NAME : **KHAIRUL ANWAR BIN ZAINAL**

MATRICS NO : **WEK010413**

MAJOR : **COMPUTER SYSTEMS & TECHNOLOGY**

SUPERVISOR : **MS. RAFIDAH MOHD. NOR**

MODERATOR: **MRS. NORNAZLITA HUSSIN**

TITLE : **SOCCER365 MMS APPLICATION**

COURSE : **LATIHAN ILMIAH AKHIR II (WXES3182)**

ABSTRACT

This project documentation describes in detail the information regarding the project Soccer365 on MMS. The first part of this thesis, WXES 3181, is primarily divided into four chapters, that is, Introduction, Literature Review, Methodology and System Design The second part, WXES 3182, consists of three chapters; System Implementation, System Testing and System Evaluation.

Chapter 1, Introduction, will give the reader an overview of what this thesis is all about. Also, the objectives for developing this project, the targeted user group, project limitations, the scope covered by this project and an estimated schedule to finish the project are clearly stated.

In Chapter 2, Literature Review, a thorough review on similar existing systems is carried out in order to obtain a better understanding with regards to the proposed project. It provides developers with some knowledge and insights into the strengths and limitations of several development tools that is available in the market. The software and scripting languages best used to develop the project is also researched.

In Chapter 3, Methodology, will be discussed the right type of methodology used for completing every activity in the System Development Life Cycle and also the tools to be used for the project.

In Chapter 4, System Design, the development of the proposed project will be discussed and ascertained. This includes system functionality design and module functionality design. It shows the modules involved in the system and the data flow when information is passed.

In Chapter 5, System Implementation, the system is implemented into a workable system based on the system requirements and design specifications.

In Chapter 6, System Testing is broken down into three distinctive phases; unit testing, integration testing and system testing. Some of the methods such as function testing, performance testing, acceptance testing and installation testing were used to facilitate this operation.

Finally, in Chapter 7, System Evaluation, the difficulties and problems encountered during the development process from the beginning until the end of the project, the solutions to the problems, the strengths and the limitations of the system being developed, and the future enhancement of the system are thoroughly discussed.

ACKNOWLEDGEMENTS

Praise be to Allah, the Almighty, who by His infinite mercy and guidance has enabled me to complete this documentation for my WXES 3182 thesis, a compulsory project for every student of the Faculty of Computer Science and Information Technology (FCSIT).

Firstly, I would like to take this opportunistic moment to convey my deepest appreciation and reserve many thanks to my dearest supervisor, Ms. Rafidah Mohd. Noor for her invaluable advice, assistance and guidance in helping me complete this documentation.

Secondly, I would also like to reserve many thanks to my moderator, Mrs. Nazlita Hussin for her suggestions and comments to better my thesis.

Finally, I would like to express my love and gratitude to my dearest parents and beloved friends, especially Aidill Bin Johari, Mohd Yohan Bin Ibrahim, Nazlin Irham Ibrahim and Amin Razak, for providing me with everlasting support and encouragement whilst completing my thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.0 INTRODUCTION

Multimedia Messaging Service (MMS) - sometimes called Multimedia Messaging System - is a communications technology developed by 3GPP (Third Generation Partnership Project) that allows users to exchange multimedia communications between capable mobile phones and other devices. An extension to the Short Message Service (SMS) protocol, MMS defines a way to send and receive, almost instantaneously, wireless messages that include images, audio, and video clips in addition to text. When the technology has been fully developed, it will support the transmission of streaming video. A common current application of MMS messaging is picture messaging (the use of camera phones to take photos for immediate delivery to a mobile recipient). Other

possibilities include animations and graphic presentations of stock quotes, sports news, and weather reports.

The market potential for MMS was immediately evident when the first commercial multimedia application debuted, in 2001. In the first 18 months after its launch in Japan and South Korea, the Sha-Mail photo messaging service from J-Phone attracted an estimated 4 million subscribers, or nearly 33 percent of J-Phone's total wireless subscriber base. That kind of consumer demand is especially impressive in light of the relatively low photo quality in Sha-Mail's early stages and the fact that the proprietary service was only for J-Phone subscribers.

Clearly there is a significant market opportunity for multimedia messaging and telecom equipment makers and wireless providers are now battling for position in an MMS market that analysts have estimated may generate nearly $70 billion annually in content and service revenue by 2007.

Telecom equipment makers and wireless providers are now battling for position in an MMS market that analysts estimate may generate nearly $70 billion annually in content and service revenue by 2007.

Welcome to the era of the multimedia messaging service (MMS). MMS enables users to combine photos, animation, video, and audio to create powerful and engaging messages. A complete departure from previous messaging formats, MMS is based on the Internet's IP technology, which means that it can support popular content formats such as MP3 for music and JPEG and GIF for photos.

New cell phones, PDAs, and other handheld devices that can send and receive colorful, kinetic multimedia messages are rapidly becoming the standard in the wireless world, and simple text messaging (which has never allowed more than 160 characters, let alone a thousand words) is already looking like a quaint relic of the past.

Although designed to take full advantage of the emerging third-generation, or 3G, standards for wireless networks, MMS also works on 2.5G standards and is already helping drive up usage and capacity on 2.5G networks. Add the faster processing speeds and increased bandwidth of 3G networks and MMS promises to deliver a wide array of exciting applications, including:- Photo greeting cards; Audio/video messaging; Video-based entertainment; Interactive games; In-depth news and financial information; Personalized advertising; Dating services; Citizen-assisted "photo 911" emergency response; and Electronic voting.

Therefore, the main reason behind the initiative to develop an MMS application solely on soccer, called **Soccer365**, is to provide soccer fans and enthusiasts alike, with a useful service that enables them to download images and wallpaper with regards to soccer, a popular sport witnessed by millions of people around the globe. This particular application will enable users to view and download these interactive contents with extreme ease by the touch of a MMS-enabled mobile phone.

1.1 OBJECTIVES

- To develop an application depicting soccer-related contents for soccer fans and enthusiasts.
- To develop an application that is user-friendly.
- To able to send and receive MMS message(s), almost instantaneously, between two engaging parties.

1.2 SCOPE

The boundaries which are set upon this particular application consist of two modules; (i) the user module and (ii) the administrator module.

For the user module, users are able to view and download the interactive content that is available.

Meanwhile, for the administrator module, the administrator is able to add the user, delete the user and update the contents of the database.

1.3 TARGETED USERS

The targeted users for this particular application are soccer fans and enthusiasts. This application will allow these groups of people to connect and retrieve data as long as they are within a wireless network.

1.4 PROJECT LIMITATIONS

Developing this particular application will prove to be time consuming as many issues are needed to be addressed and resolved. This will have to be done at an early stage before any development stages are to be initiated. Such issues are enlisted as follows:-

- Types of protocol to be supported, such as WAP and GPRS.
- Speed of data transmission rate.
- Types of devices need to be supported, such mobile phone or Personal Digital Assistant (PDA).
- Limitation of WAP-enabled devices such as small display, high latency, limited connection stability, limited memory and CPU, limited battery power and limited input facilities.

1.5 PROJECT SCHEDULE

A project schedule is planned to manage the time that needs to acomplish this project and to archive the objectives.

	Task Name	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb
1	System Analysis									
2	System Modeling									
3	System Design									
4	Implementation									
5	System Maintenance									
6	System Delivery									
7	Documentation									

Table 1.1: Gantt chart explaining project runtime

Description of each task:

1. **System Analysis:**

   The system is analyzed and the basic requirements are captured. Those requirements taken into account are the operating environment, architecture and resources available for development.

2. **System Modeling :**

This task includes the capturing of functional and non-functional requirements of the system. From these findings, the specifications of the project can be determined and therefore could be used to initiate the start of system development.

3. **System Design :**

The system design phase is concerned with synthesizing or putting all the parts together into the workable system. The design will include the functions (processes) that the system must perform using database design, data flow diagram (DFD) and object diagram.

4. **Implementation :**

During this stage, the software design is realized as a set of programs or program units.

5. **System Maintenance :**

It refers to the general process of changing the system after it has been delivered. The changes may be simple changes to correct coding errors, more extensive changes to correct design errors or significant enhancements to correct specification errors or accommodate new requirements.

6. **System Delivery :**

This is the final phase of the development process. The complete system is delivered to the user.

7. **Documentation :**

The information that describes the delivered system in detail. It consists of the product technical manuals and online information (including online versions of the technical manuals and help facility descriptions)

CHAPTER 2

LITERATURE REVIEW

2.0 INTRODUCTION

Literature review can be described as a systematic review of the published work on the topic of your study, which in this particular case, the application that is being developed. It is crucial to provide application developers with insights and understanding of the process, as a whole, before really starting to develop an application. It also provides enlightenment on how our application should be conveyed.

As a developer of the Soccer365 MMS application, the purpose of doing literature review is also to sufficiently equip developers with some knowledge of the strengths and limitation of several development tools available in the market. This is to enable the developers to choose the right tools to develop the system or application.

Most importantly, a literature review provides reviews on existing systems and these systems are further analyzed and act as reference points for the project. Some important features that need to be further improved are documented.

2.1 WHAT IS GPRS?

MMS uses General Packet Radio Services (GPRS) as the primary bearer of the service. In other words, GPRS is the mode of delivery for MMS.

General Packet Radio Services (GPRS) is a packet-based wireless communication service that promises data rates from 56 up to 114 Kbps and continuous connection to the Internet for mobile phone and computer users. The higher data rates will allow users to take part in video conferences and interact with multimedia Web sites and similar applications using mobile handheld devices as well as notebook computers. GPRS is based on Global System for Mobile (GSM) communication and will complement existing services such circuit-switched cellular phone connections and the Short Message Service (SMS).

In theory, GPRS packet-based service should cost users less than circuit-switched services since communication channels are being used on a shared-use, as-packets-are-needed basis rather than dedicated only to one user at a time. It should also be easier to make applications available to mobile users because the faster data rate means that middleware currently needed to adapt applications to the slower speed of wireless systems will no longer be needed. As GPRS becomes available, mobile users of a virtual private network (VPN) will be able to access the private network continuously rather than through a dial-up connection.

GPRS will also complement Bluetooth, a standard for replacing wired connections between devices with wireless radio connections. In addition to the Internet Protocol (IP), GPRS supports X.25, a packet-based protocol that is used mainly in Europe. GPRS is an evolutionary step toward Enhanced Data GSM Environment (EDGE) and Universal Mobile Telephone Service (UMTS).

2.2 GENERAL CONCEPT OF MMS

Sending an MMS message is like sending an SMS message

Most people are familiar with at least the basics of sending an SMS message. The following is a picture of how an SMS message is delivered. There are several bits missing, but the basic concept is:

- The message originator addresses the short message to the receiver.
- The phone contains information about SMSC (SMS Center), and the message is sent there.
- SMSC attempts to forward the message to the receiver.

If for some reason the receiver is unreachable, the SMSC stores the message for a time, and if possible, delivers the message later. If the message cannot be delivered within a certain time frame, it is eventually discarded.

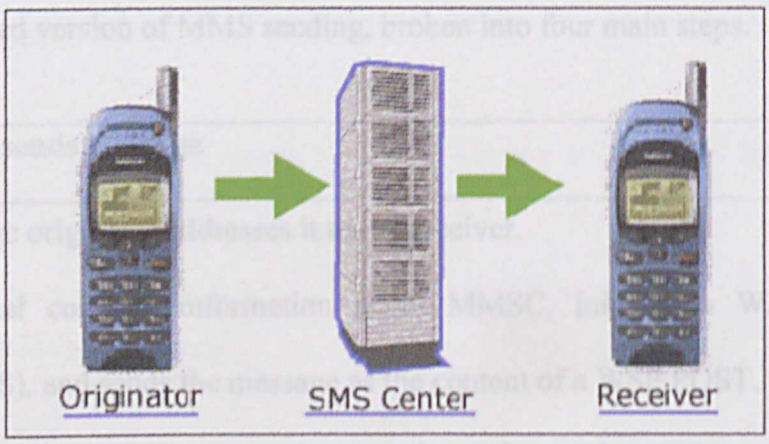

Figure 2.1: Sending an SMS message

The basic concept of sending an MMS message is exactly the same.

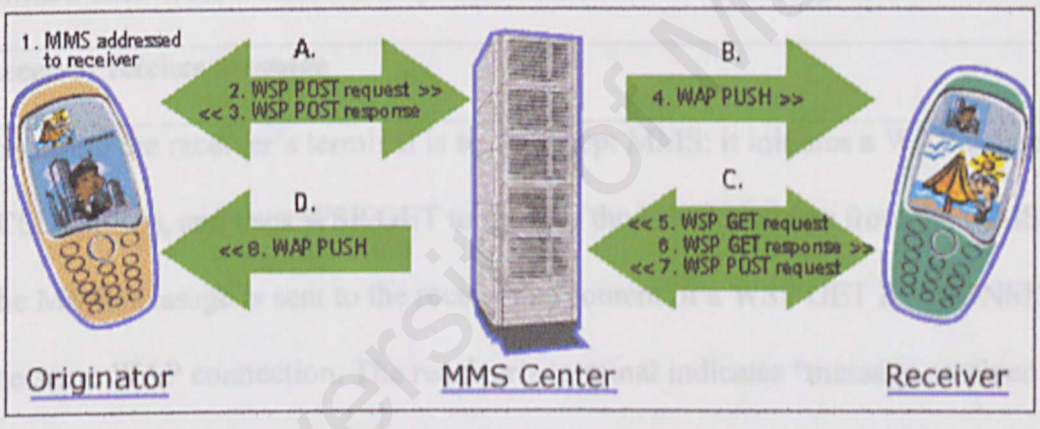

Figure 2.2: A more detailed rendering of sending an MMS message

Here is a detailed version of MMS sending, broken into four main steps.

A: Originator sends message
1. The message originator addresses it to the receiver. 2. The terminal contains information about MMSC, initiates a WAP connection (CSD/GPRS), and sends the message as the content of a WSP POST. 3. MMSC accepts the message and responds to the originator over the same WAP connection. The originator's terminal indicates "message sent."
B: MMSC informs receiver
4. MMSC uses WAP Push to attempt to send an indication message to the receiver.
C: Receiver fetches message
5. Assuming the receiver's terminal is set to accept MMS; it initiates a WAP connection (CSD/GPRS), and uses WSP GET to retrieve the MMS message from the MMSC. 6. The MMS message is sent to the receiver as content of a WSP GET RESPONSE over the same WAP connection. The receiver's terminal indicates "message received." 7. The receiver's terminal acknowledges receipt with a WSP POST message, still over the same WAP connection.
D: MMSC informs originator of delivery
8. MMSC uses WAP Push to indicate to the originator that the message was delivered. The originator's terminal indicates "message delivered."

Table 2.1: MMS Sending

2.3 POSSIBLE ELEMENTS IN AN MMS

Next we depict some other elements that may become involved in MMS issues and explain how they fit into the scheme of things.

2.3.1 E-mail Server/Gateway

A generic class of servers that nominally hosts email services that operates using the SMTP, POP and/or IMAP protocols.

2.3.2 Legacy Support

In order to support (in a way) MMS message delivery to legacy terminals, there must be some sort of legacy-messaging server. A Terminal Gateway (TGW) provides this type of service by storing MMS message content in its own local storage. It then sends an SMS message to the receiver, informing the user of a Web address where the content can be viewed via a Web browser. The Terminal Gateway (TGW) also provides users with a "shoebox" to store images in. These images can be accessed over the Internet and used to create new MMS messages, which can be sent via the TGW.

2.3.3 Subscriber Database

A database of subscriber profiles can help when deciding what type of content to deliver. For example, if A sends B an MMS message, the MMSC can determine via such a database that B does not have an MMS-capable terminal, and forward the content directly to (for example) the TGW. The TGW then takes care of getting the message to B, as described above.

2.3.4 Content Converter

In another situation, A could have sent B an image in a format that is not supported by B's terminal. The MMSC determines this using the subscriber database from above, then routes the message to a content-converting application. After conversion, the new message is sent forward.

2.3.5 Voice Mail

The Nokia Multimedia Voice Gateway interfaces between the MMSC and a Voice-mail application. Instead of receiving a text message indicating you have voice messages waiting, the voice messages can be encapsulated as MMS messages and sent directly to the phone.

2.3.6 "Foreign" MMSC

Another issue is when A and B do not belong to the same operator network. In this case the MMSC for A's network forwards the MMS message to the MMSC in B's network. B's MMSC then takes care of notifying B of an incoming message, and things go very much the same as described in Table 3: Steps in sending an MMS message. If a delivery report is to be sent to A, it is first sent from B's MMSC to A's MMSC.

2.3.7 Other

What about SMS services that allowed you to fetch, for example, logos using keywords – can you do a similar thing to fetch color images? Yes, you can. The Nokia Messaging Gateway (previously known as Nokia Artuse Messaging Platform, or NAMP), which is

Nokia's solution for this type of service, will soon have the capability to interact with an MMSC, allowing it to be used for multimedia content fetching as well. In this type of case, the user sends an SMS message to request the image. That request is routed to the Messaging Gateway, which retrieves the image and delivers it via the MMSC as an MMS message. It would also be possible, of course, for the request to be sent as an MMS message, but this would probably not be as cost-efficient (depends on the operator's billing strategies).

2.3.8 The developer

The mail server, TGW, user database, and content-converter are all examples of external applications (from the MMSC point of view). Your place as a developer will most likely be at the external application end of things. Some developers may be at the client end, however, using client MMS handlers in order to send/receive MMS messages from/to their specialized applications.

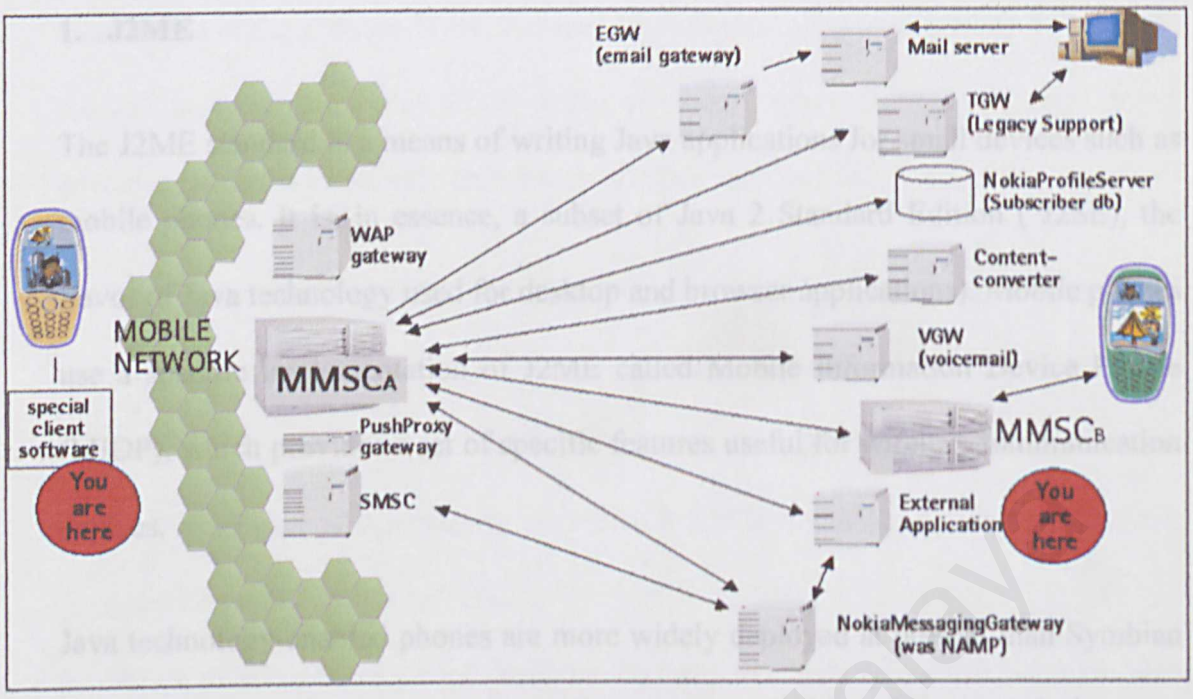

Figure 2.3: Possible elements in a Multimedia Messaging System

2.4 HARDWARE & SOFTWARE

Enlisted below are some of the hardware and software used to develop an MMS application.

Hardware:

Any MMS-enabled mobile phone.

Software:

1. J2ME
2. Symbian OS
3. WML & WML Script
4. SMIL

1. J2ME

The J2ME standard is a means of writing Java applications for small devices such as mobile phones. It is, in essence, a subset of Java 2 Standard Edition (J2SE), the flavor of Java technology used for desktop and browser applications). Mobile phones use a specific implementation of J2ME called Mobile Information Device Profile (MIDP), which provides a set of specific features useful for wireless communication devices.

Java technology-enabled phones are more widely deployed at present than Symbian OS phones, providing a larger potential user base. All things being equal, the more potential users one has, the greater the potential revenues.

Most Java technology-enabled phones allow over-the-air (OTA) provisioning, meaning that a user can simply select an application from a list on his/her phone, download it, and install it. This makes application delivery easy. Typically, the list of available applications is provided by the network operator, the phone manufacturer, or a third-party wireless portal such as Club Nokia. These intermediaries have billing systems in place, and can bill the user for application purchase, sharing this revenue with the application provider. In other words, a clear business model for achieving revenues from J2ME applications is in place.

Because J2ME is a subset of J2SE, developers with prior Java experience find it relatively easy to adapt to J2ME development. J2ME allows applications that do not hit the air network, such as solo-play games, to run on a handset—something not possible with earlier mobile phones, which were restricted to exchanging SMS

messages and the passive WAP browser environment. For applications that do use the air network to communicate with a remote server, J2ME has two strong advantages over prior mobile phone technologies such as SMS and WAP:

1. With prior technologies, the handset was simply a passive displayer of information, meaning that all processing had to be performed on the server side, with results returned to the users. With J2ME, some processing can be performed on the handset side, reducing air network traffic and the processing hit on the application provider's servers.
2. While J2ME provides high-level UI features like lists and checkboxes, it also allows developers to display whatever they wish on the screen and assign their own meanings to handset keys, permitting developers to create better, richer, and easier-to-use interfaces for their applications.

Finally, Symbian OS phones support J2ME; thus, J2ME applications can run on phones using Nokia OS as well as Symbian OS phones such as Series 60 devices.

J2ME is not, however, ideal for all applications, and for several reasons.

Most Java technology-enabled phones have tight limits on the amount of memory available for MIDlets (the name for Java applications written for the J2ME/MIDP platform). There is a limit both on total memory available for storage of all MIDlets on the phone, as well as a limit on the maximum size of individual MIDlets. In some cases, the latter limit can be as low as 30 k (although 64 k is more common). Additionally, the operator's network often imposes a limit on the size of MIDlet

downloads. Java technology is an interpreted language. Native Symbian OS applications will, all things being equal, run faster and perform better than MIDlets.

The J2ME specification provides no analog to desktop Java's Native Interface (JNI), meaning that developers cannot rely on the ability to access other features of Java technology-enabled phones, such as WAP and SMS, from within a MIDlet. In some cases, developers can do so through a manufacturer's extensions to J2ME—for example, on some Nokia phones, the Nokia SMS API allows MIDlets to send and receive SMS messages—but MIDlets that take advantage of such features are not guaranteed to run on all J2ME phones.

Though J2ME is a well-documented open standard, J2ME developers face a number of cross-platform issues, including:

- Mobile phone screen sizes vary, and J2ME has no facility to automatically scale objects (or drawing area) to the screen size. While J2ME allows developers to query a phone to determine its screen size, taking full advantage of a phone's characteristics often makes it necessary to provide slightly different versions of a MIDlet for different handsets.
- Most manufacturers, including Nokia, provide useful extensions to J2ME, such as the Nokia UI and SMS APIs; it's often advisable to take advantage of the features they provide, to ensure the best user experience on those handsets. This means that, for best results, it may be necessary to code slightly different versions of the MIDlets for different target platforms.

