



**Component Based Architecture For WAP Marketing Services**

**By**

**S.Sivachandran  
Register Number: 71206621050**

**Of**

**KUMARAGURU COLLEGE OF TECHNOLOGY**

**COIMBATORE**

**A PROJECT REPORT**

**Submitted to the**



**FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING**

*In partial fulfillment of the requirements*

*for the award of the degree*

*Of*

**MASTER OF COMPUTER APPLICATIONS**

**ANNA UNIVERSITY  
CHENNAI 600 025**

July 2009

**KUMARAGURU COLLEGE OF TECHNOLOGY****COIMBATORE - 641006****BONAFIDE CERTIFICATE**

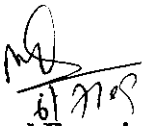
Certified that this project report titled “**Component Based Architecture For WAP Marketing Service**” is the bonafide work of “**Mr. S.Sivachandran**” (Register Number: **71206621050**) who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

*N. Jayar Kanithan*  
Supervisor

  
Head of the Department

Submitted to Project and Viva Examination held on 06.07.2009

*G. Subash*  
6/7/09  
Internal Examiner

  
External Examiner

**TO WHOMSOEVER IT MAY CONCERN**

29/05/09

Certificate is hereby issued to

**Mr. Sivachandran S (71206621050)**

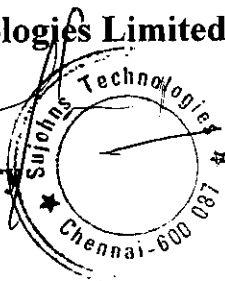
confirming the completion of the said project titled as “**Component Based Architecture For WAP Marketing Services**” in our organization for the partial fulfilment of the requirement for the award of **M.C.A. (Master of Computer Applications)** degree in **Kumaraguru College of Technology**. The project duration is from **December 22, 2008 to May 29, 2009** and engineered using **.Net** technology.

His involvement with the team, commitment to work and conduct during the period were excellent.

We wish them all the prospects in their future endeavours.

**For Sujohns Technologies Limited**

**Authorised Signatory**



## **ABSTRACT**

The Wireless Application Protocol (WAP) is an open, global specification that empowers mobile users with wireless devices (such as mobile phones) to easily access and interact with information and services on the internet. Although some components of the WAP suite have been developed, it lacks a complete general architecture are integrating software component of both the internet and wireless context in a transparent way.

This project presents a general architecture framework to develop and deploy portable applications and services accessible by WAP-Compliant mobile terminals, extending end-to-end services between terminal and business applications. In WAP, ASP.NET can be used to generate Wireless Markup Language (WML) pages dynamically.

The project is a web based software with WAP interface which can be used by any cooperates/ individuals for Customer Relationship Management operations utilizing the full features of WAP.

## ACKNOWLEDGEMENT

First and foremost I thank God for his good will and blessings showered on me throughout the project. The success of this project needs cooperation and encouragement from different quarters. Words are inadequate to express my profound and deep sense of gratitude to those who helped me in bringing out this project successfully.

I am very gladly taking this opportunity to express a special word of thanks to **Dr. M. Gururajan M.Sc., Ph.D**, Head of the Department, Kumaraguru College of Technology, Coimbatore for encouraging me to do this work.

I am very much indebted to **Mrs.V.Geetha, M.C.A., Assistant Professor** Kumaraguru College of Technology, Coimbatore for her complete assistance, guidance and support given to me throughout my project.

I would express heartfelt thanks to our internal guide **Mr. N.Jayakanthan, M.C.A.**, Senior Lecturer, Kumaraguru College of Technology as with out his best guidance it would not have been possible for me to successfully complete this project who also gave his innovative ideas at crucial times and tremendous encouragement.

It is my pleasure to express my profound gratitude to **Sujohns Technologies Limited, Chennai** for admitting into this project. I am thankful to **Ms. Suganthi** of Comprehensive Business Solutions, for his excellent guidance, timely suggestions and constant support in all my endeavors.

