream\exp\rose_color\rose_yellow.jpg"/>
        <MTSTimeElem Name="txtani_rose4_yellow" Type="Texture" On="0">
            <Target Name="text_rose4" Property="pixl"/>
            <Timeline>
                0 *
                1 MTSTexture.rose_yellow
            </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_white"
Path="C:\Metastream\exp\rose_color\rose_white.jpg"/>
        <MTSTimeElem Name="txtani_rose4_white" Type="Texture" On="0">
            <Target Name="text_rose4" Property="pixl"/>
            <Timeline>
```

```
                0 *
                1 MTSTexture.rose_white
            </Timeline>
        </MTSTimeElem>

        <!-- Preloading materials for rose5 animation -->

        <MTSTextureMap Name="rose_red"
Path="C:\Metastream\exp\rose_color\rose_red.jpg"/>
        <MTSTimeElem Name="txtani_rose5_red" Type="Texture" On="0">
            <Target Name="text_rose5" Property="pixl"/>
            <Timeline>
                0 *
                1 MTSTexture.rose_red
            </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_pink"
Path="C:\Metastream\exp\rose_color\rose_pink.jpg"/>
        <MTSTimeElem Name="txtani_rose5_pink" Type="Texture" On="0">
            <Target Name="text_rose5" Property="pixl"/>
            <Timeline>
                0 *
                1 MTSTexture.rose_pink
            </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_yellow"
Path="C:\Metastream\exp\rose_color\rose_yellow.jpg"/>
        <MTSTimeElem Name="txtani_rose5_yellow" Type="Texture" On="0">
            <Target Name="text_rose5" Property="pixl"/>
            <Timeline>
                0 *
                1 MTSTexture.rose_yellow
            </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_white"
Path="C:\Metastream\exp\rose_color\rose_white.jpg"/>
        <MTSTimeElem Name="txtani_rose5_white" Type="Texture" On="0">
            <Target Name="text_rose5" Property="pixl"/>
            <Timeline>
                0 *
                1 MTSTexture.rose_white
            </Timeline>
        </MTSTimeElem>

        <!-- Preloading materials for H animation -->

        <MTSTextureMap Name="rose_red"
Path="C:\Metastream\exp\rose_color\rose_red.jpg"/>
```

```
        <MTSTimeElem Name="txtani_h_red" Type="Texture" On="0">
                <Target Name="text_h" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_red
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_pink"
Path="C:\Metastream\exp\rose_color\rose_pink.jpg"/>
        <MTSTimeElem Name="txtani_h_pink" Type="Texture" On="0">
                <Target Name="text_h" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_pink
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_yellow"
Path="C:\Metastream\exp\rose_color\rose_yellow.jpg"/>
        <MTSTimeElem Name="txtani_h_yellow" Type="Texture" On="0">
                <Target Name="text_h" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_yellow
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_white"
Path="C:\Metastream\exp\rose_color\rose_white.jpg"/>
        <MTSTimeElem Name="txtani_h_white" Type="Texture" On="0">
                <Target Name="text_h" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_white
                </Timeline>
        </MTSTimeElem>

        <!-- Preloading materials for B animation -->

        <MTSTextureMap Name="rose_red"
Path="C:\Metastream\exp\rose_color\rose_red.jpg"/>
        <MTSTimeElem Name="txtani_b_red" Type="Texture" On="0">
                <Target Name="text_b" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_red
                </Timeline>
        </MTSTimeElem>
```

```
        <MTSTextureMap Name="rose_pink"
Path="C:\Metastream\exp\rose_color\rose_pink.jpg"/>
        <MTSTimeElem Name="txtani_b_pink" Type="Texture" On="0">
                <Target Name="text_b" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_pink
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_yellow"
Path="C:\Metastream\exp\rose_color\rose_yellow.jpg"/>
        <MTSTimeElem Name="txtani_b_yellow" Type="Texture" On="0">
                <Target Name="text_b" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_yellow
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="rose_white"
Path="C:\Metastream\exp\rose_color\rose_white.jpg"/>
        <MTSTimeElem Name="txtani_b_white" Type="Texture" On="0">
                <Target Name="text_b" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.rose_white
                </Timeline>
        </MTSTimeElem>