2. Symbian OS

An operating system for powerful smart phones that runs applications written in the C++ language and compiled to machine code. Smart phones are less widely deployed than Java-enabled phones, but more fully featured and capable of running more sophisticated applications, and will experience further market penetration over time. As with J2ME, there are competing "smart phone" operating systems in deployment, but phone manufacturers representing more than 70% of the market (including Nokia, Sony-Ericsson, Siemens, and Panasonic) have committed to the use of Symbian OS for their smart phones, making it more widely supported than other operating systems. Additionally, Symbian OS is an open standard, available for license by any hardware manufacturer, and is specifically designed for use with wireless communication devices, thus providing superior functionality. Symbian OS phones generally have several megabytes of memory available to Symbian OS applications, allowing developers to create far richer and more sophisticated applications than under J2ME. Symbian OS applications are typically coded in C++, with which many developers are familiar, making it fairly easy for experienced coders to develop applications for Symbian OS devices.

Under Symbian OS, an application can access all hardware and software features of a phone, including SMS, MMS, WAP, infrared ports, Bluetooth, and voice telephony features, as well as pre-installed applications like the phone book and calendar. This allows for many better-integrated, full-featured applications than are typically feasible with J2ME. (Note that third-party applications do have some limits as to the hardware they can access–e.g., there are some limitations on accessing the SIM card,

video memory, etc.) Because Symbian OS applications are compiled to native machine code, they typically run much faster than J2ME applications. Most Symbian OS phones support SyncML, an open standard that allows synchronization of phone books, calendars, email, etc., between PDAs, phones, desktop computers, and other computing devices. Most Symbian OS phones include an IrDA (infrared) port, and many support Bluetooth, allowing them to establish local-area wireless connections with other devices, at far lower latencies than are prevalent over a cellular air network. Because of the greater processing power and memory available on Symbian OS phones, as well as the better performance of native applications, processor-intensive applications that are difficult or impossible to implement in J2ME—such as 3D games—can be created for Symbian OS phones.

There is a smaller installed base of Symbian OS than J2ME phones at present, meaning a smaller potential market. Larger and more complicated applications—feasible under Symbian OS—typically have longer development cycles and higher budgets. Because Symbian OS applications can address the hardware-specific features of a handset, and because underlying technologies vary among handsets, it is commonly necessary to reprogram part of a Symbian OS application to port it to a different device. However, applications written for one Series 60 device will run on all others (assuming adequate available memory and similar hardware capabilities—e.g., an application that requires a camera will obviously not work on a handset that doesn't include a camera).

For Nokia devices, cross-platform issues normally arise only if developers want the same application to run on both Series 60 devices and the Nokia Communicator

(Series 80). Even here, the main issue is the different form factor and UI of the devices, not any hardware incompatibility. C++ application development is typically more challenging than Java development; with Symbian OS phones, particular care must be taken to deal with garbage collection and memory leak issues, because Symbian OS applications are expected to run for months or years without crashing. Garbage collection is much less of an issue in Java technology. Symbian OS applications tend to be much larger than J2ME MIDlets, and are therefore slow and costly to download over the air. Instead, it's typical to download them over the Internet to a desktop machine, then hot-synch them to the phone. Once the application is on the desktop machine, it can then be e-mailed to other people, or "beamed" from the phone using Bluetooth or IrDA, creating a copy protection issue that is less of a problem with J2ME applications downloaded "over the air." There are, of course, solutions, but this is a business issue that needs to be addressed.

Symbian OS applications are preferable in a number of cases:

- When an application needs to interact with other phone features like voice telephony, or to synchronize with applications on other platforms. For example, it's far easier to build a cross-platform business information management system, synchronizing with phone books and corporate information, under Symbian OS than under J2ME.
- When an application is processor- or memory-hungry, and needs more than a few K of memory to run properly.

- When it is possible to specify to users what handset they will use (e.g., for a vertical enterprise), and the superior capabilities of Symbian OS handsets ensure a better and more efficient user experience.

Most game developers will choose J2ME over Symbian OS, because the size of the potential audience matters greatly for games.

3. Wireless Markup Language (WML)

The Wireless Markup Language is a tag-based language like HTML, designed for hardware-constrained, narrow-band wireless devices with limited input/output capabilities. WML documents use a card-and-deck metaphor, whereby a card is a single unit of user interaction and a deck is a set of related cards. Like an HTML page, a card typically contains some viewable content and perhaps some user choices for selecting an option, entering some data, or navigating to another card. Instructions in a card may invoke new static or dynamic decks from content servers.

The WML specification defines the intent of individual language tags, not how a particular user agent should implement those tags. Each WAP device vendor has a great deal of latitude, in deciding how to present WML tags to the user, the data entry mechanisms available, the size of the screen, and so on. WML is user-interface independent.

WML has the following general capabilities:

- Support for text and images, including presentation hints like line breaks, formatting, and other placement clues. WAP-compliant devices are not required to support images.
- Support for user input, including text entry, choice lists, and controls that invoke tasks. For instance, you can assign a URL to a specific device button such that, when the user presses that button, a GET request is sent to that URL's server.
- A variety of navigation mechanisms, based on the Internet-standard URL naming scheme, that lets you move between the cards in a deck or from deck to deck. Each WAP device may also incorporate a history mechanisms for cards already visited, so that the user can revisit a previous card just by pressing a Back button, much like revisiting a previous page in a Web browser.
- Support for multiple languages dialects through the use of Unicode character set.
- State and context management features, most notably variables that can be passed from deck to deck, variable substitution capabilities and user agent caching of variables and decks to maximize cache hits and minimize server requests.

WML Script

WMLScript is a lightweight but extensible procedural scripting language designed to enhance WML, somewhat like JavaScript enhances HTML documents. WML handles input and output, content rendering, and event processing but it has no serious computational capabilities. WMLScript fills that void. WMLScript lets you define functions that you can call from WML programs. Within these functions you have the full power of "if...then...else" statements, assignment statements, function calls, loop constructs, weakly typed basic data types such as Booleans and integers, and more. WMLScript also includes a full, robust set of assignment, logical, arithmetic, and comparison operators.

WMLScript is also an extensible language through the use of libraries. WMLScript 1.1, the version that corresponds to the WAP 1.1 specification, includes floating-point, string, URL, and dialog libraries, a library for basic functions like type conversions and string parsing, and a library of functions for interacting with the WAP microbrowser.

Although WMLScipt can be extended with new libraries, the WAP forum has not provided an open set of API's so that third parties can define WMLScript libraries.

4. SMIL

Synchronization Multimedia Integration Language (SMIL) is a simple but powerful markup language for specifying how and when clips play. MMS messages are sent using SMIL as the presentation language. It specifies how the various parts of the

message should be presented to the user – at what time and in which place in relation to the other parts. SMIL support in early MMS-capable terminals will be limited. The first MMS messages should be thought of as "slide shows." Each slide has at most two regions: one for text and the other for an image. The layout and ordering of the slides is specified using a layout language called SMIL. The actual text and the actual images are packaged as separate message elements, but within the same message body. The SMIL presentation simply defines where and when the various message elements will be displayed—see Figure 2.4.

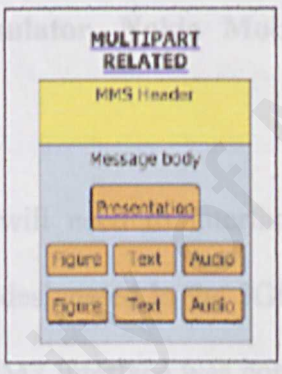

Figure 2.4: Bundling message elements

Most MMS-capable terminals will support reception of MMS messages containing multiple slides, each containing at most one image and one text part. Not all terminals will support speech or audio and not all terminals will support sending of MMS messages.

2.5 TOOLS AVAILABLE

Nokia has several tools available to help developers create MMS services, which we will introduce briefly here. *Getting Started with Nokia MMS Tools* has more information about how the tools can be used together, but for the most detailed information, please refer to the individual user's guide for each tool.

To test with real MMS terminals you will need access to a live MMSC. One possibility is to use the Forum Nokia Developer Hub services (see *forum.nokia.com/support*).

2.5.1 Nokia MMSC EAIF Emulator, Nokia Mobile Server Services (NMSS) Emulator

As a server-side developer, you will need to interface with the MMSC. This will probably happen over an interface designated in the 3GPP specs as MM7. However, as the first real specification of the MM7 interface was completed in the summer of 2002, any MMSCs from before that time will have some other, proprietary, interface for developers.

The Nokia MMS Center provides an External Application Interface (EAIF) that developers can use to communicate with the Nokia MMSC. This interface is proprietary. Now that 3GPP has finished specifying MM7, Nokia's MMSC will also have an MM7 standard-compliant interface. The EAIF interface will continue to be available in the Nokia MMSC for some time to give existing developers backward compatibility.

One of the first tools offered by Nokia for MMS developers is the EAIF Emulator, which mimics the interface between the MMSC and the third-party developer. Although

this tool is no longer supported, it is still available; it is small and fairly straightforward to use. It is only for testing applications with the Nokia EAIF interface – not the new 3GPP-specified MM7 interface.

Nokia's "next generation" emulator is the NMSS Emulator. It includes emulators for Nokia MMSC, Delivery Server, Terminal Management Server, iGMLC, and Presence Server.

The NMSS Emulator allows you to develop applications for several different Nokia products, if that is your aim. On the MMSC side it provides more in-depth information than the older tool about the messages that are sent and received, and also allows you to change the number of the recipient on-the-fly. This tool will include support for the MM7 interface as soon as it is available.

Either tool will help you test the functionality of your applications without needing to have access to a full MMSC.

2.5.2 Nokia MMS Java Library, Nokia Mobile Server Services (NMSS) API and Library

Another of Nokia's original MMS tools is the Java library for handling MMS messages. Like the EAIF Emulator, it is no longer supported, but it is still available for download, and comes with some very clear examples. It is one recommended *starting place* for MMS development. The early library does have limitations, however (also known as bugs), so we do not recommend using it as the primary tool for creating MMS services.

The newer version of this tool is the NMSS API and Library. This is the library that is supported, and it has been built in such a way that developers can create their applications using the EAIF interface, and then switch to the MM7 interface later without having to change their code. This should be the primary tool used for creating MMS services.

Both tools do essentially the same things – they can be used to construct an MMS message out of various bits and pieces, and to encapsulate the resulting message. They can then add HTTP headers to the message so that it is ready for sending to the EAIF or an EAIF Emulator. Received MMS messages can be "decapsulated" and disassembled into their separate parts.

Both libraries come with example applications, source code for the classes, and documentation

2.5.3 Nokia Developer's Suite for MMS

The Nokia Developer's Suite for MMS is a tool that integrates seamlessly with Adobe GoLive 6.0. Content can be created using Adobe GoLive, and the Nokia Developer's Suite (NDS) allows effortless encapsulation of the content into an MMS message. You can then send the MMS message to a server, EAIF Emulator, or MMS terminal emulator.

NDS for MMS can also stand alone – you can build an MMS message based solely on a SMIL file, or import one that has already been created (e.g., using one of the libraries), and then push it to a terminal emulator to see the result.

2.5.4 Nokia Series 60 SDK for Symbian OS

If you are creating applications for the client side, you will want to get the Series 60 SDK. It includes a Series 60 Emulator, so, for example, someone developing content for a Nokia 3650 or other terminal that uses the Series 60 Platform will find it very useful for seeing how content looks in an actual MMS terminal.

The Nokia Series 60 SDK for Symbian OS and accompanying documentation are available free of charge from Forum Nokia.

2.5.5 Series 60 Content Authoring SDK for Symbian OS, Nokia Edition

If you don't need the full power of the Nokia Series 60 SDK for Symbian OS, but you'd still like to see what your MMS messages will look like on a Series 60 terminal, you can install the Series 60 Content Authoring SDK.

2.5.6 Nokia Mobile Internet Toolkit

Beginning with version 3.1, the Nokia Mobile Internet Toolkit has support for MMS message testing. It features a handy MMS Wizard that can be used to easily piece together a message from various contents that the user has on hand, ready made. It will even create a presentation part for you in SMIL, which can be tweaked afterwards to get exactly the result desired. After this, the MMS message can either be pushed to one of the terminal emulators that support MMS, or saved for use (as an encoded, .mms file) with some other tool.

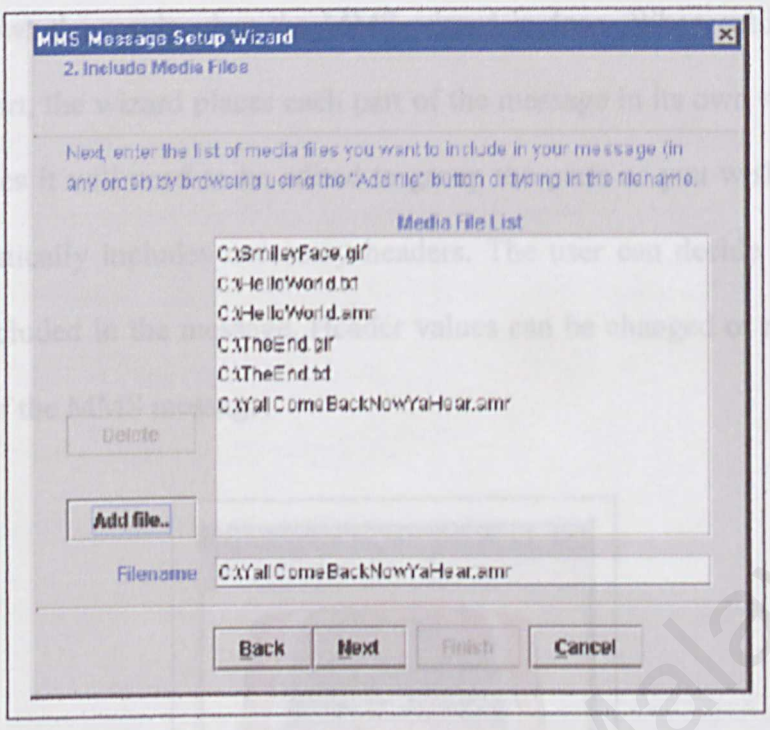

Figure 2.5: Using the MMS Message Setup Wizard

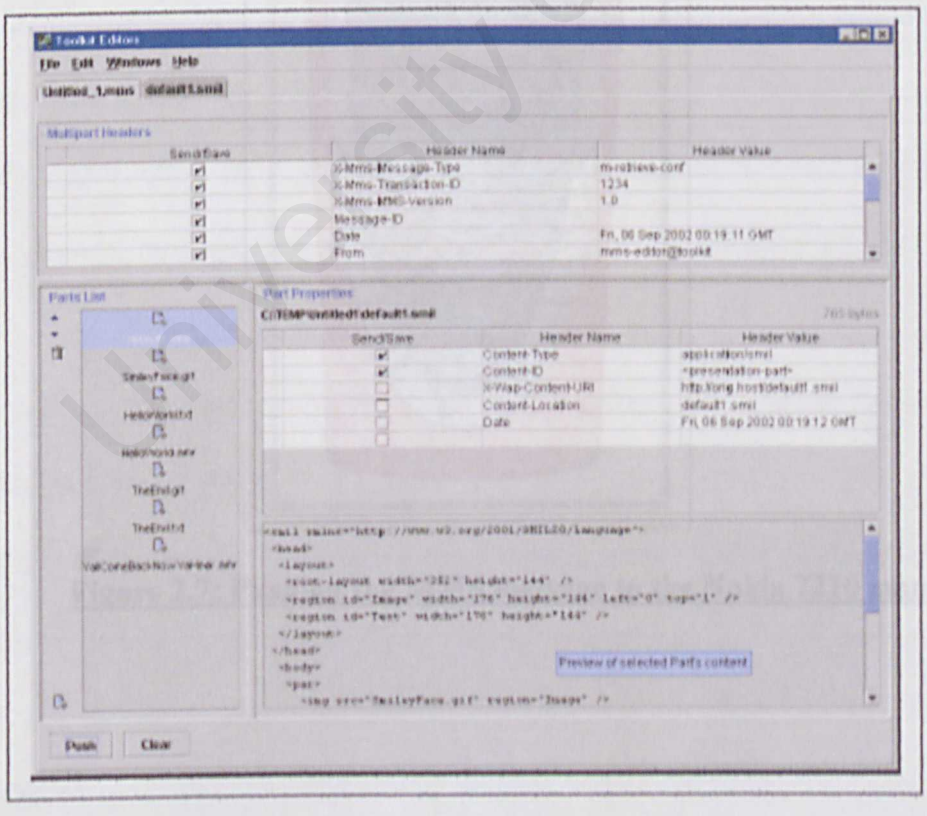

Figure 2.6: MMS message as seen by the Nokia Mobile Internet Toolkit

Figure 2.6 shows the result when the MMS wizard is done. When auto-generating the presentation part, the wizard places each part of the message in its own <par> container, so in most cases it will need to be edited to group the parts as you wish. Note that the wizard automatically includes necessary headers. The user can decide and edit which headers are included in the message. Header values can be changed or new ones added for each part of the MMS message.

Figure 2.7: Pushing the MMS message to the Nokia 7210 emulator

2.5.7 Terminal Emulators

To see how your MMS message will look in terminals with varying capabilities and screen sizes, there are several terminal emulators available:

- Series 60 Content Authoring SDK for Symbian OS, Nokia Edition
- Nokia 7210 Content Authoring SDK
- Nokia 3510i Content Authoring SDK
- Nokia 6650 MMS Concept SDK Beta
- Nokia 5100 SDK Beta

2.5.8 MMS Terminal Emulator Support for Nokia Mobile Server Services (NMSS) SDK

The MMS Terminal Emulator Support provides terminal emulator connectivity for application testing and development. It enables the sending of MMS messages from server-side applications to a terminal emulator and allows you to forward MMS messages from a terminal emulator to an application. It consists of the following elements:

- Series 60 Content Authoring SDK for Symbian OS, Nokia Edition, which emulates Series 60 based terminal
- Nokia 7210 Terminal Emulator, which emulates the Nokia 7210 terminal
- Nokia EAIF Connectivity, which allows connectivity to EAIF Service, such as Nokia Mobile Server Services SDK's Interface Emulator for MMSC or real MMSC

- Nokia Connectivity Framework, which provides the messaging environment between Terminal Emulators and Server Emulators.

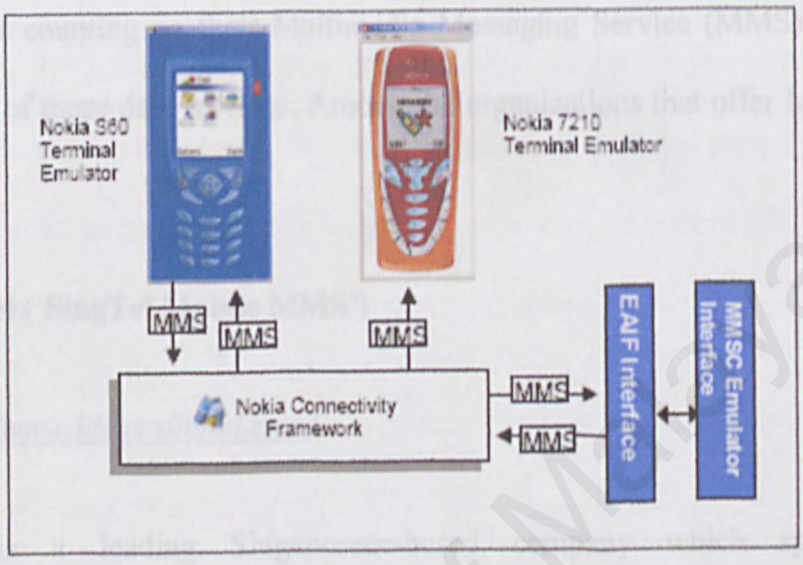

Figure 2.8: MMS Terminal Emulator Support for Nokia Mobile Server Services (NMSS) SDK

2.6 SIMILAR EXISTING TECHNOLOGIES

Lured by the remarkable success of photo messaging in Japan, virtually every operator in the world is counting on their Multimedia Messaging Service (MMS) as a way to propel growth of these data services. Among the organizations that offer MMS services are:

2.6.1 SingTel ('SingTel Mobile MMS')

(http://www.ideas.singtel.com)

SingTel is a leading Singaporean-based company which specializes in communications services. Since its incorporation in 1993, SingTel has strengthened itself as one of Asia's leading communications group, along with its regional partners, with operations and investments in more than 20 countries and territories around the world. This illustrious company has two main hubs located in Singapore and Australia.

The SingTel Group provides a wide range of communications services in Singapore and Australia, including:

- national telephone services;
- mobile communications services, including cellular, paging and satellite services;
- public data and private network services, including leased line, switched data, broadband, IP and Internet access services;

- International telephone services, including international direct dialing, calling cards, facsimile services and international corporate voice and wholesale voice services.

Through SingTel's innovative Mobile MMS service, mobile phone users are able to access exclusive news and 'live' traffic webcam shots and animated pictures, play games plus lots more -- all in full color. Mobile phone users are also able to integrate all these rich content with traditional text messages and even audio clips.

Apart from that, SingTel's Global MMS service allows mobile phone users to keep in touch with loved ones overseas.

Advantages:

1. Allows mobile phone users to access exclusive news and 'live' traffic webcam shots, animated pictures and play games.
2. Capable of sending MMS messages overseas.

Disadvantages:

1. Extensive range of contents. Doesn't entirely focus on soccer.
2. Too many services option available that may not be necessary for soccer-related issues.
3. A fee is imposed on the service.

Figure 2.9: A cut section of SingTel Mobile MMS service

2.6.2 T-Mobile USA ('My T-Mobile')

(http://www.t-mobile.com/services/picturemessaging/overview.asp)

Based in Bellevue, Washington, the U.S. operations of T-Mobile International AG & Co. K.G. consists of T-Mobile USA, Inc. (formerly VoiceStream Wireless) and Powertel, Inc. (together "T-Mobile") T-Mobile International AG & Co. K.G. is the mobile communications subsidiary of Deutsche Telekom AG (NYSE: DT). T-Mobile USA is a national provider of wireless voice, messaging and data services.

Advantages:

1. Allows you to take a picture or short video and send it wirelessly from your phone to any e-mail address.
2. No need for uploading files to a Web site or sending links to pictures rather than the pictures themselves.

Disadvantages:

1. Does not have specific user targets. For example, soccer fans and enthusiasts.
2. A fee is imposed on the service.

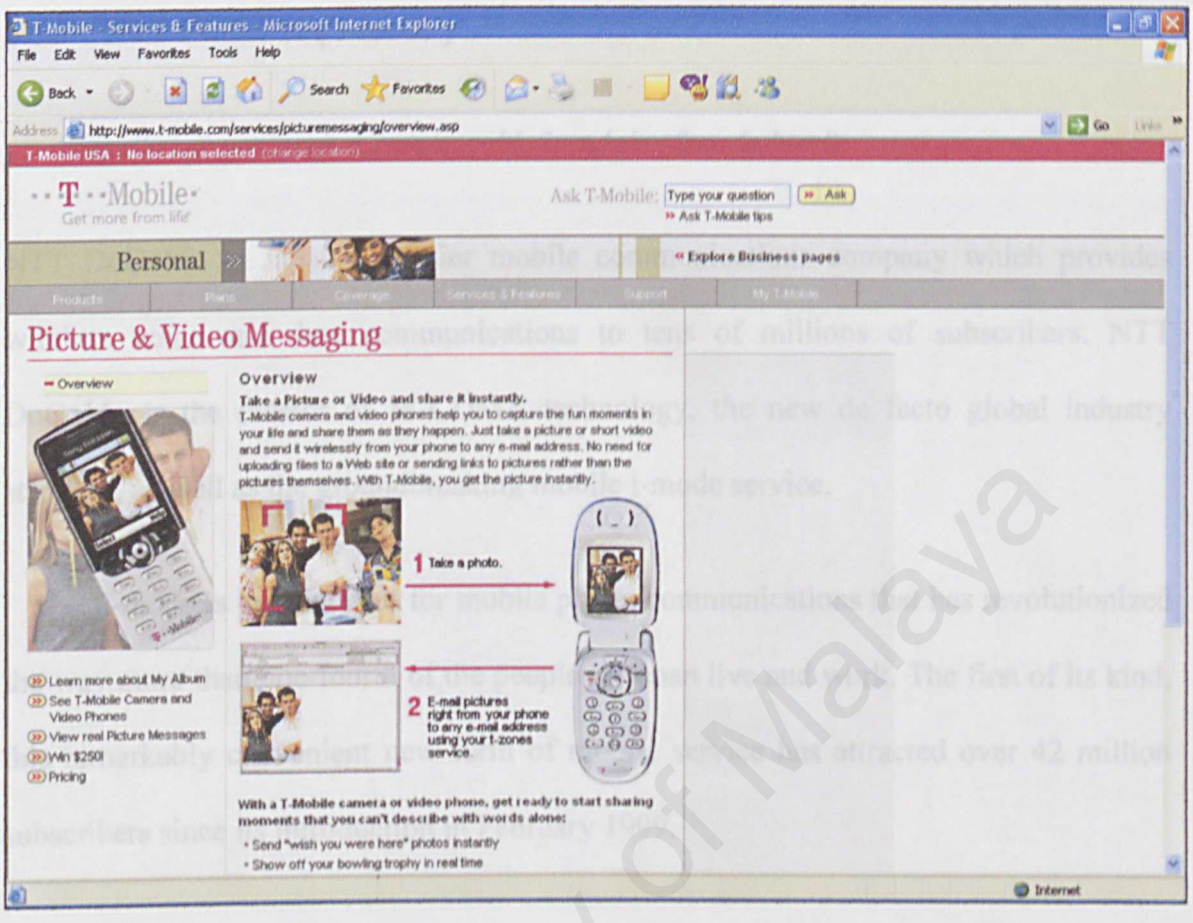

Figure 2.10: A cut section of My T-Mobile service

2.6.3 NTT DoCoMo ('i-mode')

(http://www.nttdocomo.com/corebiz/imode/try/imode.html)

NTT DoCoMo is Japan's premier mobile communications company which provides wireless voice and data communications to tens of millions of subscribers. NTT DoCoMo is the creator of W-CDMA technology, the new de facto global industry standard, as well as the groundbreaking mobile i-mode service.

i-mode is the platform for mobile phone communications that has revolutionized the way more than one-fourth of the people in Japan live and work. The first of its kind, this remarkably convenient new form of mobile service has attracted over 42 million subscribers since its introduction in February 1999.