## CONTENTS

---

Chapter No.	Title	Page No.
	<b>ABSTRACT</b> .....	iii
	<b>ACKNOWLEDGEMENT</b> .....	iv
	<b>Table of Content</b> .....	v
	<b>List of Table</b> .....	vii
	<b>List of Figures</b> .....	viii
	<b>List of Abbreviation</b> .....	ix
<b>1.</b>	<b>INTRODUCTION</b> .....	<b>1</b>
	1.1 Component Based Architecture.....	1
	1.2 WAP.....	3
	1.3 The Project.....	4
<b>2.</b>	<b>COMPANY PROFILE</b> .....	<b>5</b>
<b>3.</b>	<b>SYSTEM DESIGN</b> .....	<b>7</b>
	3.1 System Architecture.....	7
	3.2 Technology Overview.....	11
	3.3 Functional Overview.....	17
<b>4.</b>	<b>ANALYSIS OF THE PROBLEM</b> .....	<b>19</b>
	4.1 System Analysis.....	19
	4.1.1 Existing System.....	19
	4.1.2 Demerits of Existing System .....	19
	4.1.3 Proposed System.....	20
	4.1.4 Merits of Proposed System.....	21
	4.2 Analysis Tools.....	22
	4.2.1 Dataflow Diagram.....	22
	4.2.2 Process Flow Diagram.....	27
	4.2.3 ER Diagram.....	33
	4.3 Feasibility Study.....	34
	4.3.1 Operational Feasibility.....	34
	4.3.2 Technical Feasibility.....	34
	4.3.3 Economic Feasibility.....	35

4.4	Justification.....	35
5.	<b>REQUIREMENTS SPECIFICATION.....</b>	<b>36</b>
5.1	Software Requirements.....	36
5.2	Hardware Requirements.....	36
5.3	Third Party Interface/Software .....	36
6.	<b>DESIGN AND IMPLEMENTATION.....</b>	<b>37</b>
6.1	Input/Output Design.....	37
6.2	Database Design.....	38
7.	<b>CONCLUSION.....</b>	<b>47</b>
	<b>Appendixes.....</b>	<b>47</b>
A	Sample Code.....	47
B	Screen Shots.....	50
	<b>REFERENCES.....</b>	<b>68</b>

## List of Tables

<b>S. No</b>	<b>Name of Table</b>	<b>Page No.</b>
1	Appointments_master	39
2	Appointments_reschedule	40
3	customer_master	41
4	Emp_master	41
5	post_master	42
6	product_master	42
7	sales_register	43
8	service_master	44
9	service_reschedule	45
10	uid_sequence	46



## List of Figures

<b>S. No</b>	<b>Name of Figure</b>	<b>Page No.</b>
1	Component Based Architecture	2
2	WAP Architecture	8
3	Architecture of ADO.net	15
4	Wireless Application Protocol Network Architecture	16
5	Level 0 DFD	22
6	Level I DFD for Admin	23
7	Level II DFD for Admin	24
8	Level I DFD for Mobile Client	25
9	Level II DFD for Mobile Client	26
10	Process Flow Diagram For Admin	27
11	Process Flow Diagram For Mobile Client	30
12	ER Diagram	33

## List of Abbreviations

Acronyms	Full Form
API	Application Programming Interface
GPRS	General Packet Radio Service
GSM	Global System Mobile
PDA	Personal Digital Assistants
URL	Uniform Resource Locator
UDP	User Datagram Protocol
WAP	Wireless Application Protocol
TCP	Transmission Control Protocol
WWW	World Wide Web

# INTRODUCTION

This section gives details about what a Component Based Architecture and what WAP is. Then it gives an introduction for the project – “Component Based Architecture for WAP Devices”.

## 1.1. Component Based Architecture

---

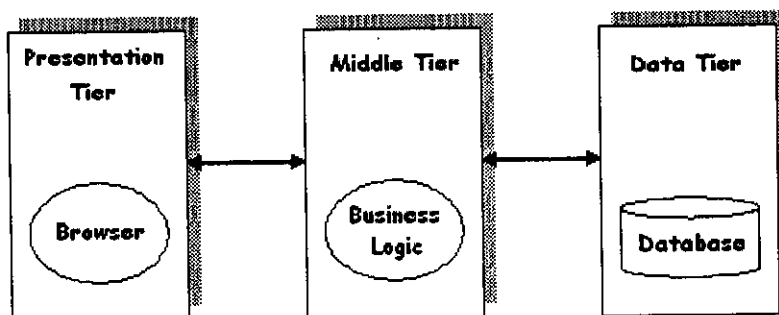
In early days web based software solutions were given using Two Tier Architecture. One tier called the *Server* had the database and the program to access it. The second tier called *Client* had the business logic and presentation logic. Since two logics were present in the same tier maintaining the system was a problem. Recently web based software solutions are given by Component Based Architecture.

Component Based Architecture (CBA) is a client-server architecture which consists of -

- Presentation Layer - user interface
- Middle Layer - functional process logic or business logic
- Data Layer - data storage and data access

**Presentation tier** contains the programs that provide the graphical user interface (GUI) and application-specific entry forms or interactive windows.

**Middle Tier** is located on a local area network (LAN) server or other shared computer. The business logic acts as the server for client requests from workstations. In turn, it determines what data is needed (and where it is located) and acts as a client in relation to a third tier of programming that might be located on a mainframe computer.