        <!-- Preloading materials for cake side animation -->

        <MTSTextureMap Name="mat_candycane"
Path="C:\Metastream\exp\cake_material\mat_candycane.jpg"/>
        <MTSTimeElem Name="txtani_candycane" Type="Texture" On="0">
                <Target Name="text_side" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_candycane
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="mat_chococuststripe"
Path="C:\Metastream\exp\cake_material\mat_chococuststripe.jpg"/>
        <MTSTimeElem Name="txtani_chococustripe" Type="Texture" On="0">
                <Target Name="text_side" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_chococuststripe
                </Timeline>
        </MTSTimeElem>
```

```
        <MTSTextureMap Name="mat_chocostraw"
Path="C:\Metastream\exp\cake_material\mat_chocostraw.jpg"/>
        <MTSTimeElem Name="txtani_chocostraw" Type="Texture" On="0">
                <Target Name="text_side" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_chocostraw
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="mat_choppednuts"
Path="C:\Metastream\exp\cake_material\mat_choppednuts.jpg"/>
        <MTSTimeElem Name="txtani_choppednuts" Type="Texture" On="0">
                <Target Name="text_side" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_choppednuts
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="mat_custstrawkiwi"
Path="C:\Metastream\exp\cake_material\mat_custstrawkiwi.jpg"/>
        <MTSTimeElem Name="txtani_custstrawkiwi" Type="Texture" On="0">
                <Target Name="text_side" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_custstrawkiwi
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="mat_cutiefrutie"
Path="C:\Metastream\exp\cake_material\mat_cutiefrutie.jpg"/>
        <MTSTimeElem Name="txtani_cutiefrutie" Type="Texture" On="0">
                <Target Name="text_side" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_cutiefrutie
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="mat_justchoc"
Path="C:\Metastream\exp\cake_material\mat_justchoc.jpg"/>
        <MTSTimeElem Name="txtani_justchoc" Type="Texture" On="0">
                <Target Name="text_side" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_justchoc
                </Timeline>
        </MTSTimeElem>
```

```
        <MTSTextureMap Name="mat_polkadot"
Path="C:\Metastream\exp\cake_material\mat_polkadot.jpg"/>
        <MTSTimeElem Name="txtani_polkadot" Type="Texture" On="0">
                <Target Name="text_side" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_polkadot
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="mat_raspspecial"
Path="C:\Metastream\exp\cake_material\mat_raspspecial.jpg"/>
        <MTSTimeElem Name="txtani_raspspecial" Type="Texture" On="0">
                <Target Name="text_side" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_raspspecial
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="mat_whipcream"
Path="C:\Metastream\exp\cake_material\mat_whipcream.jpg"/>
        <MTSTimeElem Name="txtani_whipcream" Type="Texture" On="0">
                <Target Name="text_side" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_whipcream
                </Timeline>
        </MTSTimeElem>

        <!-- Preloading materials for cake top animation -->

        <MTSTextureMap Name="mat_top"
Path="C:\Metastream\exp\cake_material\mat_top.jpg"/>
        <MTSTimeElem Name="txtani_top" Type="Texture" On="0">
                <Target Name="text_top" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_top
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="mat_top_whipcream"
Path="C:\Metastream\exp\cake_material\mat_top_whipcream.jpg"/>
        <MTSTimeElem Name="txtani_top_whipcream" Type="Texture" On="0">
                <Target Name="text_top" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_top_whipcream
                </Timeline>
        </MTSTimeElem>
```

```
        <MTSTextureMap Name="mat_top_choco"
Path="C:\Metastream\exp\cake_material\mat_top_choco.jpg"/>
        <MTSTimeElem Name="txtani_top_choco" Type="Texture" On="0">
                <Target Name="text_top" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_top_choco
                </Timeline>
        </MTSTimeElem>

        <MTSTextureMap Name="mat_top_straw"
Path="C:\Metastream\exp\cake_material\mat_top_straw.jpg"/>
        <MTSTimeElem Name="txtani_top_straw" Type="Texture" On="0">
                <Target Name="text_top" Property="pixl"/>
                <Timeline>
                        0 *
                        1 MTSTexture.mat_top_straw
                </Timeline>
        </MTSTimeElem>

</MTSScene>
```