With i-mode, cellular phone users get easy access to more than 83,000 Internet sites, as well as specialized services such as e-mail, online shopping and banking, ticket reservations, and restaurant advice. Users can access sites from anywhere in Japan, and at unusually low rates, because their charges are based on the volume of data transmitted, not the amount of time spent connected. NTT DoCoMo's i-mode network structure not only provides access to i-mode and i-mode-compatible content through the Internet, but also provides access through a dedicated leased-line circuit for added security.

NTT DoCoMo had the foresight to create i-mode at a time when the Japanese market for mobile phones was reaching maturity and users were in need of new services. i-mode didn't just generate new revenue for the company in a saturated marketplace — it

was a resounding success with customers and has redefined the meaning of mobile communications.

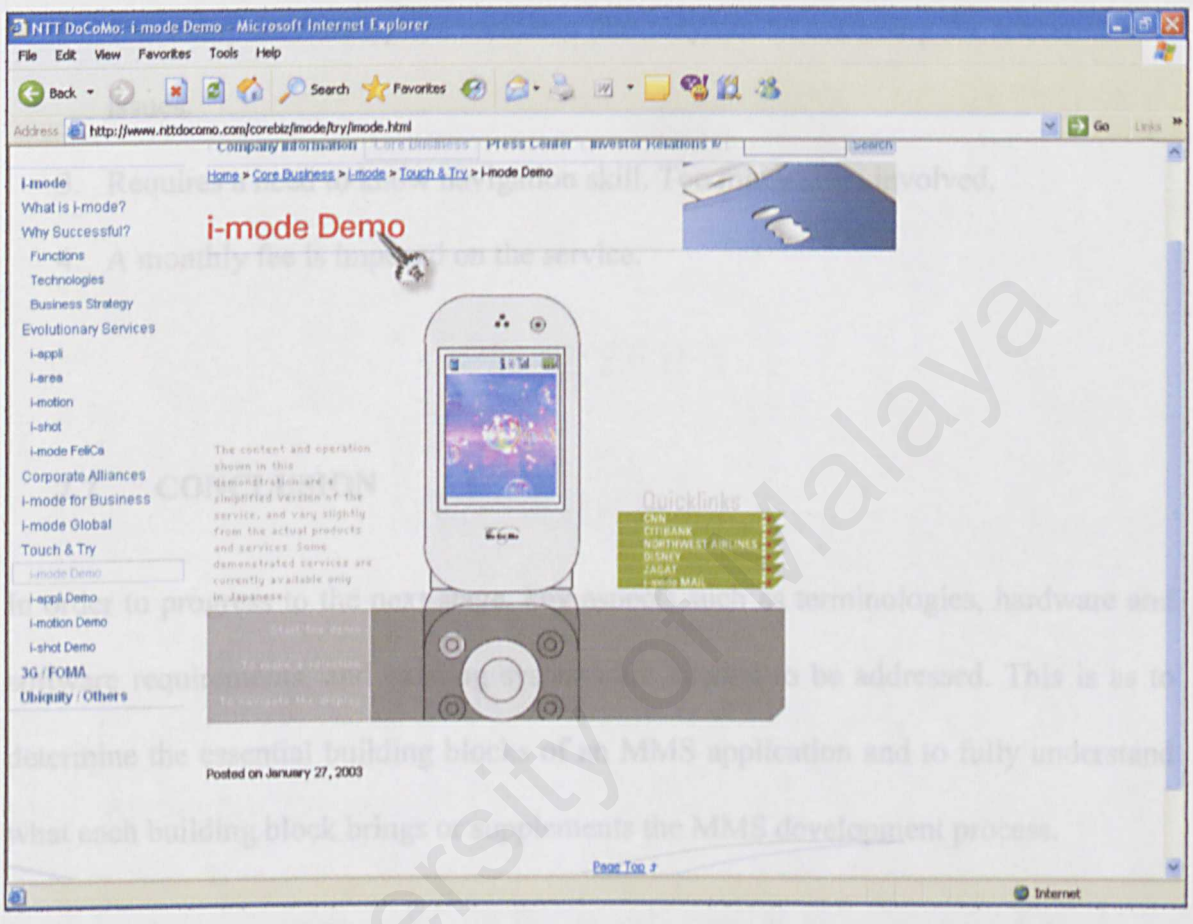

Figure 2.11: A cut section of the i-mode service

Advantages:

1. Enable easy access to more than 4,300 Japanese and English-language web sites.
2. Offers specialized services such as e-mail, online shopping and banking, ticket reservations, and restaurant advice.

Disadvantages:

1. Extensive range of contents. Doesn't entirely focus on soccer.
2. Too many services option available that may not be necessary for soccer-related issues.
3. Requires a need to know navigation skill. Too many steps involved.
4. A monthly fee is imposed on the service.

2.7 CONCLUSION

In order to progress to the next stage, key aspects such as terminologies, hardware and software requirements, and existing systems are needed to be addressed. This is as to determine the essential building blocks of an MMS application and to fully understand what each building block brings or supplements the MMS development process.

Moreover, by examining existing systems, we are able to obtain a brief idea on what our system should look like once it has finished. Through this also, we are able to alleviate uncertainties such as how to create our user interface or how to make it user-friendly and so on.

CHAPTER 3

METHODOLOGY

3.0 INTRODUCTION

Methodology is an integral part of system development in which to determine the software process model, that we would want our system to be based upon. It also highlights the essential tools that we are going to use to build our proposed system.

A software process is a set of activities and associated results which lead to the production of a software product. These may involve the development of software from scratch although it is increasingly the case that new software is developed by extending and modifying existing systems. In this case, it is the proposed system.

However, a software process model is an abstract representation of a software process. Each process model represents a process from a particular perspective so only provides partial information about the process.

Although there are many different software processes, there are fundamental activities which are common to all software processes. These are:

1. **Software specification**

   The functionality of the software and constraints on its operation must be defined.

2. **Software design and implementation**

   The software to meet the specification must be produced.

3. **Software validation**

   The software must be validated to ensure that it does what the customer wants.

4. **Software evolution**

   The software must evolve to meet changing customer needs.

3.1 METHODOLOGY USED

After careful consideration, I have selected the 'Waterfall' model as my preferred software process model to develop my system based upon.

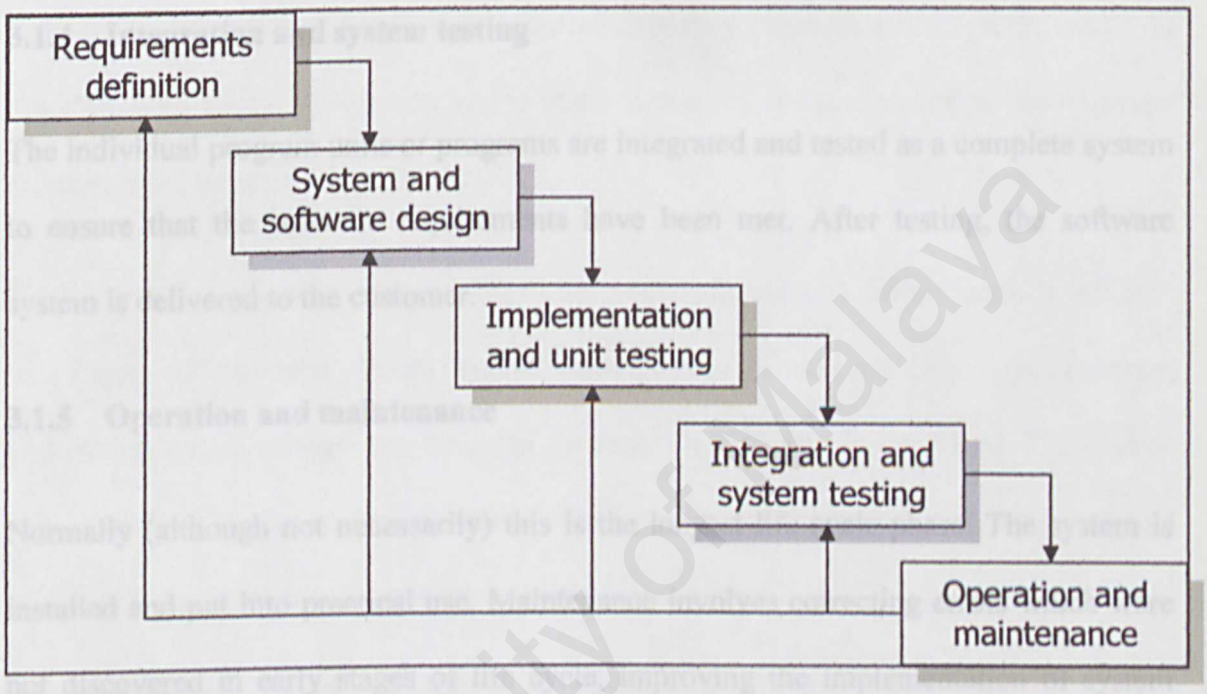

Figure 3.1: Waterfall Model

3.1.1 Requirements analysis and definition

The system's services, constraints and goals are established by consultation with system users. They are then defined in detail and serve as a system specification.

3.1.2 System and software design

The system design process partitions the requirements to either hardware or software systems. It establishes overall system architecture. Software design involves identifying and describing the fundamental software system abstractions and their relationships.

3.1.3 Implementation and unit testing

During this stage, the software design is realized as a set of programs or program units. Unit testing involves verifying that each unit meets its specification.

3.1.4 Integration and system testing

The individual program units or programs are integrated and tested as a complete system to ensure that the software requirements have been met. After testing, the software system is delivered to the customer.

3.1.5 Operation and maintenance

Normally (although not necessarily) this is the longest life-cycle phase. The system is installed and put into practical use. Maintenance involves correcting errors which were not discovered in early stages of life cycle, improving the implementation of system units and enhancing the system's services as new requirements are discovered.

3.2 CHARACTERISTICS OF THE 'WATERFALL' MODEL

In principle, the result of each phase is one or more documents which approved ('signed are off'). The following phase should not start until the previous phase finished. In practice, these stages overlap and feed information to each other. During design, problems with requirements are identified, during coding design problems are found and so on. The software process is not a simple linear model but involves a sequence of iterations of the development activities.

Because of the costs of producing and approving documents, iterations are costly and involve significant rework. Therefore, after a small number of iterations, it is normal to freeze parts of the development, such as the specification, and to continue with the later development stages. Problems are left for later resolution, ignored or are programmed around. This premature freezing of requirements may mean that the system won't do what the user wants. It may also lead to badly structured systems as design problems are circumvented by implementation tricks.

During the final life-cycle phase (operation and maintenance) the software is put into use. Errors and omissions in the original software requirements are discovered. Program and design errors emerge and the need for new functionality is identified. The system must therefore evolve to remain useful. Making these changes (software maintenance) may involve repeating some or all previous process stages.

The problem with the waterfall model is its inflexible partitioning of the project into these distinct stages. Commitments must be made at an early stage in the process and this means that it is difficult to respond to changing customer requirements. Therefore, the waterfall model should only be used when the requirements are well understood. However, the waterfall model reflects engineering practice. Consequently, software processes based on this approach are still used for software development, particularly when this is part of a larger systems engineering project.

3.3 SYSTEM REQUIREMENTS

Software system requirements are often classified as functional or non-functional requirements or as domain requirements.

3.3.1 Functional Requirements

The functional requirements for a system describe the functionality or services that the system is expected to provide.

For the Soccer365 MMS application, the functional requirements are:

- Info Module - This module allow users to view information on various world-famous soccer clubs that exist within the soccer region, such as Manchester United, Barcelona, etc. In addition, users are also able to view information on their favorite soccer players that they idolize. Finally, users are able to access the help index when they require assistance on using the application.
- Services Option Module – this module allow users to view and download animation and pictures depicting soccer-related contents.

3.3.2 Non-Functional Requirements

Non-functional requirements, as the name suggests, are those requirements which are not directly concerned with the specific functions delivered by the system.

The following are the non-functional requirements that are embedded into the proposed system:

- **User-friendly**

An important aspect that the system must possess as to avoid any unsolicited difficulties to users when they are manipulating with the functions of the system. A good user interface is required to fulfill this criterion.

- **Reliability**

The system should be maintainable. Whenever a problem occurs, the system should be able to detect the problem and debug it with ease. Moreover, the system should be stable at all times, meaning it shouldn't in any circumstances 'hang'.

- **Effectiveness**

Image resolution should be sufficient and optimized in order for the user to obtain quality picture content on their mobile phones.

- **Efficiency**

The proposed system should be capable of running quickly and efficiently in response to every user request. Users should be able to view their request without much delay.

3.4 TECHNOLOGY CONSIDERED

Listed below are the hardware and software specifications for the development of this particular application.

Hardware:

- 500 MHz Intel Pentium and above.
- 64 MB or more.
- An MMS-enabled mobile phone
- 65,000 color or better display card.
- Windows 2000, Windows 2000ME, Windows N.T 4.0 or Windows XP

Software:

3.4.1 Macromedia Dreamweaver MX

Dreamweaver includes an HTML editor that performs all the basics of other HTML editors, plus allows you to manipulate and control HTML code directly using the Quick Tag Editor. Also learn how Dreamweaver allows the customization of web sites by creating Cascading Style Sheets, JavaScript, DHTML, ASP, and XML. Other topics include the Timelines inspector, behaviors, and Dreamweaver's full-fledged FTP client.

In the design category, the macromedia Dreamweaver MX, there are certain new futures such as improve workspace layout, predefined sample page layouts and code, Cascading Style Sheets (CSS) support an enhanced dream weaver templates.

3.4.2 Nokia WAP Toolkit

The Nokia Mobile Internet Toolkit is an excellent tool that should be kept in every wireless developer's toolbox. It allows developers to remain a step ahead of the general market by being able to build and test tomorrow's features—such as MMS, WTAI, and XHTML—today.

3.4.3 Microsoft SQL Server 2000

Microsoft SQL Server is a scalable, high-performance database management system. Designed to meet the requirements of distributed client-server computing, SQL Server is tightly integrated with the Microsoft BackOffice family of servers to enable organizations to improve decision-making and streamline business processes.

Selecting a database platform is one of the most important decisions your company will make. Microsoft SQL Server is your solution to complex business problems. SQL Server's built-in Internet integration gives organizations the ability to build Active Web sites, conduct business on the Internet and build corporate Intranet sites using open, high-performance solutions. Reduced complexity for users, administrators and developers means quicker, easier-to-use business solutions at lower costs.

3.4.4 ASP. Net

ASP.NET (sometimes referred to as ASP+) is the latest version of Microsoft's Active Server Pages technology (ASP). ASP.NET is different than its predecessor in two major ways: it supports code written in compiled languages such as Visual Basic, C++, C#, and Perl, and it features server controls that can separate the code from the content, allowing WYSIWYG editing of pages. Although ASP.NET is not backwards compatible with ASP, it is able to run side by side with ASP applications. ASP+ files can be recognized by their .aspx extension.

3.4.5 Microsoft IIS

IIS is an internet file and application server included in the Win NT option pack. IIS is user-friendly because it is easy to configure and can be used alone as web server. It also guarantees the same security, networking, and an administrator and user functionality because in inherits all Windows NT futures. IIS also can help admin secure websites and develop and deploy server. It can support a variety of application such as Common Gateway Interface (GUI), ASP, and Socket Secure Layers (SSL).

3.5 CONCLUSION

In this chapter, the methodology has been discussed in detail including system requirement and tools for the project selected. Next chapter 4, system design will be discussed.

CHAPTER 4

SYSTEM ANALYSIS & DESIGN

4.0 INTRODUCTION

The methodology of creating a computer system that takes into account such factors as user needs, performance levels, database design, hardware specifications, and data management.

4.1 PROCESS MODELLING

Process modeling refers to modeling business process or the functional or aspects of the system. These include reading data into a process, processing or transforming the data into information, writing data to a data store, and printing reports. In other words, process modeling refers to these processes that business needs to read input data, transform data into useful information, and output the information in the form of report.

Processes in a system are depicted using Data Flow Diagrams (DFDs). The DFDs are drawn at different levels to represent different levels of details. At the highest level the internal details are omitted, only input to the system and the entities that supply the data, and the output from system to the entities that need the data shown. The lower level is more detail on highest level.

4.2 DATA FLOW DIAGRAM

A Data Flow Diagrams (DFDs) is a technique used to show graphically the flow of data through a business system and the processes performed by the system. The DFDs gives the overview of the system inputs and outputs, processes and the flow of data through each system.

The advantages using DFDs are:

- It is easy to use and understand since only four symbols are used.
- It enables the system to be structured into independent units of a desirable size and helps the analysts to better understand the relationships between the systems and their subsystems.
- It is used as a communication tool between the analysts and the users.
- Its helps the analysts to identify the required data and processes of the proposed system and making sure that they have been defined.

- It gives the analysts freedom from committing to the technical implementation of the system too early.

4.3 SYSTEM MODULE

The project system consists of three modules, namely Info Module, Services Option Module and the Administrator Module. Below is a diagram that shows the overall layout of the system module.

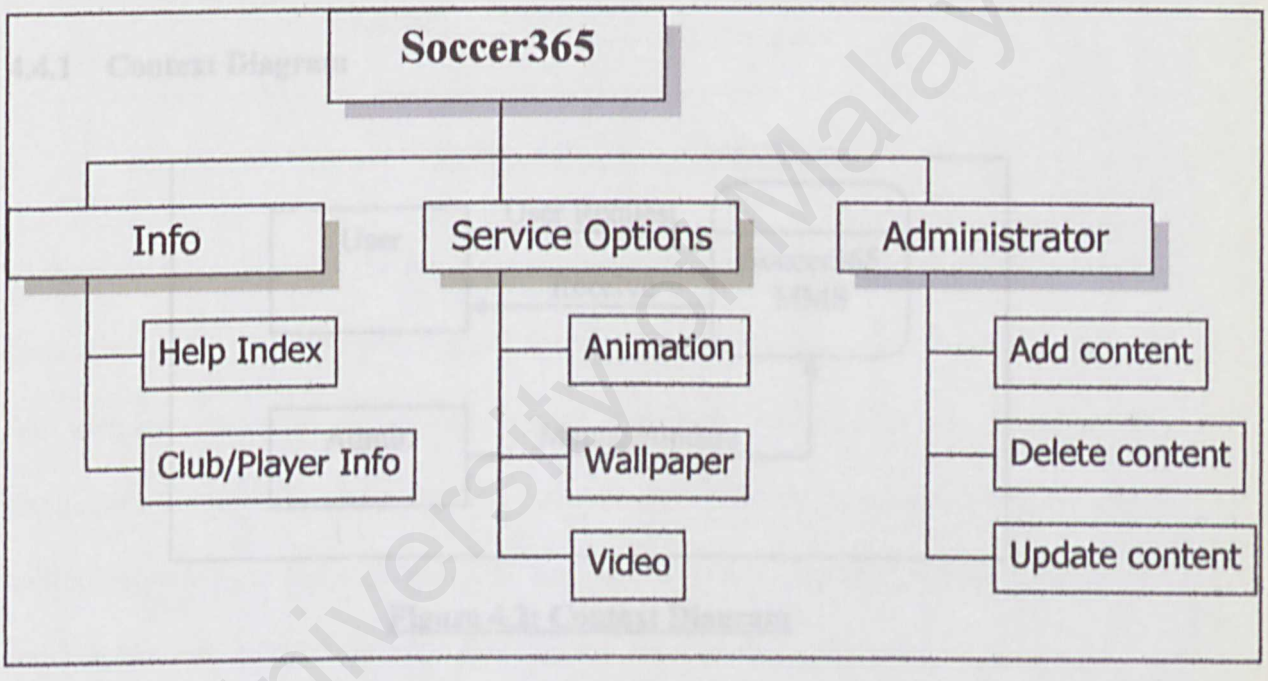

Figure 4.1: System Module Diagram

Info Module: This module is broken down into two sub-modules. Firstly, users are able to access the help index when they require assistance on using the application. Users are also able to view information on various world-famous soccer clubs that exist within the soccer region, such as Manchester United, Barcelona, etc. In addition, users are also able to view information on their favorite soccer players that they idolize.

Service Options Module: This module has two sub-modules. It allows users to view and download animation or pictures depicting soccer-related contents.

Administrator Module: This module enables to administrator to add the user, delete the user and update the contents of the database.

4.4 SYSTEM FUNCTIONALITY

4.4.1 Context Diagram

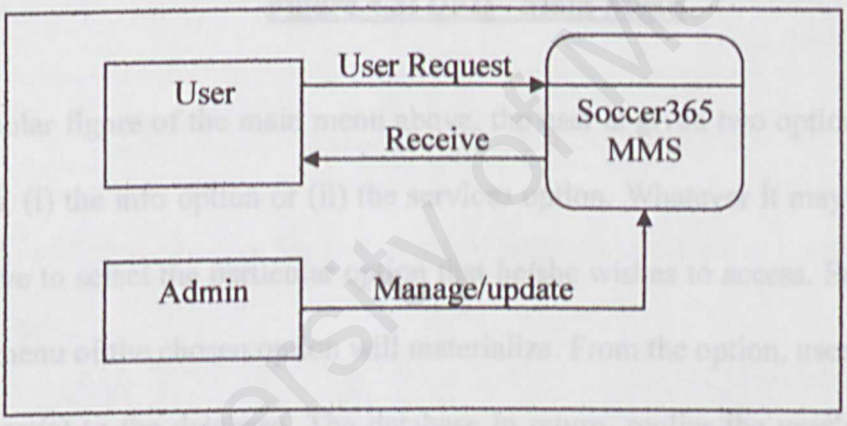

Figure 4.2: Context Diagram

In the above figure of the context diagram, the user will submit his/her request to the Soccer365 MMS application. Subsequently, the application will then reply the user's request by sending the particular information that the user requested. Moreover, the administrator is able to manage and update the contents within the Soccer365 MMS application.