**Figure 1.1 – The Component Based Architecture**

**Data Tier** includes the database and a program to manage read and write access to it. While the organization of an application can be more complicated than this, the 3-tier view is a convenient way to think about the parts in a large-scale program.

The Tiers are developed and maintained as independent modules, most often on separate platforms. The CBA is considered to be a software architecture and a software design pattern. Apart from the usual advantages of modular software with well defined interfaces, the CBA is intended to allow any of the three tiers to be upgraded or replaced independently as requirements or technology change. For example, a change of operating system from Microsoft Windows to Unix would only affect the user interface code.

Typically, the user interface runs on a desktop PC or workstation and uses a standard graphical user interface, functional process logic may consist of one or more separate modules running on a workstation or application server, and an RDBMS on a database server or mainframe contains the data storage logic. The middle tier may be multi-tiered itself (in which case the overall architecture is called an "n-tier architecture").

We implement this three tier architecture using ASP.NET in our project.

## 1.2. WAP

---

Wireless Application Protocol or WAP is an open international standard for applications that use wireless communication. Its principal application is to enable access to the Internet from a mobile phone or PDA. The WAP standards suite is maintained by an industry consortium called the WAP Forum. Founded by Ericsson, Motorola, Nokia, and Openwave (then known as Unwired Planet) in June 1997, the WAP Forum now includes hundreds of member companies that are infrastructure providers, software companies, and content providers. The goal of the WAP Forum is to address the problems of wireless Internet access, ensuring that access is not limited by vendor or underlying network technology. Since its creation, the Wireless Application Protocol has passed through minor revisions (from 1.0 to 1.1, 1.2, and 1.2.1). WAP 2 is the first major revision since 1998.

A WAP browser is to provide all of the basic services of a computer based web browser but simplified to operate within the restrictions of a mobile phone. WAP is now the protocol used for the majority of the world's mobile internet sites, known as WAP sites. The Japanese i-mode system is currently the only other major competing wireless data protocol.

Mobile internet sites, or WAP sites, are websites written in, or dynamically converted to, WML (Wireless Markup Language) and accessed via the WAP browser. Before the introduction of WAP, service providers had extremely limited opportunities to offer interactive data services. Interactive data applications are required to support now commonplace activities such as -

- email by mobile phone
- tracking of stock market prices
- sports results
- news headlines
- music downloads

### 1.3. The Project

---

Most Internet technologies are designed for desktop and computers running on reliable networks with relatively high bandwidth. Hand-held wireless devices, on the other hand, have a more constrained computing environment. They tend to have less memory, less powerful CPUs, different input devices, and smaller displays. The Wireless Application Protocol (or WAP), which is a specification developed by the WAP Forum, take advantage of the several data-handling approaches already in use.

Developing wireless applications using WAP technologies is similar to developing web pages with a markup language (eg HTML) because it is browser based. The Wireless Application Protocol (WAP) is an open, global specification that empowers mobile users with wireless devices (such as mobile phones) to easily access and interact with information and services on the internet. Although some components of the WAP suite have been developed, it lacks a complete general architecture are integrating software component of both the internet and wireless context in a transparent way.

This project presents a general architecture framework to develop and deploy portable applications and services accessible by WAP-Compliant mobile terminals, extending end-to-end services between terminal and business applications. In WAP, ASP.NET can be used to generate Wireless Markup Language (WML) pages dynamically.

The project is a web based software with WAP interface which can be used by any cooperates/ individuals for Customer Relationship Management operations utilizing the full features of WAP.

## COMPANY PROFILE

### Profile

The company was incorporated on 9th March 2000 as a public limited company with a share capital of Rs.100 million, by former executives of Soffia Software Limited which is subsequently acquired by Quintegra Solutions, who understood the problems of deploying various variety of software solutions. Together they launched their first product in the year 2002, cPay, which was immediately embraced by organizations such as Java Green, Anand Group.

Sujohns Technologies Limited, a leading corporate company with nation wide presence, is the single largest source for a wide range of software products and services. Ever since its inception from the year 2000, Sujohns focus has always been the excellence in quality of products and software services.

Sujohns is a leading provider of scalable integrated enterprise software solutions for various organizations such as financial, manufacturing, health care, transportation, retail and pharmaceuticals which are located in India and abroad. Sujohns has clearly established itself as a fast-growing and highly respected technology company attracting strong financial backing and blue chip clients.

Sujohns is “The software firm with the rich set of software services & products”. Sujohns ever since its inception from the year 2000 has placed itself on the path of continuous growth. Within a period of four years it has a sweeping turnover of Rs.10 million.