4.4.2 DFD

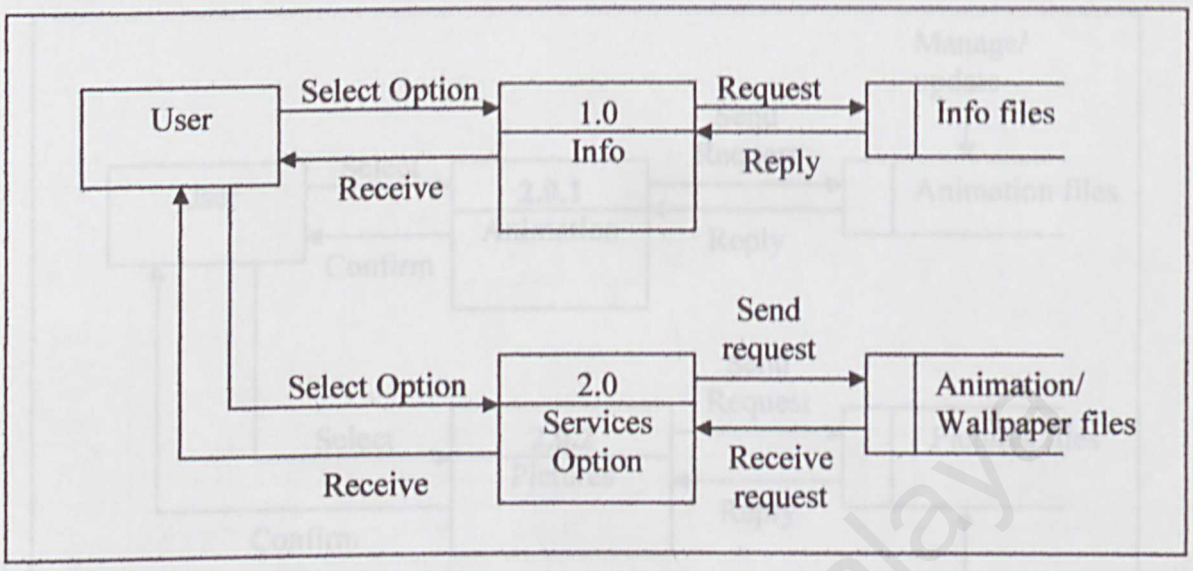

Figure 4.3: DFD - Main Menu

In the particular figure of the main menu above, the user is given two options to choose from, that is, (i) the info option or (ii) the services option. Whatever it may be, the user will then have to select the particular option that he/she wishes to access. Subsequently, the display menu of the chosen option will materialize. From the option, users are able to submit request(s) to the database. The database in return, replies the user's request by sending the particular information that the user requested.

4.4.3 Child Diagram

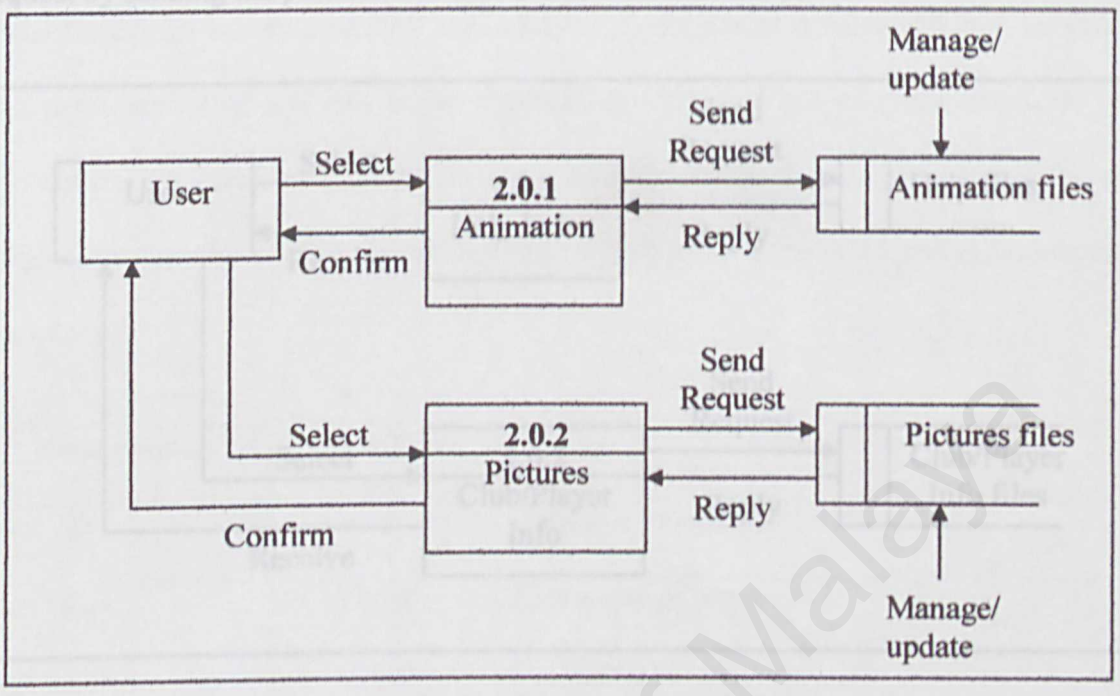

Figure 4.4: Child Diagram - Services Option Menu

In this particular figure, the user has two choices, either to select the animation option or the pictures option.

If the user chooses to select the animation option, the display menu of this option will then materialize. He or she will then have requested a particular animation. The animation module will send the user's request to its respective database, animation files. The database in return, replies the user's request by sending the particular information that the user requested.

On the other hand, selecting the pictures option will prompt the display menu of this option to materialize. The user may then choose any pictures she wants to download. Subsequently, the pictures module will send the user's request to the respective

database, which in this case, the pictures files. The database in return, replies the user's request by sending the particular information that the user requested.

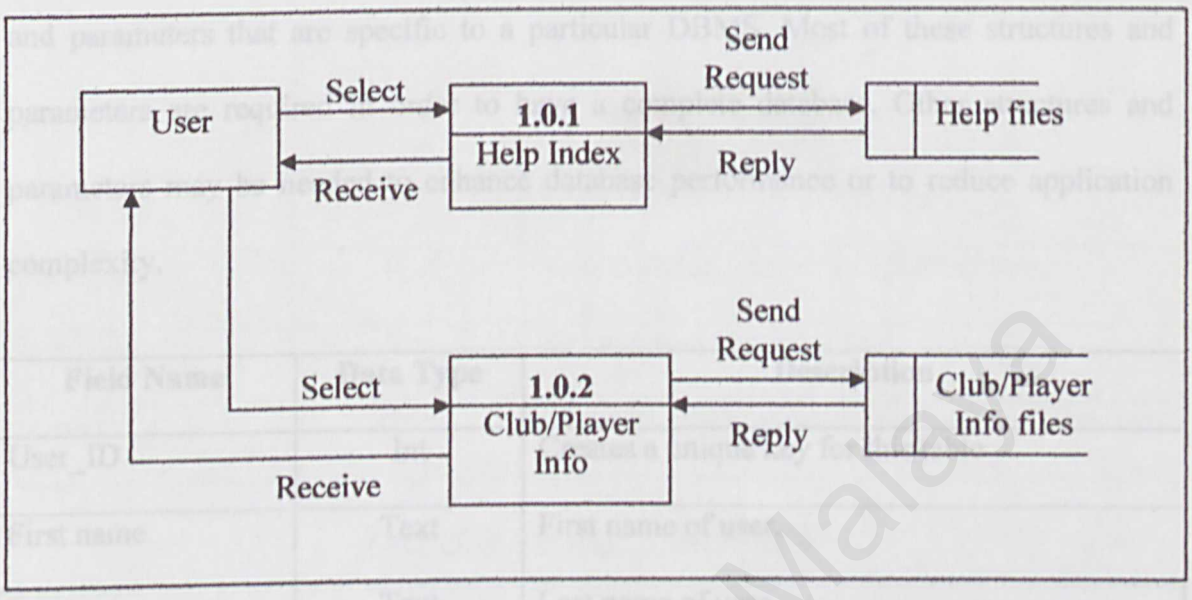

Figure 4.5: Child Diagram - Info Menu

From the info menu, the user is able to access two options, the help index and the club/player info. Whichever the user selects, the respective option will forward the request to the respective database. For the help index, the help files database will be accessed whereas for the club/player info, the club/player info files will be accessed. These databases will then reply the user's request by sending the particular information that the user requested.

4.5 DATABASE DESIGN

Database design is a process by which a physical data model is enhanced with structures and parameters that are specific to a particular DBMS. Most of these structures and parameters are required in order to have a complete database. Other structures and parameters may be needed to enhance database performance or to reduce application complexity.

Field Name	Data Type	Description
User_ID	Int	Creates a unique key for this table
First name	Text	First name of user.
Last name	Text	Last name of user
E-mail	char	E-mail address of user
Mobile phone	Text	Mobile phone number of user
Nickname	Text	Nickname of user
Password	char	User password to login into the system

Table 4.1: User Profile

Field Name	Data Type	Description
Admin_ID	Int	Creates a unique key for this table
First name	Text	First name of administrator.
Last name	Text	Last name of administrator
E-mail	char	E-mail address of administrator
Mobile phone	Text	Mobile phone number of administrator
Nickname	Text	Nickname of administrator
Password	char	Administrator password to login into the system

Table 4.2: Administrator Profile

Field Name	Data Type	Description
Info_ ID	Int	Creates a unique key for this table
Index	Int	Info index to select
Help	Memo	Display help index.
Information	Memo	To display soccer information

Table 4.3: Information field

Field Name	Data Type	Description
Animation_ID	Int	Creates a unique key for this table
Index	Int	User performs selection based on this number
Animation	Jif, gif, jpeg,	Animation content
URL	Text	Specifies the location of the content

Table 4.4: Animation field

Field Name	Data Type	Description
Picture_ ID	Int	Creates a unique key for this table
Index	Int	User performs selection based on this number
Picture	Jif, gif, jpeg,	Picture content
URL	Text	Specifies the location of the content

Table 4.5: Picture field

Field Name	Data Type	Description
User_ID	Int	This is a primary key
Info	memo	View information services
Animation	Jif, gif, jpeg,	Animation content
Picture	Jif, gif, jpeg,	Picture content
Administrator_ID	Int	This is a primary key
Receiver_ID	int	Receiver destination

Table 4.6: Server storage

4.6 CONCLUSION

System design is imperative and should be taken into consideration before any implementation is initiated in order to get the overall system flow. Also, to clearly define the ideas on how a system is to be developed.

CHAPTER 5:

SYSTEM IMPLEMENTATION

5.0. INTRODUCTION

The system implementation stage revolves around stages in development environment. In this phase, the system requirements and design are being implemented and convert into program code. All plans of the modules and interface design that have been suggested before will be implemented with using the suitable programming codes and software tools. Design characteristics such as low coupling and high cohesion, should also be program characteristics, so that the algorithms, functions, interface and data structure can be traced easily from design code and back again.

The main objective for development phase is actually to ensure that the system that we built have fulfilled all the plan and target that have been mentioned and agreed in the previous phases before. However due a certain limitations on the programming

languages, selected Database Management System, development platform which contradicts to the actual system design and therefore certain modifications are needed in order to develop the system in accordance to the limitations of the development tools chosen to develop the system.

5.1. DEVELOPMENT ENVIRONMENT

5.1.1. Hardware Used

- Any WAP enabled Mobile Phone with MMS Services
- 500 MHz Intel Pentium and above.
- 256 MB or more.
- Operating system Windows 2000 above, and Windows XP.

5.1.2. Software Used

- Internet Information Service (IIS) 7.0 - web server
- Microsoft Access 2003 - database
- Internet Explorer 6.0 - web browser
- Microsoft Frontpage 2003 – 1st code and interface editor
- Macromedia Dreamweaver MX – 2nd code and interface editor
- Adobe Photoshop 7.0 - image editor
- Microsoft Word 2003 - thesis documentation

5.2. DEVELOPMENT OF PROGRAMMING LANGUAGE

5.2.1. HTML (Hyper Text Markup Language)

HTML tags are used to mark-up HTML elements. HTML tags are surrounded by the two characters < and >. The surrounding characters are called angle brackets. HTML tags normally come in pairs like and . The first tag in a pair is the start tag; the second tag is the end tag. The text between the start and end tags is the element content. HTML tags are not case sensitive; means the same as <B>.

5.2.2. ASP (Active Server Page)

Active Server Pages (ASP) makes it easy to generate dynamic content for the Web and to build powerful Web applications. Whether you are a Web designer or a Web developer, this introduction explains how ASP can help you.

For the HTML Author

If you are an HTML author, you will find that server-side scripts written in ASP are an easy way to begin creating more complex, real-world Web applications. If you have ever wanted to store HTML form information in a database, personalize Web sites according to visitor preferences, or use different HTML features based on the browser, you will find that ASP provides a compelling solution. For example, previously, to process user input on the Web server you would have had to learn a language such as Perl or C to build a conventional Common Gateway Interface (CGI) application. With ASP, however, you can

collect HTML form information and pass it to a database using simple server-side scripts embedded directly in your HTML documents. If you are already familiar with scripting languages such as Microsoft VBScript or Microsoft® JScript® (JScript is the Microsoft implementation of the ECMA 262 language specification), you will have little trouble learning ASP. If you want to get started right away with ASP, see the ASP Tutorial. You can then return to these topics for more detailed information on writing server-side scripts.

For the Experienced Web Scripter

If you are skilled at a scripting language such as VBScript, JScript, or PERL, you already know how to use Active Server Pages. What more, in your ASP pages you can use any scripting language for which you have installed a COM compliant scripting engine. ASP comes with VBScript and JScript scripting engines, but you can also install scripting engines for PERL, REXX, and Python, which are available through third-party vendors.

For the Web Developer and Programmer

If you develop back-end Web applications in a programming language, such as Visual Basic, C++, or Java, you will find ASP a flexible way to quickly create Web applications. Besides adding scripts to create an engaging HTML interface for your application, you can build your own COM components. You can encapsulate your application's business logic into reusable modules that you can call from a script, from another component, or from another program.

The Active Server Pages Model

A server-side script begins to run when a browser requests an .asp file from your Web server. Your Web server then calls ASP, which processes the requested file from top to bottom, executes any script commands, and sends a Web page to the browser. Because your scripts run on the server rather than on the client, your Web server does all the work involved in generating the HTML pages sent to browsers. Server-side scripts cannot be readily copied because only the result of the script is returned to the browser. Users cannot view the script commands that created the page they are viewing.

5.2.3. SQL Statement (Structured Query Language Statement)

An SQL query is a query you create by using an SQL statement. You can use Structured Query Language (SQL) to query, update, and manage relational databases such as Microsoft Access. When you create a query in query Design view, Access constructs the equivalent SQL statements behind the scenes for you. In fact, most query properties in the property sheet in query Design view have equivalent clauses and options available in SQL view. If you want, you can view or edit the SQL statement in SQL view. However, after you make changes to a query in SQL view, the query might not be displayed the way it was previously in Design view.

Some SQL queries, called SQL-specific queries, can't be created in the design grid. For pass-through, data-definition, and union queries, you must create the

SQL statements directly in SQL view. For sub queries, you enter the SQL in the Field row or the Criteria row of the query design grid.

Where SQL statements are used

You can use SQL statements in many places in Access where you can enter the name of a table, query, or field. In some cases, Access fills in the SQL statement for you. For example, when you use a wizard to create a form or report that gets data from more than one table, Access automatically creates an SQL statement that it uses as the setting for the Record Source property of the form or report. When you create a list box or combo box with a wizard, Access creates an SQL statement and uses it as the setting for the Row Source property of the list box or combo box. Without using a wizard, you can generate an SQL statement for the Record Source or Row Source properties by clicking the Build button next to either of these properties, and then creating a query in query Design view. You can also use SQL statements programmatically in:

- The SQL Statement argument of the RunSQL macro action.
- Code as a literal string, or as an SQL statement that includes variables and controls.
- The SQL property of a QueryDef objects to change the underlying SQL statement of a query.

You can type an expression in an SQL SELECT statement, or in WHERE, ORDER BY, GROUP BY, or HAVING clauses. You can also type an SQL

expression in several arguments and property settings. For example, you can use an SQL expression as a:

- Where Condition argument of the OpenForm or ApplyFilter action.
- Domain or criteria argument in a domain aggregate function.
- Setting for the RecordSource or RowSource property in forms and reports.

5.2.4. JavaScript

JavaScript was designed to add interactivity to HTML pages. JavaScript is a scripting language - a scripting language is a lightweight programming language. JavaScript is lines of executable computer code. A JavaScript is usually embedded directly in HTML pages. JavaScript is an interpreted language (means that scripts execute without preliminary compilation). Everyone can use JavaScript without purchasing a license. JavaScript is supported by all major browsers, like Netscape and Internet Explorer.

5.3. DATABASE COMPONENT

5.3.1. Component Diagram 1

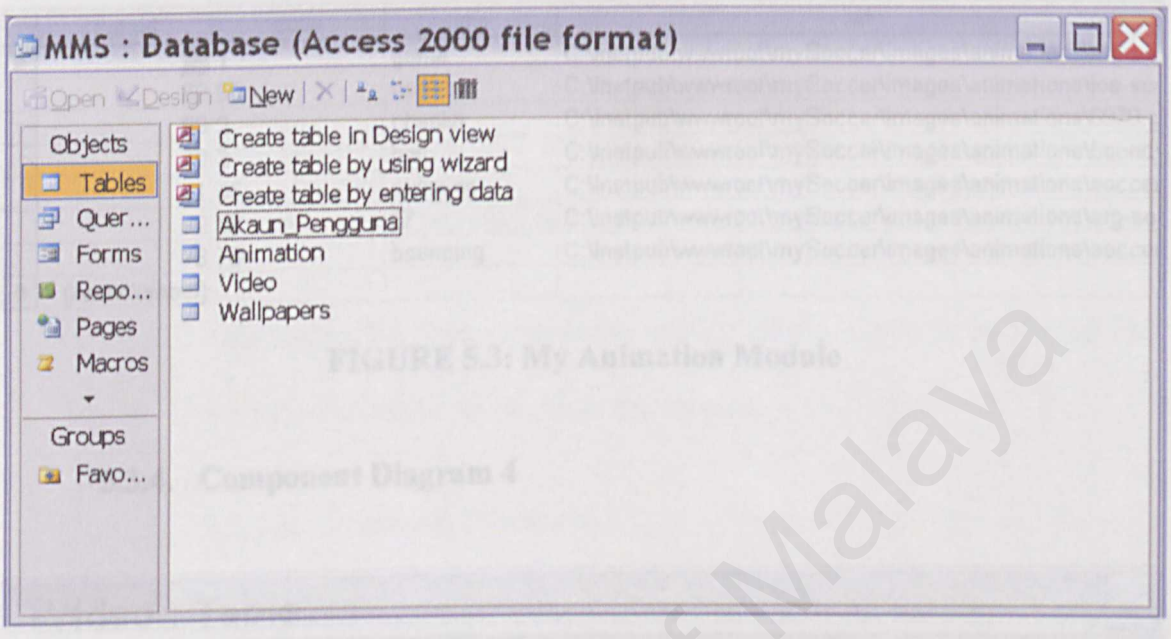

FIGURE 5.1: List of Tables

5.3.2. Component Diagram 2

Akaun_Pengguna : Table

no_pengguna	name	password	tahap
K189	pawen	pawen	Kakitangan
K351	aidil	aidil	Kakitangan
K413	khaizer	khaizer	Pentadbir

FIGURE 5.2: My Login Module

5.3.3. Component Diagram 3

Animation : Table

dataid	code	info	url
67	1	game	C:\Inetpub\wwwroot\mySoccer\images\animations\arg-so
68	99	cartoon	C:\Inetpub\wwwroot\mySoccer\images\animations\joe-so
69	2	chaplin	C:\Inetpub\wwwroot\mySoccer\images\animations\6970.g
70	3	ball	C:\Inetpub\wwwroot\mySoccer\images\animations\bounc
71	45	spinning	C:\Inetpub\wwwroot\mySoccer\images\animations\socce
72	56	67	C:\Inetpub\wwwroot\mySoccer\images\animations\arg-so
73	78	bouncing	C:\Inetpub\wwwroot\mySoccer\images\animations\socce
(AutoNumber)			

FIGURE 5.3: My Animation Module

5.3.4. Component Diagram 4

Video : Table

dataid	code	info	url
65	11	nike football	C:\Inetpub\wwwroot\mySoccer\video\nike_football.3gp
66	22	rivaldo	C:\Inetpub\wwwroot\mySoccer\video\Rivaldo.3gp
67	1110	perak	C:\Inetpub\wwwroot\mySoccer\video\perak.3gp
(AutoNumber)			

FIGURE 5.4: My Video Module

5.3.5. Component Diagram 5

Wallpapers : Table

dataid	code	info	url
64	22	united we stand	C:\Inetpub\wwwroot\mySoccer\images\wallpapers\uws.jp
65	33	victory!	C:\Inetpub\wwwroot\mySoccer\images\wallpapers\victorio
66	23	mickey mouse	C:\Inetpub\wwwroot\mySoccer\images\wallpapers\muem
67	1	dartboard	C:\Inetpub\wwwroot\mySoccer\images\wallpapers\dart.jp
68	2	The Best	C:\Inetpub\wwwroot\mySoccer\images\wallpapers\thebes
(AutoNumber)			

FIGURE 5.5: My Wallpaper Module

5.4. MODULE IMPLEMENTATION

5.4.1. Administration Module

5.4.1.1. Login & Logout

This login function will perform the username and password checking before the admin or staff can log into the system.

This logout function cancels the session of the admin or staff after they have finished their work inside the system.

5.4.1.2. Profile & Password

This function allowed the admin or staff to edit their profile and change their password too as well. It will be their username by default.

5.4.1.3. Add, edit & delete item

Some of the information listed here can be modified by the admin or the staff. They can edit the information or data that they or the other user have entered. They can also add some information that they might want to. And, they can also delete certain information for example the unwanted registered user.

5.4.1.4. Status

This function allows users to view data that is stored in the database. It also views data details and information.

5.4.2. User Module

5.4.2.1. Download Animation

This function allows users to select and download their preferred animation content

5.4.2.2. Download Video

This function allows users to select and download their preferred video content

5.4.2.3. Download Wallpaper

This function allows users to select and download their preferred wallpaper content.

5.5. CONCLUSION

During system implementation, system requirements and designs were converted into program codes. Besides, it also involves development of the environment setting such as the operating system and the database server. Several software tools were used to deploy the design into machine-readable language and then, turn them, in order to produce the required applications.

CHAPTER 6

SYSTEM TESTING

6.0. INTRODUCTION

After the program has been coded, it is time to test the program. Testing is one of the vital parts of system development. During this particular phase, the system is tested for bugs and errors. When suspected bugs or errors are detected, the process of fault remover is initiated so that these unsolicited events can be removed. After the removal process, the system is then able to function properly without any errors occurring. Moreover, the objective of this phase is also to determine whether the system requirements have been met.

In developing a large system, testing usually involves three stages. There are:

- Unit Testing
- Integration Testing
- System Testing

6.1. UNIT TESTING

The primary goal of unit testing is to confirm that the unit has been coded correctly and performs the function and logic of what it is supposed to perform. This stage of testing verifies the component functions properly with types of input expected from studying the component's design.

The first step in unit testing is to examine the code. The code for each component is reviewed and is compared it its documentation for misunderstanding, inconsistencies and other fault. This process is also known as code review. Secondly, we can use the Control Object Testing technique to do the unit testing. Command buttons are clicked to test their functionality and text boxes are tested with different data types and also null value to make sure invalid data will not cause any fault.

After we perform testing on different data types such as numbers, characters or date, we then begin testing on certain functions because some control objects will only accept certain data types. Invalid data type can be traced by the system without causing any error.

Next, we perform the test case. Test case is developed to ensure that the input is properly converted to the desired output. So, to test a component, input data and condition are chosen. Then the component is allowed to manipulate the data and output is observed.

6.2. INTEGRATION TESTING

When unit testing has been successfully completed, meaning that everything is working accordingly and no such errors are detected, it is time to combine them into a working system. This integration is planned and coordinated so when a failure occurs, the failure can be found easily. There are a few techniques which can be performed in the integration testing. Bottom-up integration, Bid-Bang Integration, Sandwich Integration and Comparison of Integration Strategies.

The selected approach for this system's integration testing is the Bottom-up Integration technique. This is one of the most popular approaches which combines and integrate components into a larger system. When this method is used, each component in the lower level of the system is tested individually first. Then the next components to be tested call the previous tested ones. This approach is followed repeatedly until all components are included in the testing.

6.3. SYSTEM TESTING

System testing is required to ensure that the system meets the user requirements that have been defined earlier. There are several important steps in system testing:

- Function Testing
- Performance Testing
- Acceptance Testing
- Installation Testing

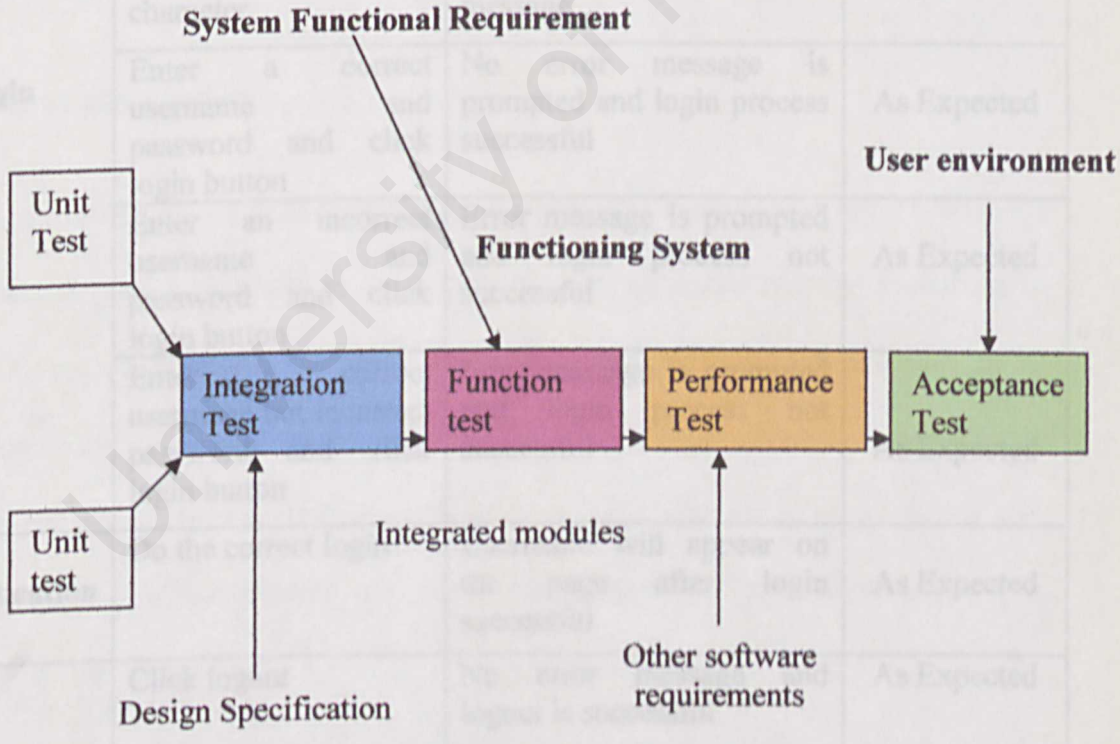

FIGURE 6.1: System Testing Diagram

6.3.1. Function Testing

A function testing checks whether the integrated system performs its function accordingly to the specific requirements. Therefore, use case can be used as a tool for the function testing.

- Use Case in Administration Module

Use Case	Activities Tested	Expected Result	Result
Login	Enter nothing and click button	Cannot be submitted. JavaScript prompted error message.	As Expected
	Enter non-alphabet and non-numeric character.	Cannot be submitted. JavaScript prompted error message.	As Expected
	Enter a correct username and password and click login button	No error message is prompted and login process successful	As Expected
	Enter an incorrect username and password and click login button	Error message is prompted and login process not successful	As Expected
	Enter a correct username but incorrect password and click login button	Error message is prompted and login process not successful	As Expected
Authentication	Do the correct login	Username will appear on the page after login successful	As Expected
Logout	Click logout	No error message and logout is successful	As Expected
	Click previous page or secure page after logout	Page login will come out to prevent un-authorize user	As Expected

TABLE 6.2: Use Case in Administration Module

6.3.2. Performance Testing

After we are convinced that the function is working as specified, the performance test compares the integrated components with the non-functional requirements. These non-functional requirements consist of user-friendliness, correctness, functionality, reliability, flexibility and efficiency.

6.3.3. Acceptance Testing

Acceptance testing is done to make sure that the system meets customer understanding of the requirements, which may be different from the developer. This testing will be done only when the system has been delivered to the customer.

6.3.4. Installation Testing

Installation testing is the stage for the system testing. The installation testing is required for the whole system which is to be delivered to the customer and the system to be installed at the customer site. The installation testing performs the required testing on the integration of the software which had been developed with the hardware at the customer site. The testing is required because we need to make sure that the system is well integrated together with the hardware and performs properly on the customer site.

6.4. CONCLUSION

System testing has been carried out successfully. Testing is integral for ensuring that the system functions properly and meet users' requirement and expectations.

CHAPTER 7:

SYSTEM EVALUATION

7.0. INTRODUCTION

System evaluation will document the outcome that has been finalized (*Soccer365* MMS Application). This chapter will cover the difficulties and problems encountered during the development process from the beginning until the end of the project, the solutions to the problems, the strengths and the limitations of the system being developed, and the future enhancement of the system. Finally, the conclusion will conclude the project as a whole.

The documentation for the system evaluation of *Soccer365* MMS application will tackle the issues as listed below:

- The problems encountered and the solution
- The strengths and limitations of the system
- Future enhancement for the system
- The conclusion of the whole project

7.1. PROBLEMS ENCOUNTERED AND THE SOLUTION APPLIED TO IT

When you are directly involved with the development phase of any particular project, you are bound to encounter some form of difficulty or challenge that may hinder your progress towards your objective. Hence, it is in that similar situation that I managed to place myself in when I first started developing my proposed system. I had to overcome many obstacles, such as to determine what kind of scripting language that I would eventually use for my particular system. To make matters simple, I have listed some of the main problems that have surfaced and the ideal solutions that have been applied to overcome it:

7.1.1. Lack of Experience in Developing Huge System

The main requirement for a developer when developing an overall system is the developer's level of expertise and experience in handling such an enormous task. Initially, I first realized I had encountered some difficulty was when I had to define the project scope of the proposed system and how to ascertain the logic flow of the system. Furthermore, gathering basic information solely through the

Internet and from reference books does not guarantee you adequate information about the inner workings of MMS development. Therefore, in order to develop an overall system, we need to gather and acquire information from other means of sources in order to fully understand and grasp the real concept of how a fully-developed system actually runs in the real environment.

Through improvisation, I visited many on-line forums located throughout the Internet to find out additional information on how to develop a suitable infrastructure or framework for my MMS applications. In addition, I have also consulted my colleagues and seek their feedback on the proposed system that I intend to develop.

7.1.2. Lack of Experience in Programming Languages

As we can see in this particular system, there are many programming languages that have been used in the implementation of the system. Despite the various programming languages that have been coded into the proposed system, we had to make sure that they run perfectly for the development of *Soccer365* MMS application, especially on the client side

In Soccer365 MMS application, I have implemented the following programming languages:

- HTML (Hyper Text Mark-Up Language)
- ASP (Active Server Page)
- JavaScript
- VBScript

For those who are particularly familiar with all the programming languages that I have listed above, they may understand the pros and cons in using these programming languages. There are many aspects in which we have to consider and scrutinize. For example, we need to check the syntax, examine it carefully with the variable and most importantly we must understand and have an in-depth knowledge of the programming languages that is being used.

I was able to overcome this problem simply because I have done some research on the basics necessities on the usage of the particular programming languages at hand. In addition, I have also carried out series of discussions with my project partner and colleagues on the usage of these programming languages and how it best fits into my system.

7.1.3. Lack of Technique in Designing Interface

This sub-system is designed specifically for the client side (user side). By not possessing the right skills and techniques, designing an interactive interface may prove to be a daunting task. I have to design a compelling and eye-catching web page so that prospected customers are instantly attracted and encouraged to visit my web page and to see for themselves, the services that I have to offer.

7.1.4. Time Constraint

I faced major time constraint to complete my project on time as I also need to concentrate on my final year examinations which will be around the corner. However, I believe that it is due to lack of experience rather than time constraint, which has prompted me to come to the decision of reserving more time for the development of my proposed system

Subsequently, I had to adjust and reschedule my time so that I am able to focus more attention on the development of my system. There was a particular period when I had to switch my time of between developing the system and preparing for the mid-term examinations. Only during the mid-semester holidays was I able to fully concentrate and increase my efforts on the development of the system as compared to the normal week days.

7.2. STRENGTHS

There are some advantages or strengths that can be found in my proposed system as compared to other existing systems. There are:

7.2.1. Interesting Interfaces

This particular system has an attractive interface design. It has been carefully designed so that users are compelled to visit the particular web page straight away. The menu functions are very clear and easy for the users to find and manage.

Overall, the interface of this system is good. It will definitely create a good impression on first-time users. The consistent design of each page will make the navigation a stroll in the park. Hyperlinks in this web page are arranged strategically and accordingly based on its purposes. These arrangements are provided to facilitate supportive browsing.

7.2.2. Free Download

Easily the most unique feature of the *SOCCER365* MMS application. Users are able to download many soccer-related animation, video, and wallpaper contents by the click of a mouse. Just press send on the emulator and then users are able enjoy their downloaded mobile content totally free of change. This concept is very similar to a user who downloads free ring tones, screensaver and wallpaper images using mobiles phones or PDA.

7.2.3 User Friendly and Easy to understand

The functions that I have provided in this system are relatively easy to understand. It is carefully thought of in such a way that all it facilitates user browsing experience. This will inadvertently broaden the user's understanding of this particular system and make navigation plain simple. Hence, users are able to download soccer-related mobile contents such as animation, video and wallpaper related without breaking a sweat.

7.2.4 Easy Administration Modules

This system consists of an easy-to-manage administration module. With this module, the administrator can easily add, edit and upload soccer-related mobile contents. They can also find out or add information about certain cars and their models.

7.3. LIMITATIONS

7.3.1. Security

The password is vulnerable because it is not posted in encrypted format from the login form or when it is stored in the system database therefore, it is susceptible to attacks from hackers and crackers.

7.3.2. No Rules and Regulation

There are no rules and regulation applied to the proposed system. Anyone who possesses a mobile phone that supports WAP and GPRS can easily have access and attain the mobile contents within the system.

7.3.3. No Database Backup

There is no database backup service provided for this particular system. This may reduce the integrity and the reliability of the *Soccer365* MMS application in the long run when the database gets corrupted.

7.3.4. Limited Download Feature

There is limited download feature. User are only able to download animation, video and wallpaper contents only.

7.4. FUTURE ENHANCEMENT

In this project, we have able to fulfill the user requirements and the functionalities of the system, but it can be further enhanced and become a more stable system. The following are some forms of enhancements that can be done to the system in the future:

7.4.1. Increase the Security

Facilitate the encryption and decryption process of sensitive data such as password. Password should be encrypted as it is send to server and stored in the database.

7.4.2. Database Backup

Backup the database periodically to increase the reliability and integrity of the data stored in the *SOCCER365* MMS application database.

7.4.3. Add More Download Features

In the future, we could provide more download features such as ring tones, news update and many more with regard to soccer.

7.5. PROJECT CONCLUSION

After strenuous months of developing the *SOCCER365* MMS application, I would say that most of the system's initial objectives have been met. The system requirements which clearly state the importance of such aspects such as functional requirement, non-functional requirement, software requirement and hardware requirement and also the methodology used have been fulfilled.

The development of this particular system has given me countless benefits and provided me with many learning curves. I've gained invaluable experience and acquire new knowledge with regard to system development. Furthermore, I've learned to use many tools that I haven't had the opportunity to use before such as Macromedia Dreamweaver MX. Moreover, this particular project has provided me with a useful platform to harness and sharpen my programming skills and learn many ways to configure different types of supporting tools. For example, Nokia Mobile Internet Toolkit and the Series 60 SDK. In essence, the important lessons which I have learnt and gathered over these past few months will definitely enrich and prepare me for the challenging future ahead.

Although the system has been developed successfully, limitations still exists partly due to the timing constraint imposed during the development phase. I truly believe that these limitations could be overcome if future enhancements were to be made.

REFERENCES:

1. Ian, Sommerville. (2001). Software Engineering. 6th ed. Addison Wesley.
2. Dr. P Sellappan (2000) Management and Method: Software Engineering. Sejana Publishing.
3. Christoffer, Andersson. (2001). GPRS and wireless applications: Professional Developer's Guide. John Wiley & Sons.
4. Aneesha Bakharia, Premier Press: Macromedia Dreamweaver MX 2004.
5. *Creating Database Web Application with PHP and ASP* / Jeanine Meyer. *1st ed.* Hingham, Mass : Charles River Media, c2003

WEBSITE REFERENCES:

1. www.whatis.com
2. www.forum.nokia.com
3. http://www.ideas.singtel.com
4. http://www.t-mobile.com
5. http://www.nttdocomo.com
6. http://java.sun.com
7. http://www.hotscripts.com
8. http://www.palowireless.com
9. http://www.slackerhtml.com
10. http://www.wirelessdevnet.com

APPENDIX A:

INSTALLATION GUIDELINE

1. HARDWARE AND SOFTWARE REQUIREMENT:

i. Hardware requirement:

- Any WAP enabled Mobile Phone with MMS Services
- 500 MHz Intel Pentium and above.
- 256 MB or more.
- 65,000 color or better display card.
- Operating system Windows 2000 above, and Windows XP.

ii. Software requirement:

- Internet Information Service (IIS) 7.0 - Web server
- Microsoft Access 2003 - Database

- Internet Explorer 6.0 - Web browser
- Microsoft FrontPage 2003 – 1st code and interface editor
- Macromedia Dreamweaver MX – 2nd code and interface editor
- Adobe Photoshop 7.0 - Image Editor
- Microsoft Word 2003 - Thesis Documentation
- ASP

2. GUIDELINE:

i. Install Nokia Mobile Internet Toolkit

**** You have to be a registered user to download this application***

- Download Nokia Mobile Internet Toolkit.

  http://www.forum.nokia.com

- Extract ***NMIT_41*** to a local directory and run install.exe
- Key in your forum Nokia user's ID and the serial number of the application. You can obtain the serial number by requesting it from Nokia.
- After the installation process is completed, go to ***Start >All Program > Nokia Developer Tool s> Nokia Mobile Internet Toolkit > NMIT 4.1*** to run the application.

ii. Install Series 60 Content Authoring SDK 1.0.6

- Download Series 60 Content Authoring SDK 1.0.6

  http://www.forum.nokia.com

- Run S60v106_Install and install it in your local directory.

iii. Send MMS messages

Nokia Mobile Internet Toolkit,

- Click on ***File → New → Messaging → MMS Message***.
- Add your parts list at the bottom-left corner of the screen and then click on '***Push***'. Before you do this, make sure that your ***Series 60 Content Authoring SDK 1.0.6*** is running. On the other hand, you can also initiate the ***Series 60 Content Authoring SDK 1.0.6*** through the NMIT's '***SDK Panel***'.
- Your Series 60 Content Authoring SDK 1.0.6 will then receive the content that you have just sent.

iv. Open Soccer365 MMS application

- Make sure your files are saved at your local server.

Example: C:\Inetpub\wwwroot\Soccer365

- To access the Soccer365.com main page, key in the following address in the URL

Example: *http://localhost/Soccer365*

APPENDIX B:

SOURCE CODES

1. HTML (Hyper Text Markup Language)

Main Page - constructed by using inline frames. Three files were combined to form the main page. Those files are default.asp, default_menu.asp and mobile1.asp. Here, I will only highlight the essential codes that were needed to achieve this.

Default.asp

```
<table width="778" cellpadding="0" cellspacing="0">
<tr><td align="center">

<!--main page -->

<table width="784" cellpadding="0" cellspacing="0" align="center">
```

```
<tr><td width="782">
<table width="100%" cellspacing="0" cellpadding="0">
<tr>
<!--logo and nav-->
<td width="453" height="159" align="left">
<table width="85%" cellpadding="0" cellspacing="0" bordercolor="#FFFFFF">
<tr>
 <td width="4">  </td>
 <td colspan="3"><img src="images/banner.gif" width="360" height="72"></td>
 </tr>
<tr><td> </td>
 <td width="126" bordercolor="#FFFFFF"><div align="center">
      <span style="text-transform: uppercase">
      <font color="#FFFFFF" size="4" face="Verdana">
      <a target="main" href="animation.asp">
      <span style="text-decoration: none"><font
color="#FFFFFF">animation</font></span></a></font></span></div></td>
 <td width="127" bordercolor="#FFFFFF"><div align="center">
  <span style="text-transform: uppercase">
      <font color="#FFFFFF" size="4" face="Verdana">
      <a href="video.asp">
      <span style="text-decoration: none"><font
color="#FFFFFF">video</font></span></a></font></span></div></td>
```

```
<td width="126"><div align="center"><span style="text-transform: uppercase">
        <font color="#FFFFFF" size="4" face="Verdana">
        <a href="wallpaper.asp">
        <span style="text-decoration: none"><font
color="#FFFFFF">wallpaper</font></span></a></font></span></div></td>
 </tr>
</table></td>
```

Default_menu.asp

```
<table width=140 id="table2"><tr><td>
<script language="JavaScript">
var Link = new Array();
Link[0] = "0|Info";
Link[1] = "1|Club|http://champions.fantasy-
manager.co.uk/stats.aspx?page=teamSearch|";
Link[2] = "1|Player|http://champions.fantasy-
manager.co.uk/stats.aspx?page=playerSearch|";
Link[3] = "0|Access|http://|";
Link[4] = "1|Admin|login.asp|";
Link[5] = "0|Back To Main";
Link[6] = "1|Back To Main|mobile1.asp|";
startup(0);
```

```
</script>
</td> </tr> </table>
```

Mobile1.asp

```
<table width="566" cellpadding="2" cellspacing="2" border="0" id="table4">
<tr><td colspan="3" align="left" height="10"></td>
</tr>
<tr><td colspan="3" align="left"><p class="white style1 style6"><b>
<font color="#FFFFFF" face="Verdana">Some of the WALLPAPERS on
offer:</font></b> </p></td>
</tr><tr>
<td colspan="3" align="left" height="8"></td>
</tr><tr>
<td width="182" align="center" valign="top" height="146"><img
src="images/anti_leeds.gif" width="177" height="142"></td>

<td width="179" align="center" valign="top" height="146"><img src="images/dart.gif"
width="177" height="142"></td>

<td width="186" align="center" valign="top" height="146"><img
src="images/Mickey_Mouse.gif" width="177" height="142"></td>
</tr>
</table>
//-----------------------------------------------------------------------------------
```

```
<table width="559" cellpadding="2" cellspacing="2" border="0" id="table6">
<tr><td colspan="4" align="left" height="25"><p class="white style6 style1"><b>
        <font face="Verdana" color="#FFFFFF">Some of the
        ANIMATIONS on offer: </font></b> </p></td>
</tr>

<tr><td colspan="4" align="left" height="8"></td>
</tr><tr>
<td width="0" align="center" valign="top"><br></td>
<td width="205" align="center" valign="top"><img src="images/sf.gif" width="148"
height="144"></td>
<td width="184" align="center" valign="top"><img src="images/soccer05.gif"
width="181" height="148"></td>
<td width="139" align="center" valign="top"><img src="images/Trophee3.gif"
width="69" height="123"></td>
</tr>
</table>
//-----------------------------------------------------------------------------------------
<table width="93%" cellpadding="2" cellspacing="2" id="table7">
 <tr>
  <td width="98%" align="center" valign="top">
      <table width="573" cellpadding="2" cellspacing="2" border="0" id="table8">
    <tr>
      <td colspan="2" align="left"><p class="white style1 style6"><b>
```

```
                    <font face="Verdana" color="#FFFFFF">Some of the VIDEOS
on offer:</font></b> </p></td>
    </tr>
    <tr>
      <td colspan="2" align="left" height="13"></td>
    </tr>
    <tr>
      <td width="279" align="center" valign="top" height="245"><font
face="Verdana">
<object id="objMovie6" classid="clsid:CFCDAA03-8BE4-11cf-B84B-
0020AFBBCCFA" height="210" width="240">
<param name="controls" value="ImageWindow,ControlPanel">
<param name="autostart" value="-1">
<param name="SRC" value="file://C:\Inetpub\wwwroot\mySoccer\video\ronaldo.3gp"
ref>
<param name="SHUFFLE" value="0">
<param name="PREFETCH" value="0">
<param name="NOLABELS" value="0">
<param name="LOOP" value="0">
<param name="NUMLOOP" value="0">
<param name="CENTER" value="0">
<param name="MAINTAINASPECT" value="0">
<param name="BACKGROUNDCOLOR" value="#000000">
```

```
<embed src="http://play.rbn.com/?url=cspan/g2cspan/live/cspan1-
g2.rm&proto=rtsp&plugin=1?embed" type="audio/x-pn-realaudio-plugin" height="180"
width="240" controls="ImageWindow" console="_master" autostart="true"
name="objMovie6"><br><embed
src="http://play.rbn.com/?url=cspan/g2cspan/live/cspan1-
g2.rm&proto=rtsp&plugin=1?embed" type="audio/x-pn-realaudio-plugin" height="30"
width="240" controls="ControlPanel" console="_master" autostart="true">
</object>

</font></td>
    <td width="280" align="center" valign="top" height="245"><font
face="Verdana">
<object id="objMovie4" classid="clsid:CFCDAA03-8BE4-11cf-B84B-
0020AFBBCCFA" height="210" width="240">
<param name="controls" value="ImageWindow,ControlPanel">
<param name="autostart" value="-1">
<param name="SRC" value="file://C:\Inetpub\wwwroot\mySoccer\video\rivaldo.3gp"
ref>
<param name="SHUFFLE" value="0">
<param name="PREFETCH" value="0">
<param name="NOLABELS" value="0">
<param name="LOOP" value="0">
<param name="NUMLOOP" value="0">
<param name="CENTER" value="0">
```

```
<param name="MAINTAINASPECT" value="0">
<param name="BACKGROUNDCOLOR" value="#000000">
<embed src="http://play.rbn.com/?url=cspan/g2cspan/live/cspan1-g2.rm&proto=rtsp&plugin=1?embed" type="audio/x-pn-realaudio-plugin" height="180" width="240" controls="ImageWindow" console="_master" autostart="true" name="objMovie4"><br><embed src="http://play.rbn.com/?url=cspan/g2cspan/live/cspan1-g2.rm&proto=rtsp&plugin=1?embed" type="audio/x-pn-realaudio-plugin" height="30" width="240" controls="ControlPanel" console="_master" autostart="true">
</object>
</font>
</td></tr>
</table>
</td></tr>
</table>
```

2. SQL SCRIPT

Connection

```
<%
 Set Conn = Server.CreateObject("ADODB.Connection")
 Connstring = "DRIVER={Microsoft Access Driver (*.mdb)}; "
 Connstring = Connstring & "DBQ=" & Server.MapPath("db\mms.mdb")
 Conn.Open(Connstring)

%>
```

User Session

```
<%Dim Apples
Set Apples = Server.CreateObject("ADODB.Connection")
ConnStr = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=" & Server.MapPath("db\mms.mdb") & ";" & _
        "Persist Security Info=False"

Apples.Open(ConnStr)

SQLtemp = "SELECT * FROM akaun_pengguna WHERE name = '" &
Request.form("login") & "' "

Set rs =Server.CreateObject("ADODB.Recordset")
rs.Open SQLtemp,Apples,3,3
while not rs.eof

dim username
username = rs("name")

response.cookies("passes") = username

        If      Request.Form("login")       =       trim(rs("name"))        AND
Request.Form("password") = trim(rs("password")) Then
                Session.Contents("tahap") = rs("tahap")
                Session.Contents("id") = rs("no_pengguna")
                Session("allow") = True
                Response.redirect("response.asp")
   Else

        Response.redirect("login.asp?msg="&"Login+Has+Been+Unsuccesful!")
                'Response.redirect("login.asp")
        End If

rs.MoveNext
Wend
OnError
Response.redirect("login.asp?msg="&"Login+Has+Been+Unsuccesful!")
'OnError Response.redirect("login.asp")
rs.Close
Apples.Close
set Apples = Nothing
%>
```

Get Data From Database

```
<%
Set Conn = Server.CreateObject("ADODB.Connection")
Connstring = "DRIVER={Microsoft Access Driver (*.mdb)}; "
Connstring = Connstring & "DBQ=" & Server.MapPath("db\MMS.mdb")
Conn.Open(Connstring)

Set rs = Server.CreateObject("ADODB.Recordset")
sql = "Select * From [Wallpapers] where [dataid] = " &request.querystring("id")
& " "
rs.Open sql, Conn, 3, 3

  If Not rs.Eof Then
        dataid=rs.Fields("dataid")
        data1=rs.Fields("code")
        data2=rs.Fields("info")
        data3=rs.Fields("url")
        'data4=rs.Fields("data4")
     End If
%>
```

3. JAVA Scripts

Alert Message


```
<script language="JavaScript">
<!--
function check(){

var err_msg, msgstring, err;

 err_msg = "Enter : ";
```

```
msgstring = "";
err = false;

        if (document.form1.elements.username.value == "" )
        {
                err = true;
                msgstring = "Username";
        }

        if (document.form1.elements.password.value == "" )
       {

                if ( err == false )
                {
                    err = true;
                   msgstring = "password";
                }
          Else
                {
                msgstring = msgstring + " ,Password ";
                }
        }



if ( err == true )
{
        alert(err_msg + msgstring);

   }else
        {

        var errors="";
        if (errors!="") {
        alert(errors);
        return false;
        }

        Else
        {

document.form1.action = "login1.asp";
document.form1.submit();
}
}
}
function back(){
```

```
        history.back();
}
//-->
</script>
```

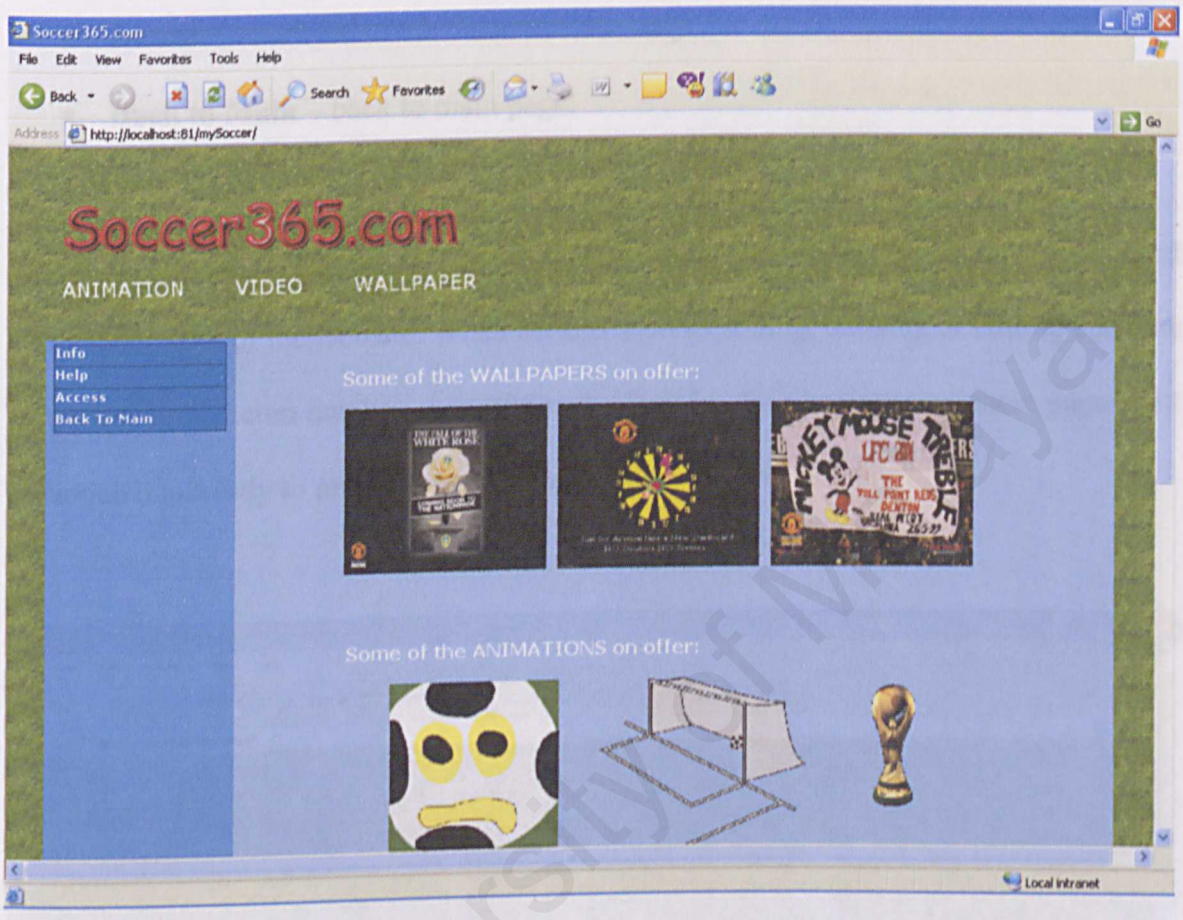

Figure 1: Soccer365.com Main Page

Description of Figure 1:

On the left side column of the page, there is a drop down list where users are able to access links to particular pages within the website. These links consists of:

- **Info** – which can be divided into two sub-menus; ***Club*** and ***Player*** info, that link to a particular soccer website.

  Club enlists the names of famous soccer clubs that exists within the soccer world.

On the other hand, ***Player*** entails details on famous soccer players that ply their trade with their respective clubs.

- **Access** – restricted to administrator use only.
- **Back to Main** – back to main page.

Below the Soccer365.com banner are three separate links to animation, video and wallpaper pages of the website. When actions such as adding, deleting or editing contents of the Soccer365.com database is performed, albeit by the administrator, these pages will be updated instantly to provide users with the latest mobile contents.

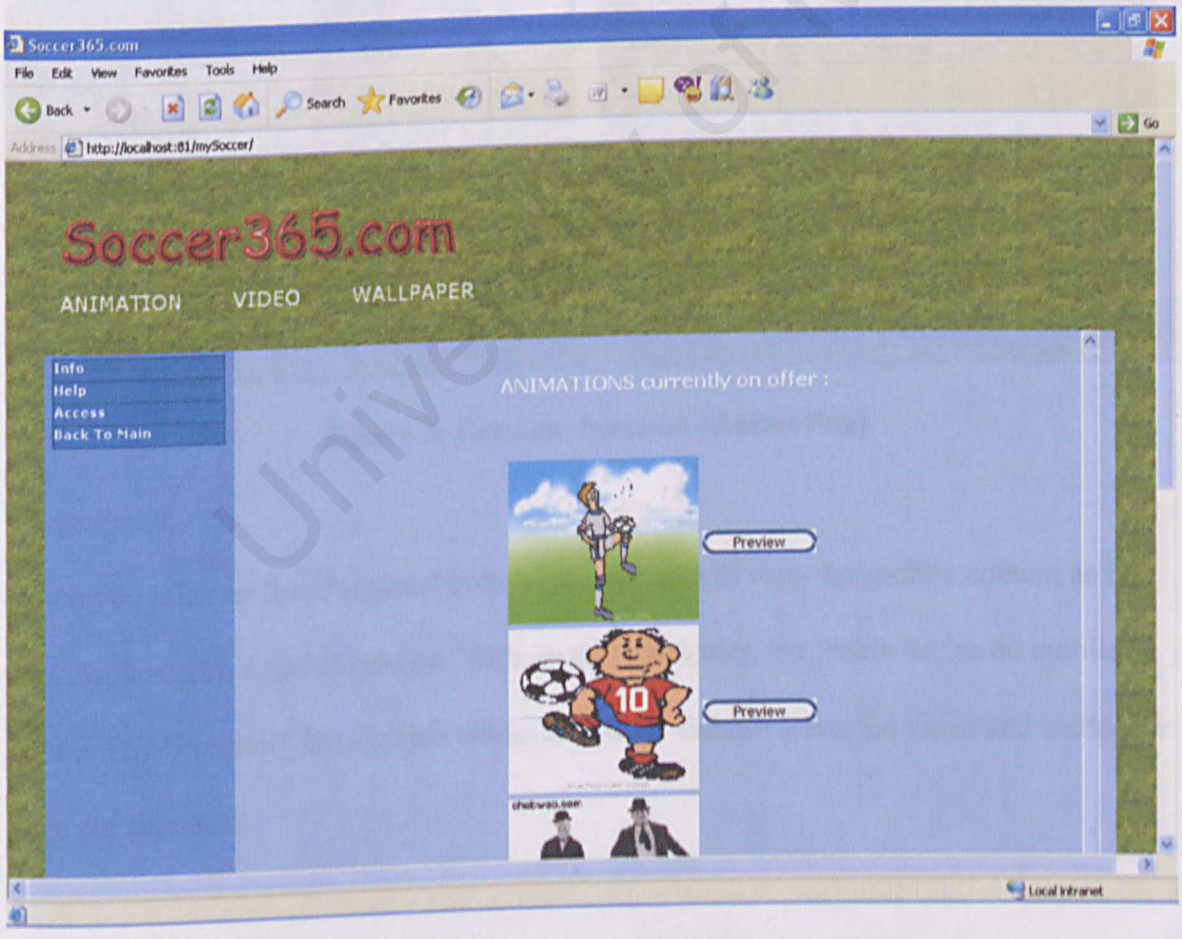

Figure 2: Animation Page

The animation page consists of all animation mobile contents that are stored within the database. These contents are automatically generated form the database.

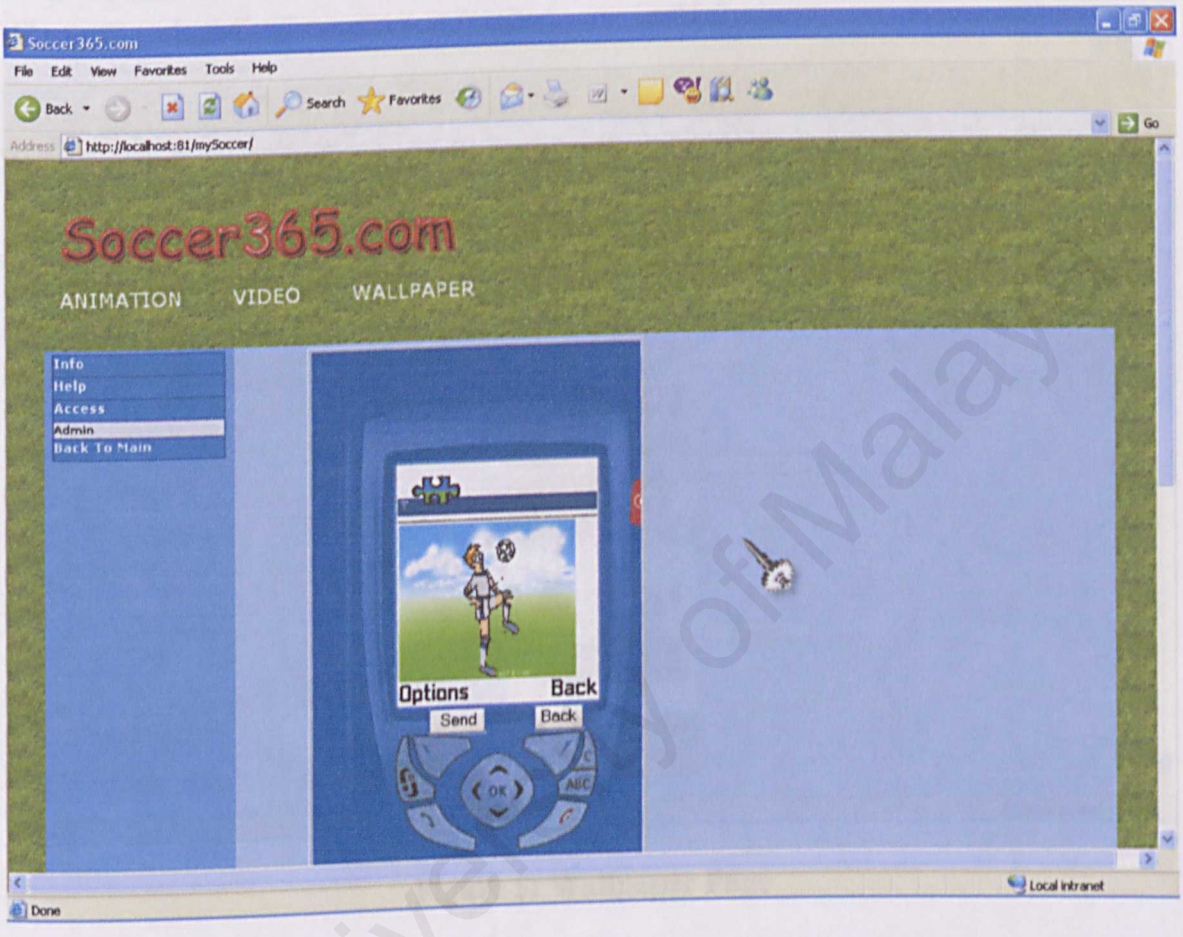

Figure 3: Preview function (Animation)

When you click on the 'Preview' button, you are able to view the mobile content as if it was displayed on a mobile phone. Well, in this particular, the Nokia Series 60 mobile phone. The 'Preview" button also works in similar fashion when the video and wallpaper page are accessed.

Figure 4: Video Page

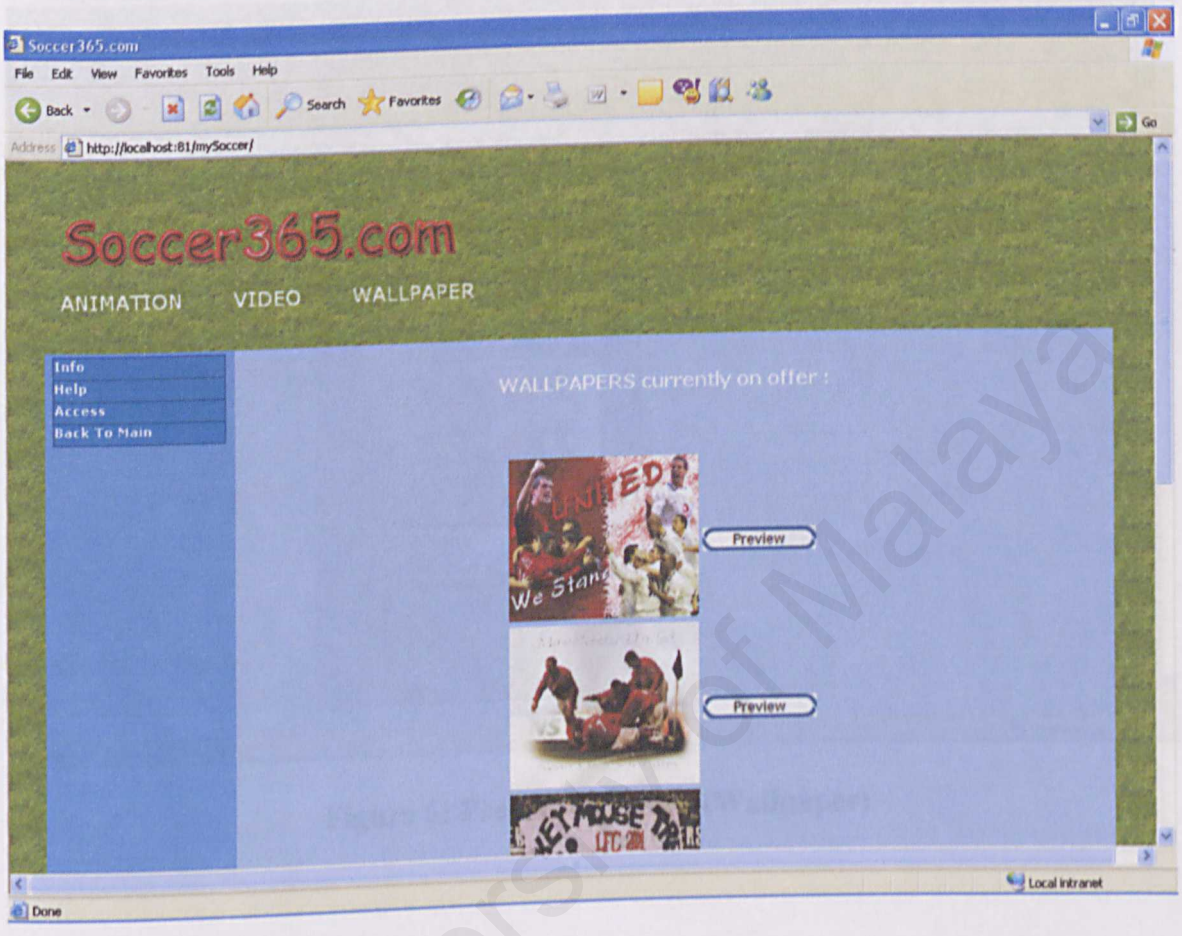

Figure 5: Wallpaper Page

Figure 6: Preview function (Wallpaper)

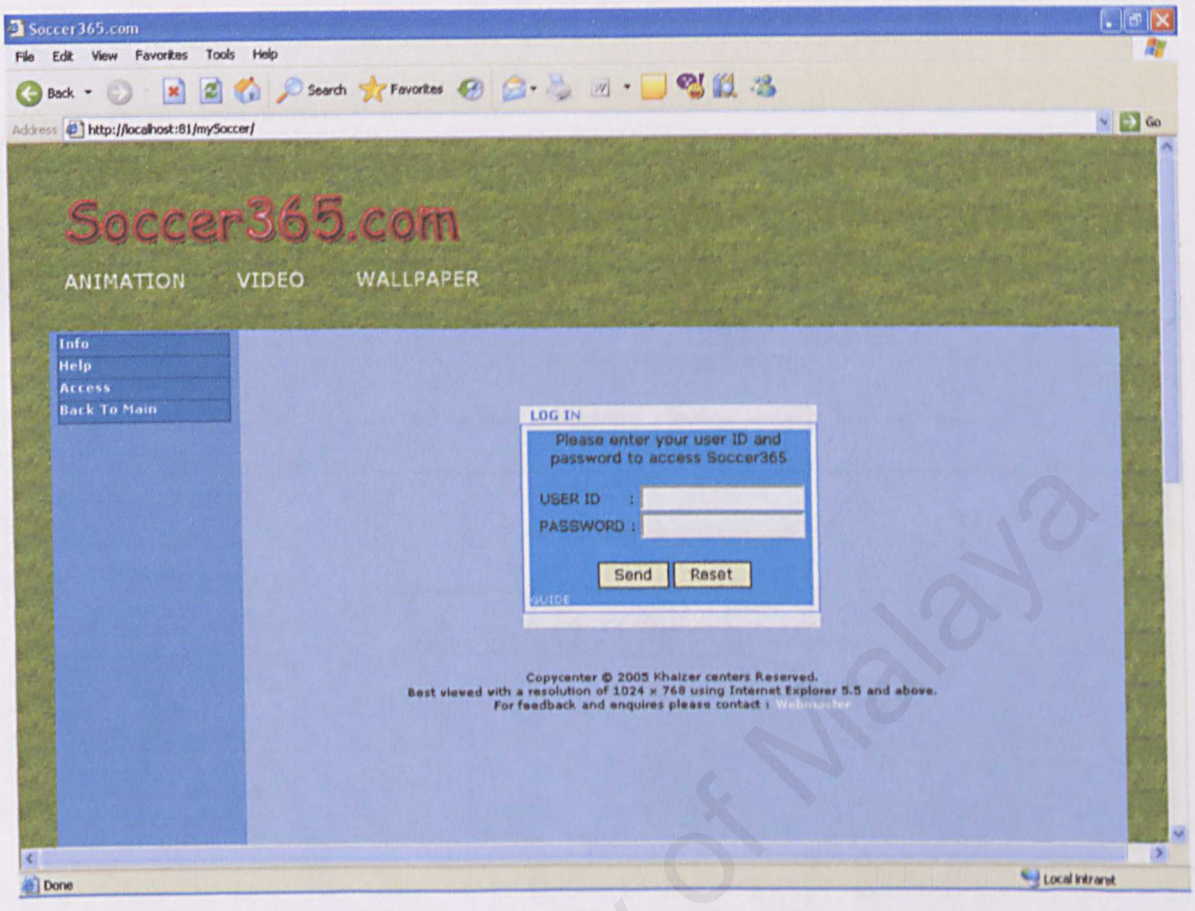

FIGURE 7: Administrator Login

Description of Figure 2:

- Enter User ID and password.
- The password must be the valid.
- Click log in to enter the administrator page.
- Example :

  User ID: ***admin***

  Password: ***admin***

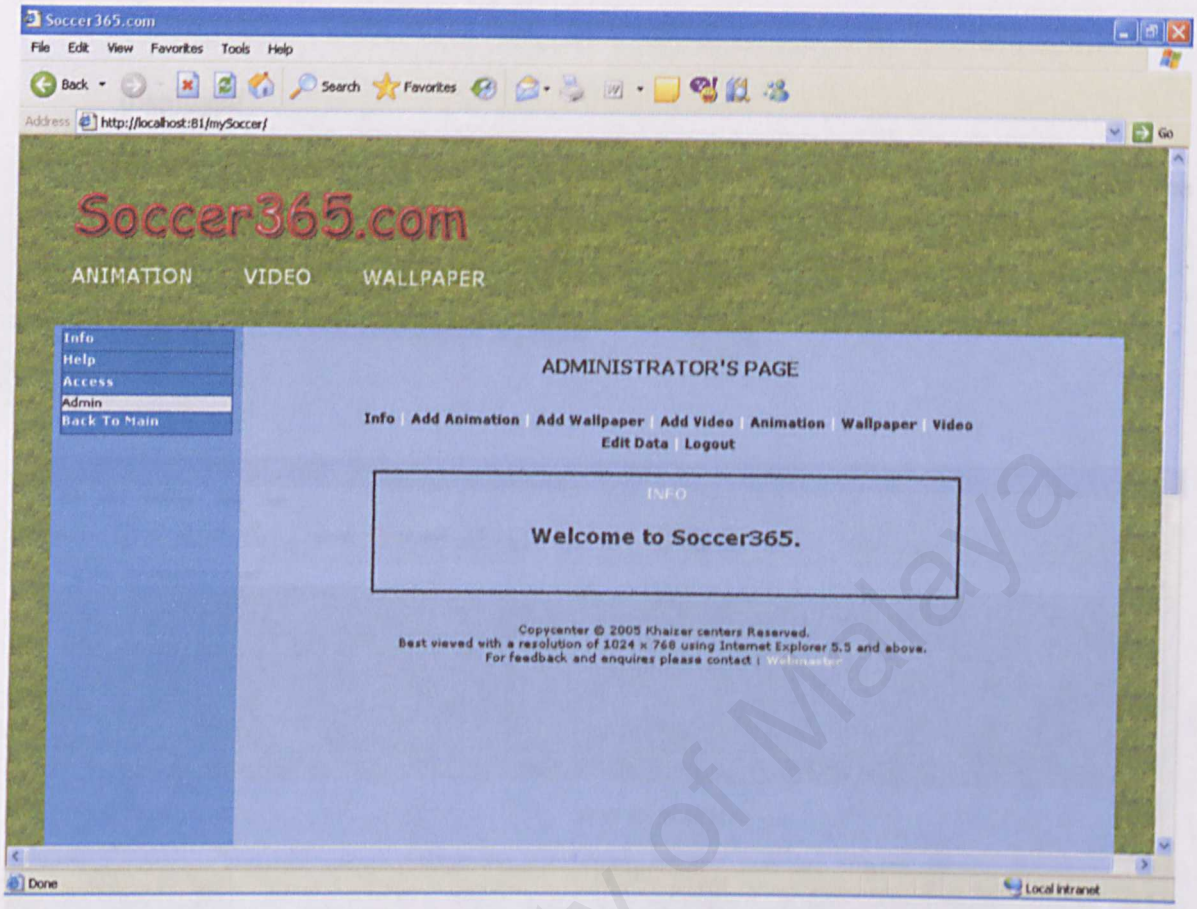

Figure 8: Administrator's Page

Figure 3 entails information on as follows:

1. **Add animation** – Add animation data to the database.
2. **Add wallpaper** – Add wallpaper data to the database.
3. **Add video** – Add video data to the database.
4. **Animation table** – View animation table. Information such as data ID, code, and info are displayed.
5. **Wallpaper table** – View wallpaper table. Information such as data ID, code, and info are displayed.

6. **Video table** – View video table. Information such as data ID, code, and info are displayed.
7. **Edit data** – Perform editing and deleting actions to current data that resides in the database.
8. **Logout** – Exit administrator's page.

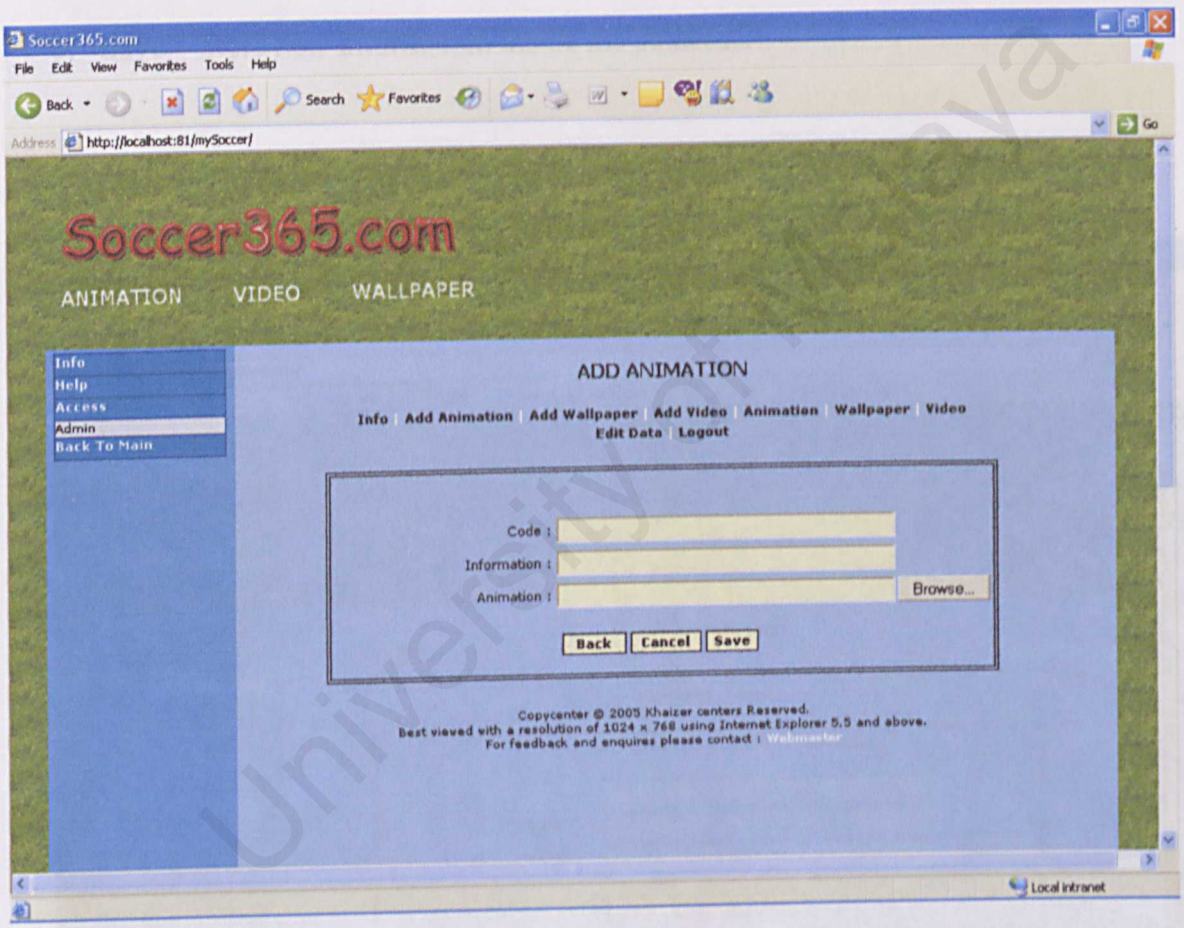

Figure 9: Add Animation

Figure 4 entails information on as follows:

1. **Code** – Enter code for data identification.

2. **Information** – Enter the name you intend to associate the animation with. E.g. Boy Kicking Ball.
3. **Browse** - Choose the respective animation you want to add into the database. E.g. Boy_kicking_ball.gif
4. **Back** – Go to previous page that you have visited.
5. **Cancel** – Abort.
6. **Save** – Store new information inside the database.

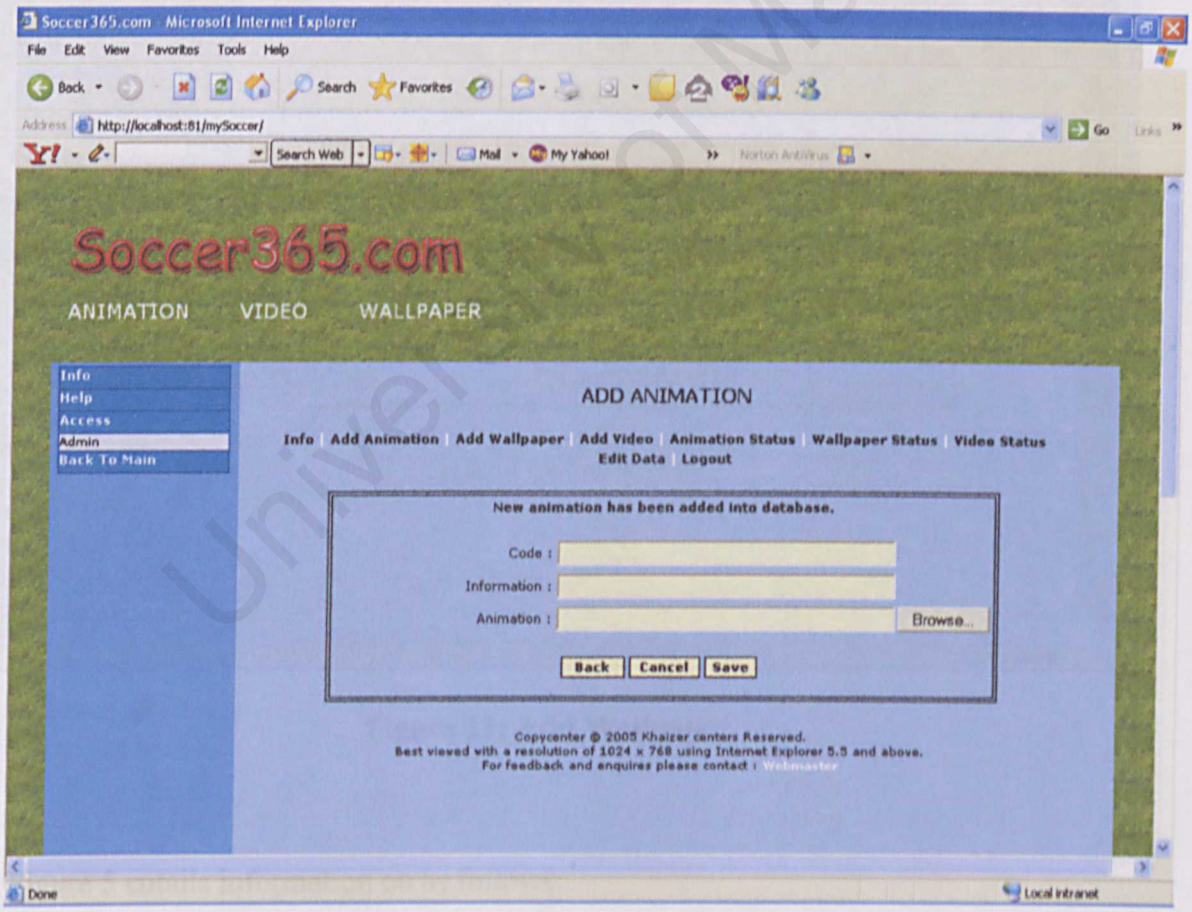

Figure 10: Animation content has been added into database

After you have added your preferred animation content, a message will appear on the screen stating that your animation content has been stored into the Soccer365.com database. A similar message will also appear when you add wallpaper and video contents into the database.

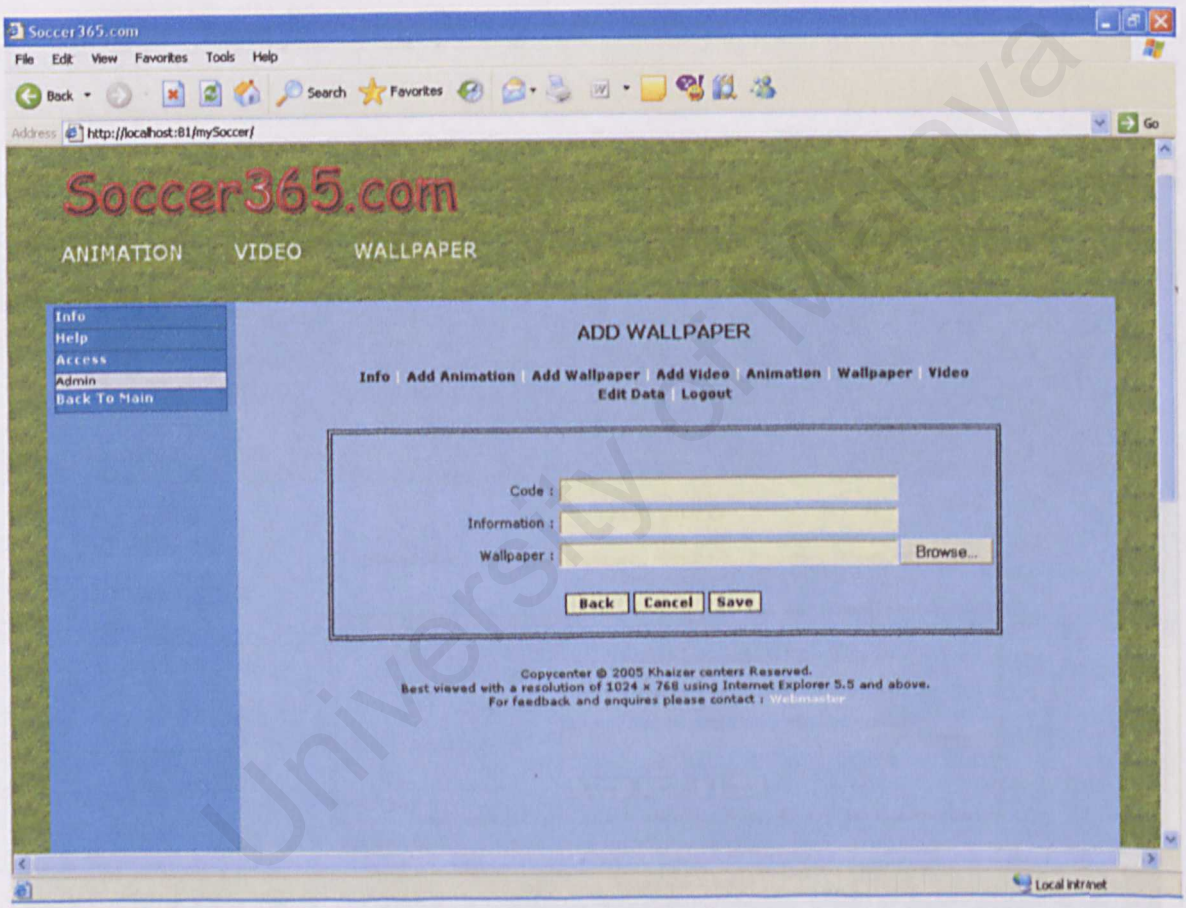

Figure 11: Add Wallpaper

Figure 5 entails information on as follows:

1. **Code** – Enter code for data identification.

2. **Information** – Enter the name you intend to associate the wallpaper with. E.g. Cristiano Ronaldo.
3. **Browse** - Choose the respective wallpaper you want to add into the database. E.g. Cristiano_Ronaldo.jpeg
4. **Back** – Go to previous page that you have visited.
5. **Cancel** – Abort.
6. **Save** – Store your wallpaper inside the database.

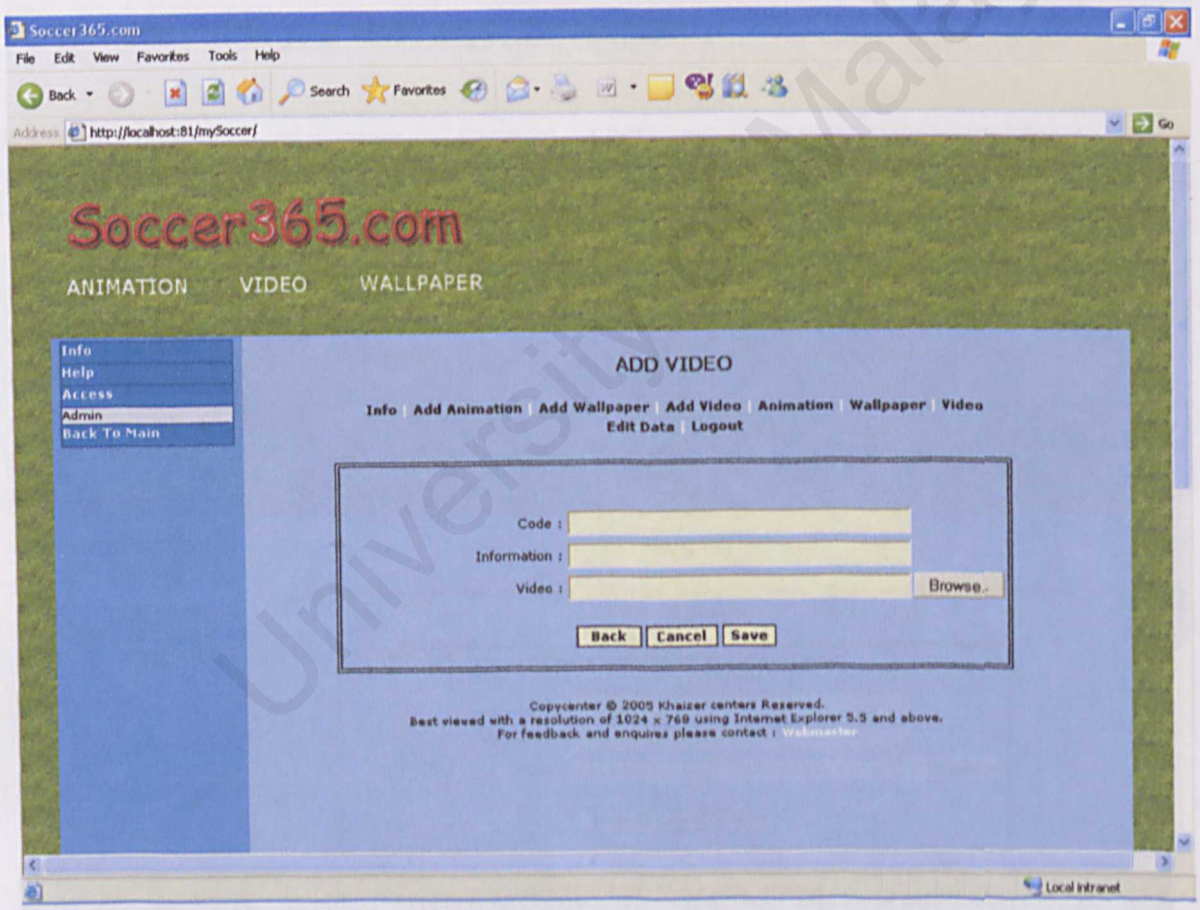

Figure 12: Add Video

Figure 6 entails information on as follows:

7. **Code** – Enter code for data identification.
8. **Information** – Enter the name you intend to associate the video with. E.g. Ronaldo scores.
9. **Browse** - Choose the respective video you want to add into the database. E.g. Ronaldo.3gpp ***(3gpp format supported only)***
10. **Back** – Go back to previous page that you have visited.
11. **Cancel** – Abort.
12. **Save** – Store your video inside the database.

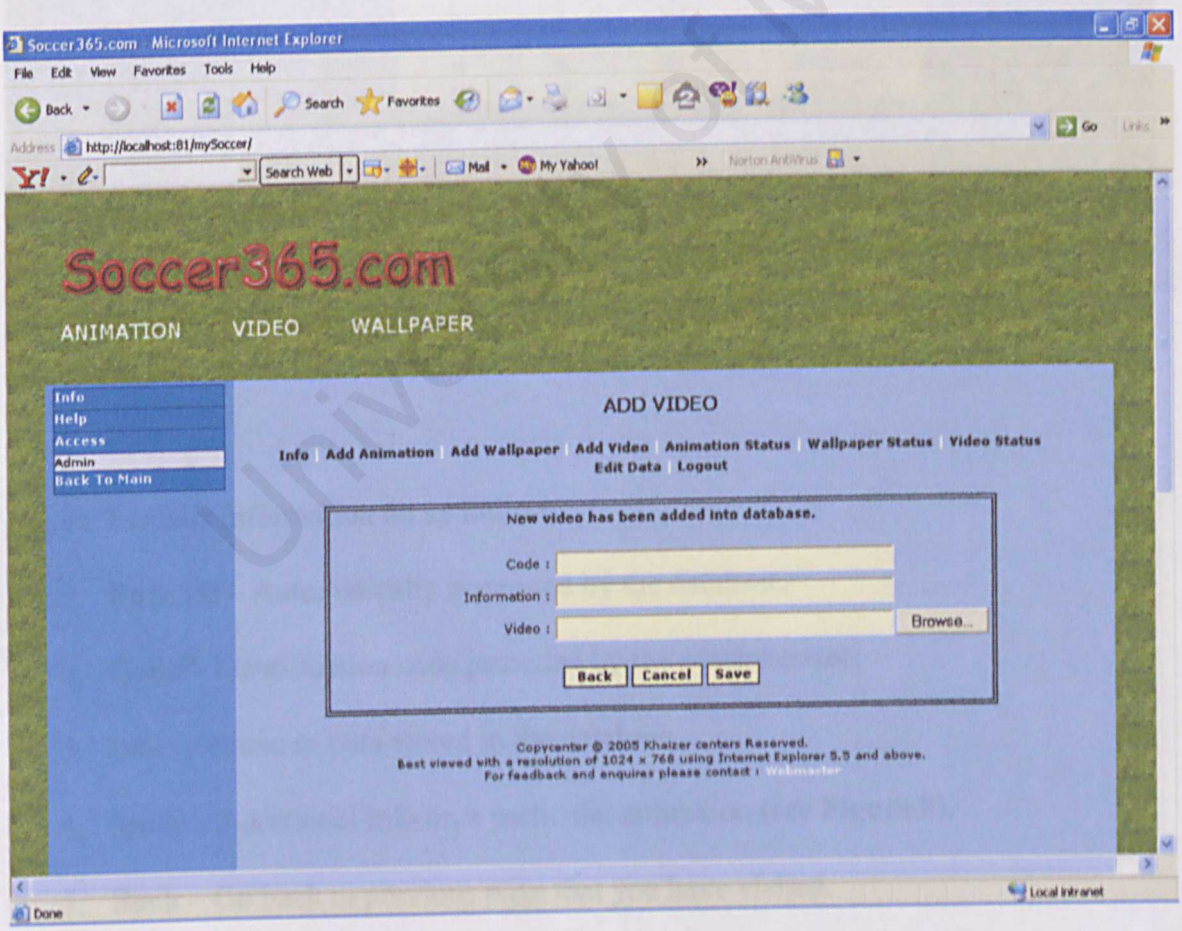

Figure 13: Video content has been added into the database

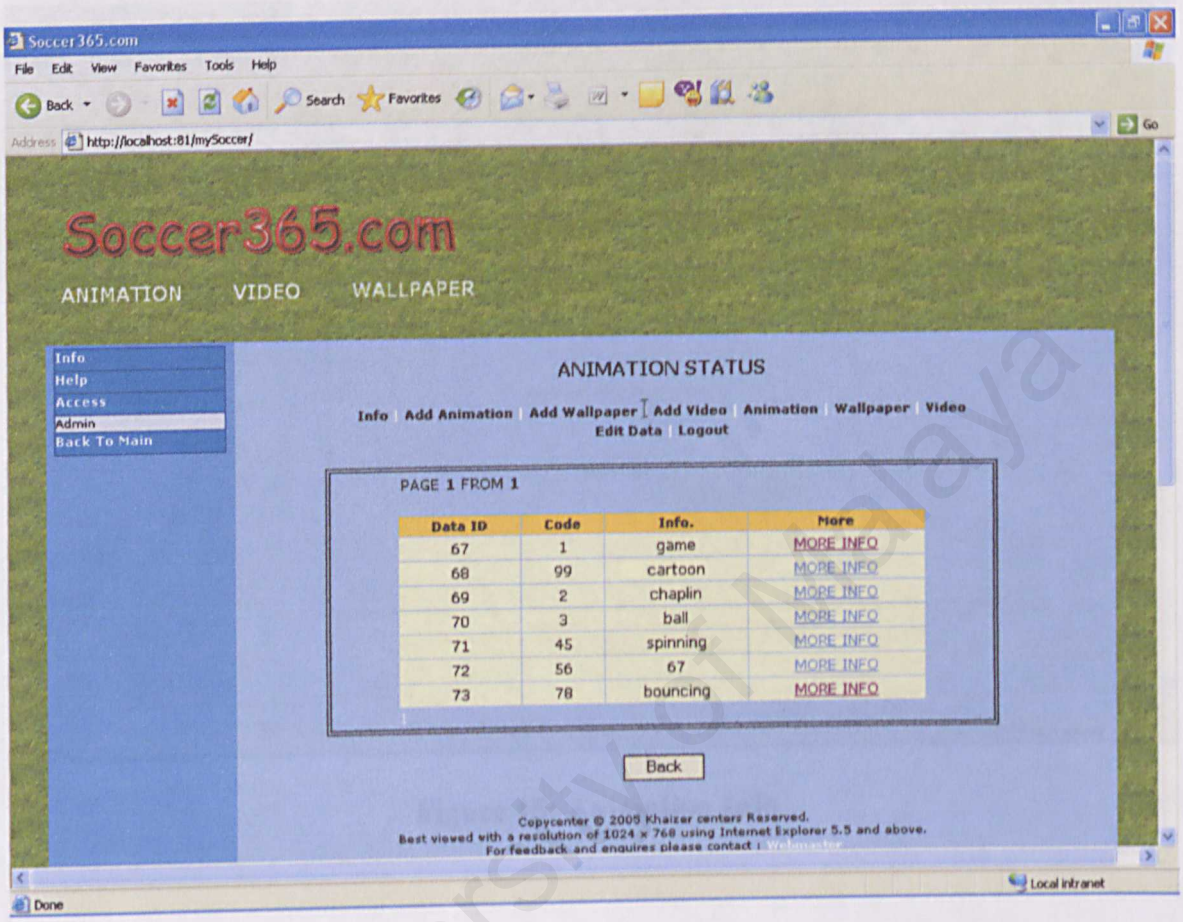

Figure 14: Animation Table

Figure 7 entails information on as follows:

1. **Data ID** – Automatically generated by the database.
2. **Code** – Identification code provided by the administrator.
3. **Info.** – Name of data stored in the database.
4. **More** – Additional info on a particular animation **(see Figure 8).**
5. **Back** – Go back to previous page that you have visited.

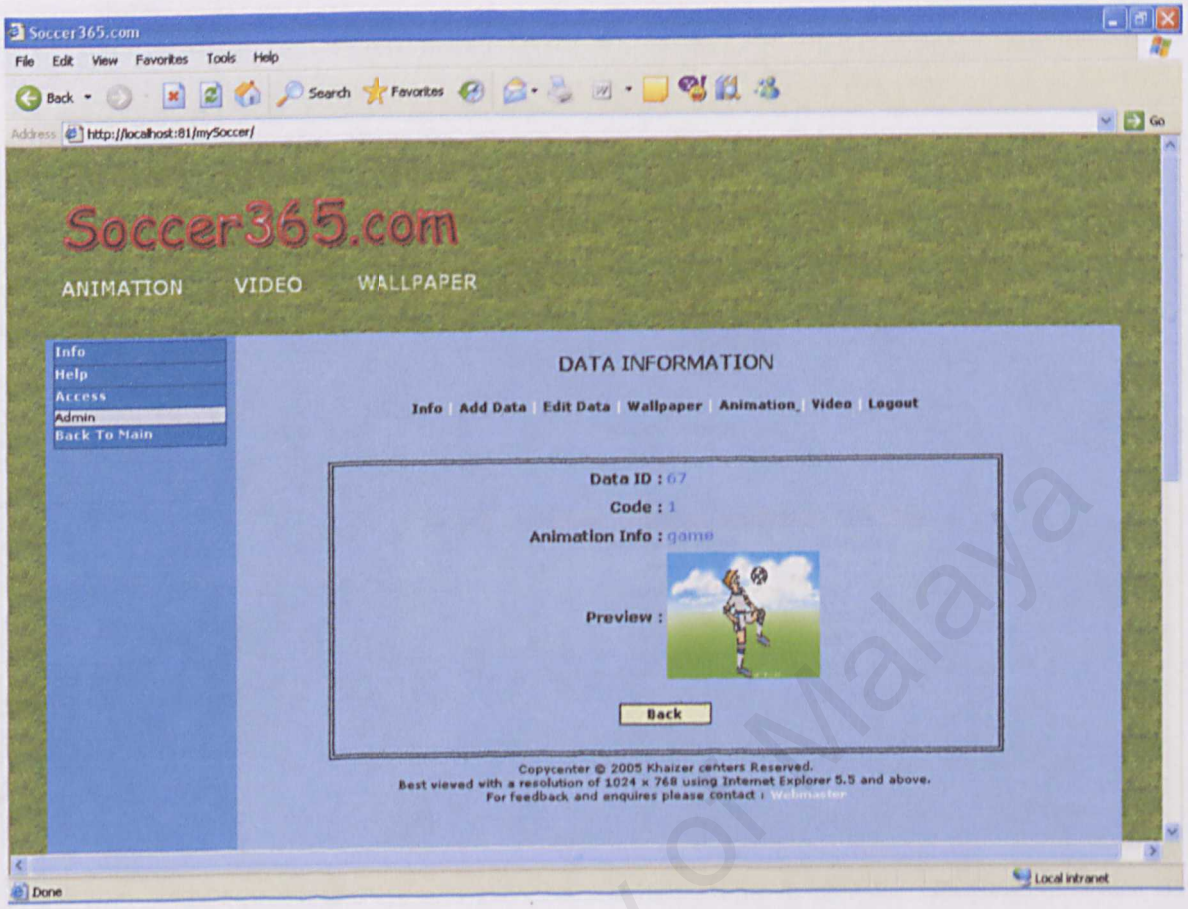

Figure 15: Animation Info

Once you have entered the desired content, whether it may be animation, wallpaper or video, you can then see the data information on that particular content through the '**MORE INFO**' link.

Figure 16: Wallpaper Table

Operation is similar to that of the Animation Table **(See Figure 7).**

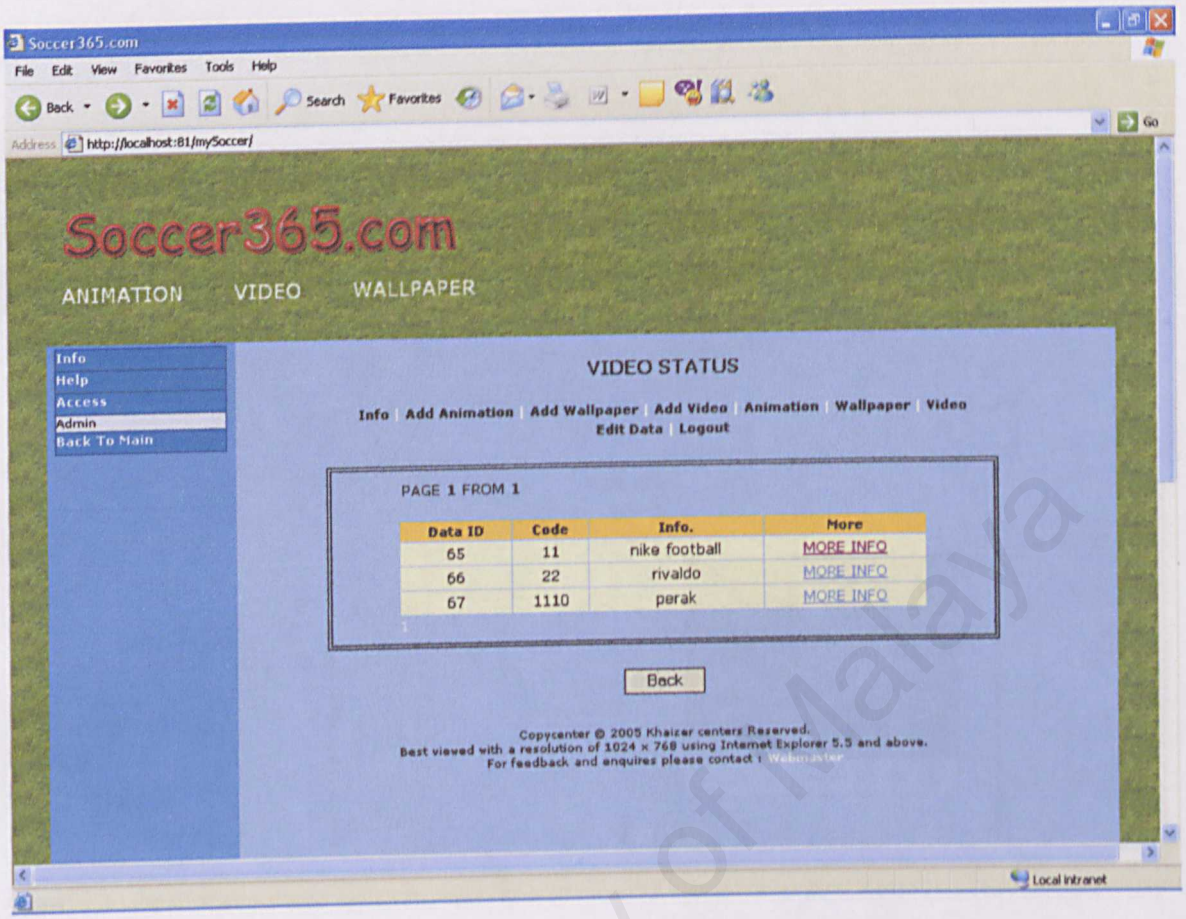

Figure 17: Video Table

Operation is similar to that of the Animation Table **(See Figure 7).**

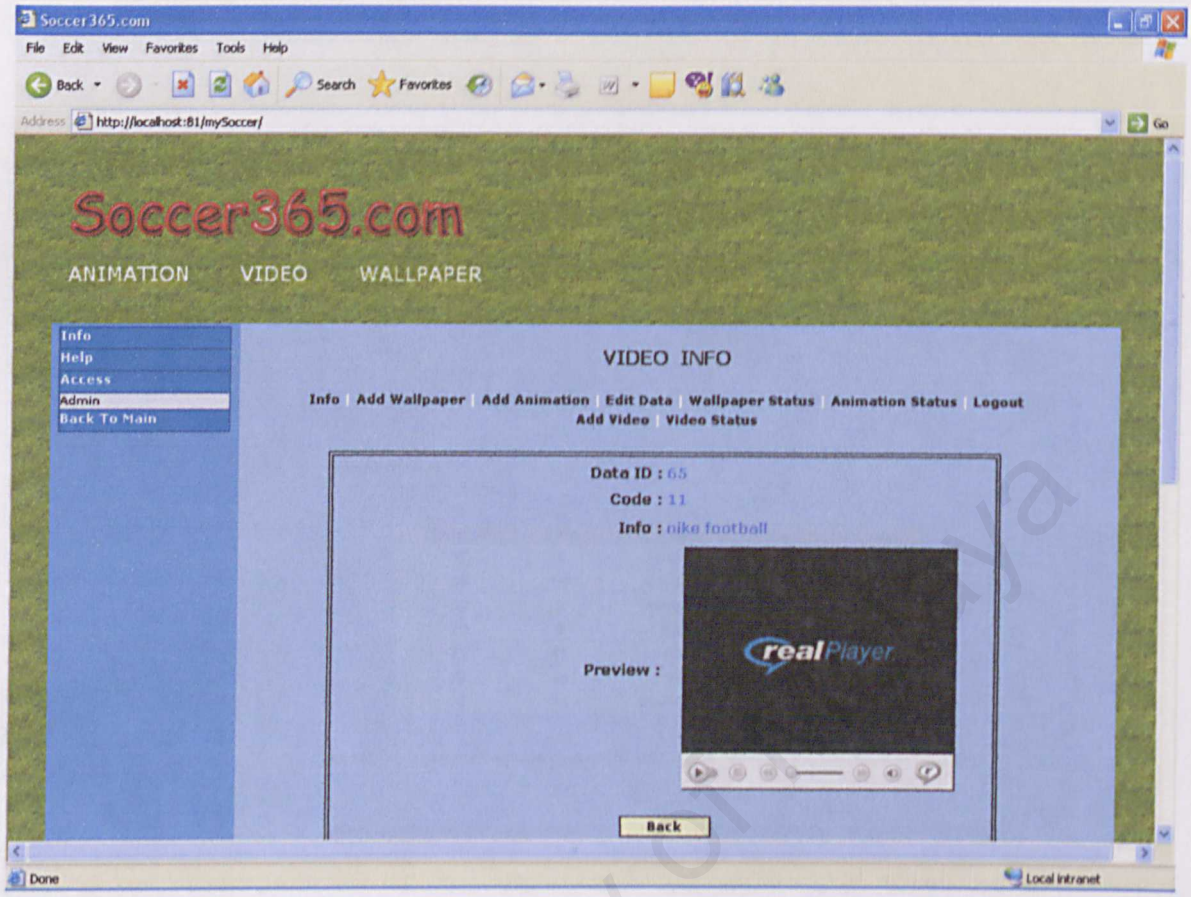

Figure 18: Video Info

The administrator of this particular system is able to view a video clip of the video once he or she has added that particular video into the database. This can be achieved by clicking on the 'MORE INFO' link.

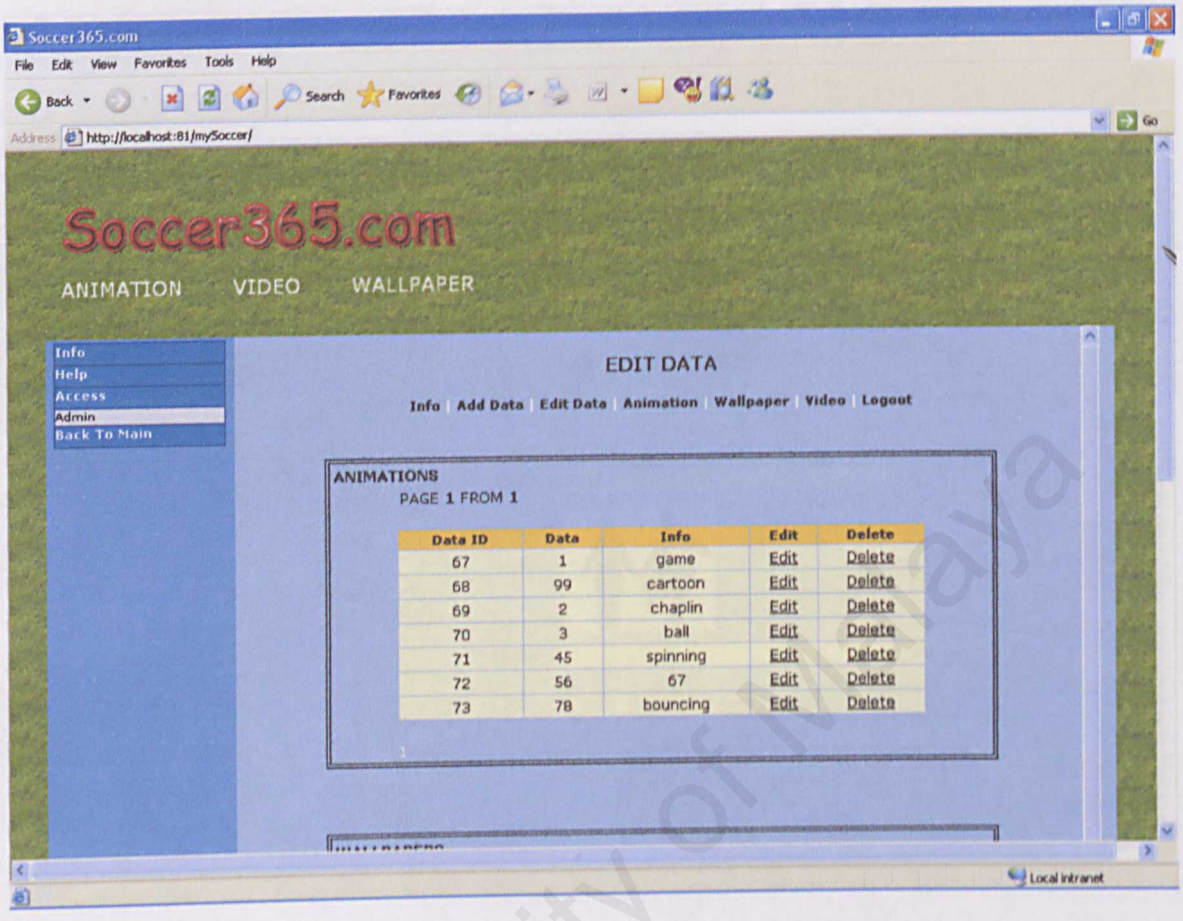

Figure 19: Edit Data

Edit Data provides the administrator with an overall look of all data that has been stored inside the database. All mobile contents such as animation, video and wallpaper can be edited or from this page.

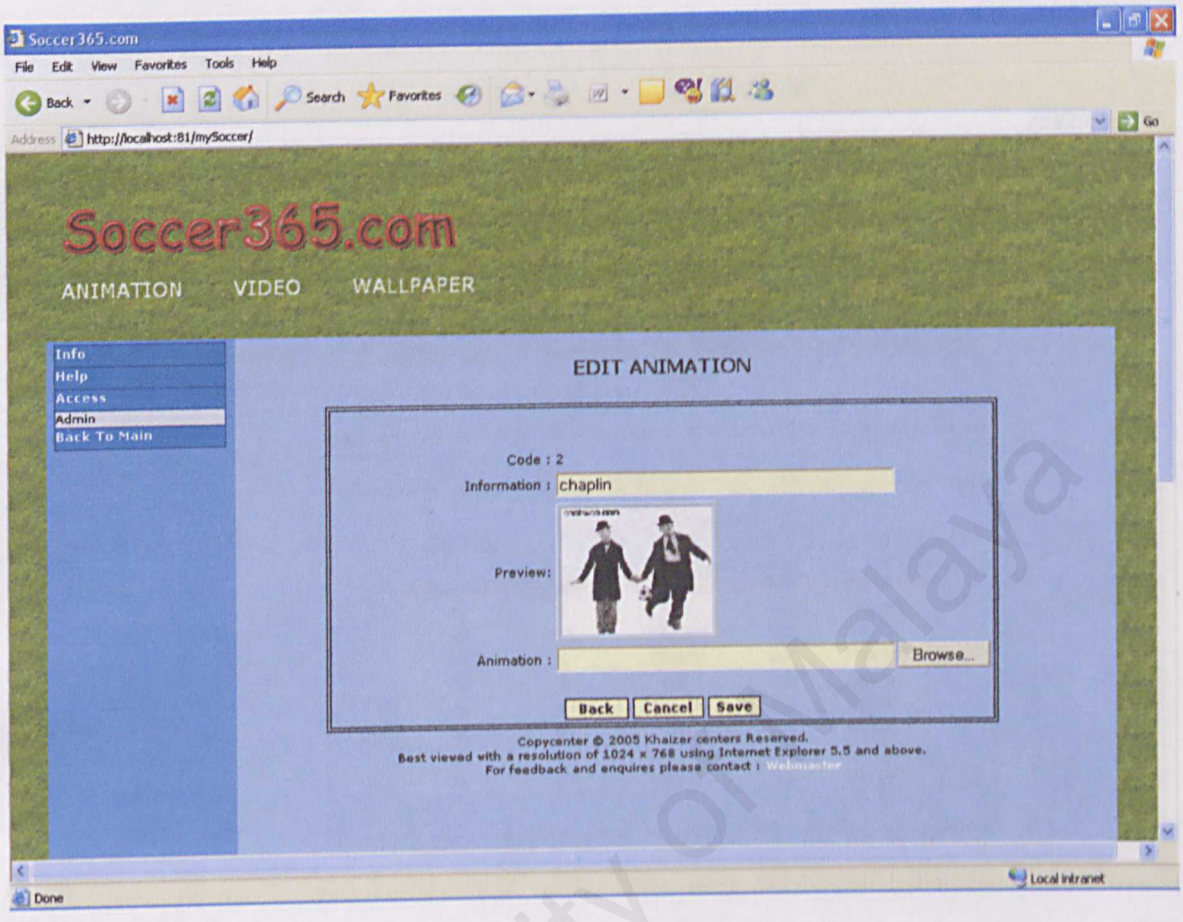

Figure 20: Edit Function

When you click on 'Edit', you will be directed to the 'Edit Animation' page where you can choose to rename the content or choose another animation to replace the current one with.

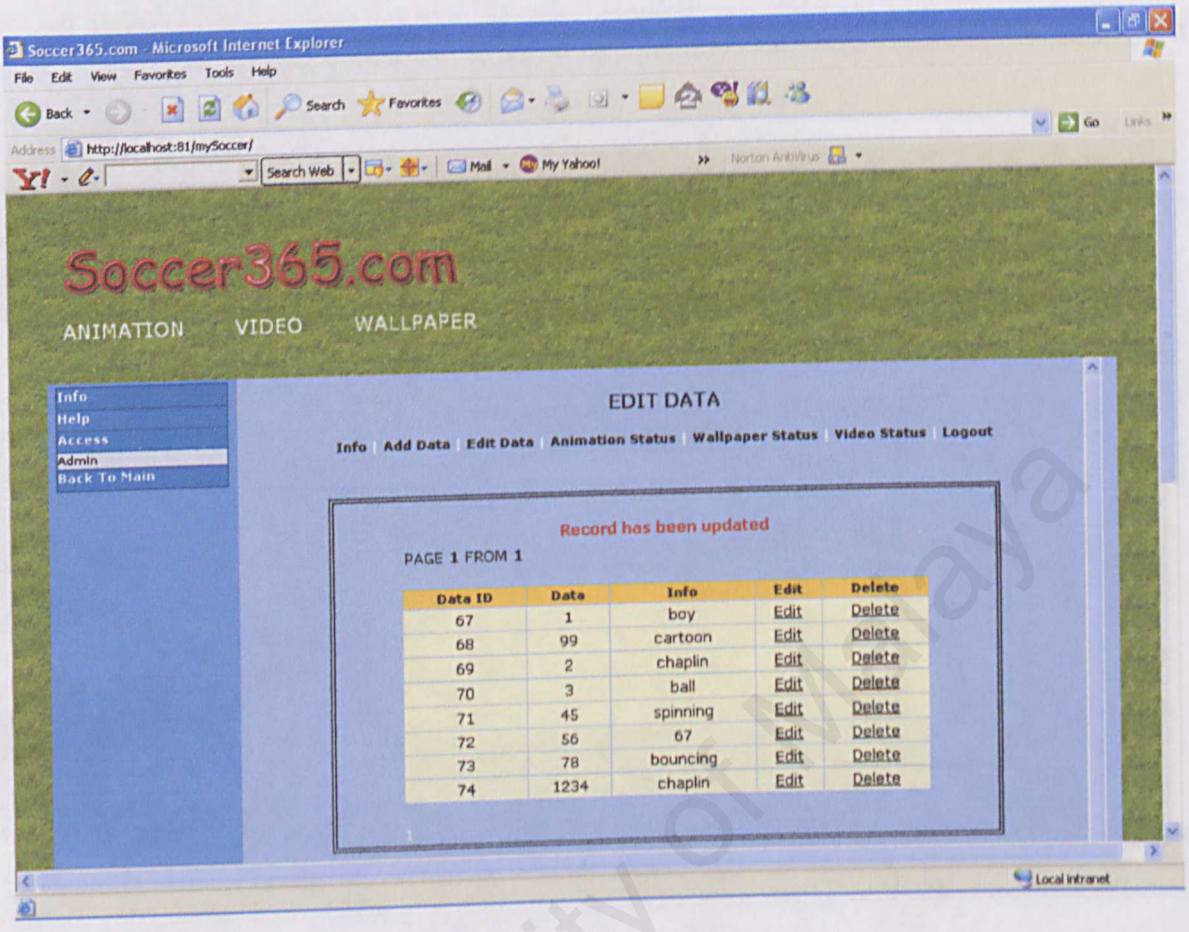

Figure 21: Record update

After you have edited your preferred content, you will then be directed back to the 'Edit Data' page. A message will then appear stating that your database record has been updated. This editing process also applies with the video and wallpaper contents.

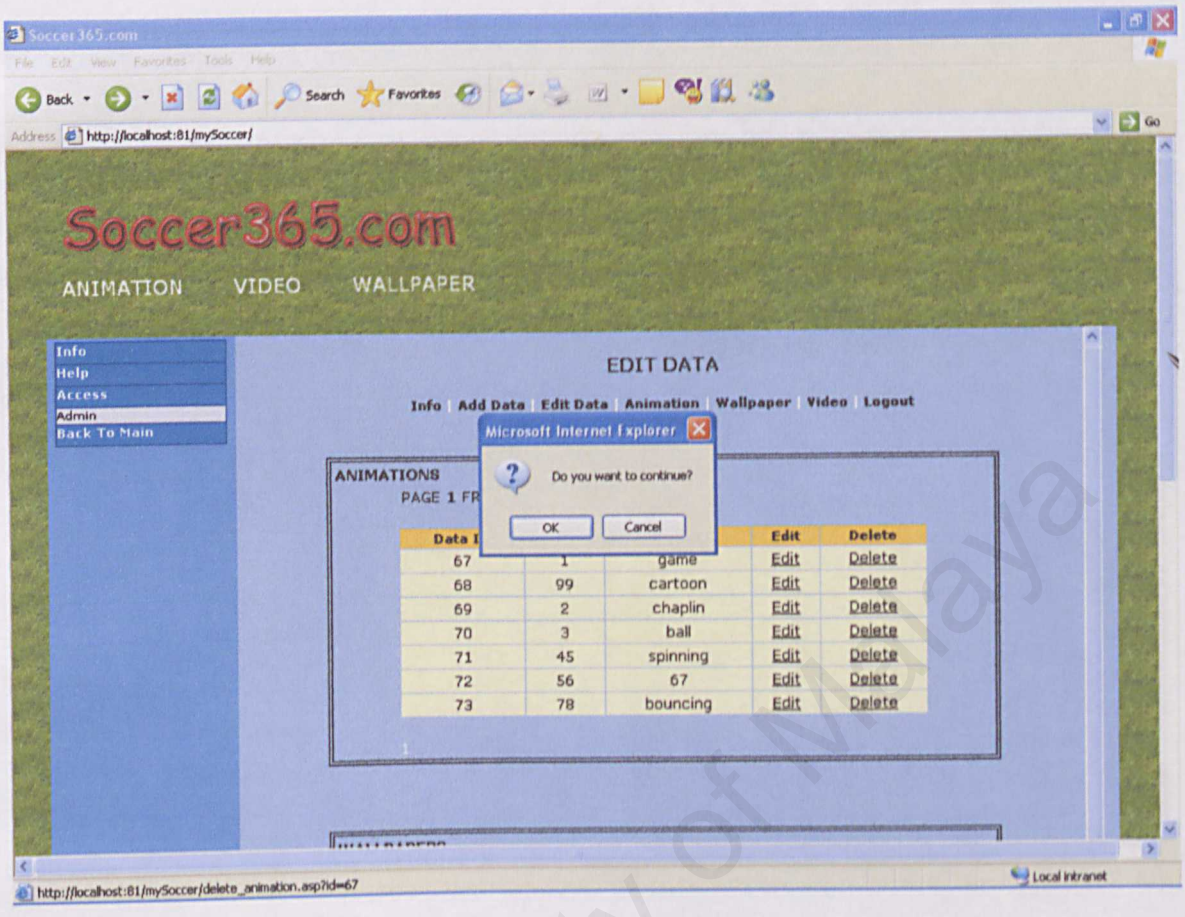

Figure: Delete Function

When you click on 'Delete', a window prompt will request whether you want to proceed with the deletion or abort it.