Technology partnerships and knowledge derived from persons having years of experience have enabled us to deliver state-of-the-art solutions and to sustain customer loyalty. Our Services and Product offerings have been built with the combination of the latest in technology and understanding client needs in various business areas. We offer technology solutions tailored for your business.

## **Vision**

To be a technology company that understands the needs of a customer and deliver quality software solutions by qualified professionals.

## **Quality Policy**

We commit to deliver quality software and value-added services to our customers, for the first time and every time.

## **Core Team: -**

The core management team of the company consists of engineers and professionals who are also the promoters and major shareholders of the company.

The common features are

- ◆ High Quality data communication lines for easy and quick upload/download of data
- ◆ Secured development environment
- ◆ Round the clock pantry services
- ◆ Completely networked work environment
- ◆ Highly alert security

Sujohns has high-speed Internet connection linked to Wide Area Network. The network links all offices to provide data, voice and video communication. The links enables to provide cost-effective maintenance and Co-development of software with the clients. It also provide an effective medium to communicate with customers via electronic mail and video-conferencing



## SYSTEM DESIGN

### 3.1. System Architecture

---

The main goal of our architecture is to extend basic functions of the WAP framework with more sophisticated services such as support for transactions. The proposed four-tier architecture integrates standard components of WAP and WWW domains and allows realizing the following aims:

- End-end communication between mobile terminal (Micro Browser) and web server (i.e., Business logic);
- Remote access to databases;
- Support for transactions, integrating transactional services offered by DBMS;

The four layers are:

1. User Layer - i.e., a wireless terminal with a WML Micro Browser.
2. Network Layer - represented by the WAP gateway, which acts as an HTTP proxy.
3. Application Layer - composed by the application server implementing the business logic.
4. Data Layer - composed by the DBMS servers and the hosted databases. The Data Layer can also integrate different sources of data, such as legacy applications and semi-structured data (HTML, XML).

In the design, the WAP gateway operates as proxy whereas the Application server is executed on a different host. The design choice allows:

- To separate the WAP-based connectivity operations by the application implementation;
- To provide open and portable solution based on the use of standard APIs.

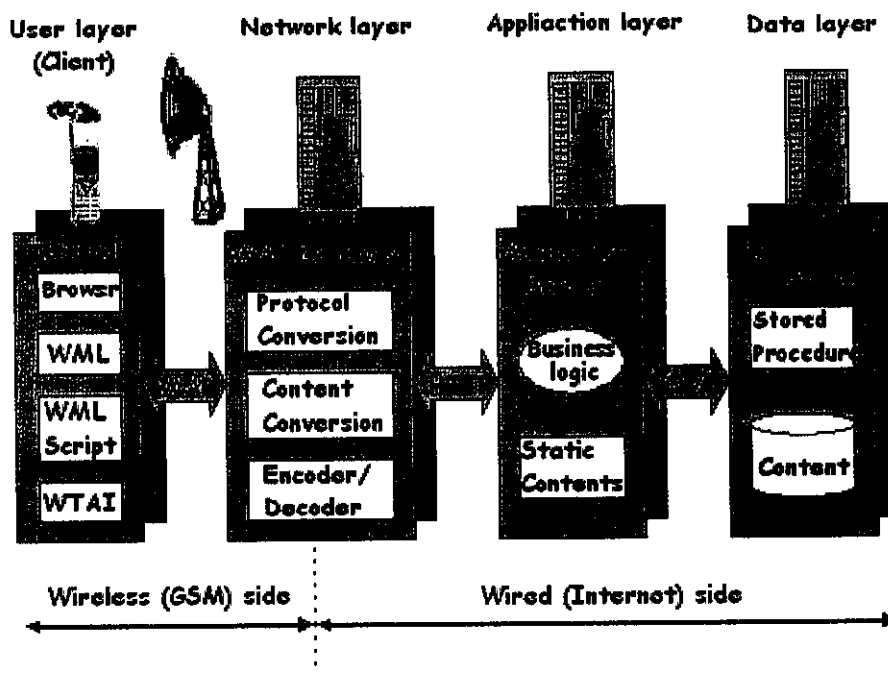


Figure 6.1 – The WAP Architecture

### Layer 1 : User Layer (Mobile Client)

A WAP-complaint wireless terminal (cellular phone or PDA) represents the user layer. It is generally characterized by the following constraints:

- Small, low-resolution displays and limited user-input facilities. Displays have few lines of text and low resolution. Terminal lack of mouse and keyboard, thus requiring new User Interface difference by typical GUI. Current I/O devices are touch-screen and voice menu.
- Limited computational resources (CPU, RAM) duration and power of batteries.

The main software components of the WAP complaint terminal are:

- The client-side implementation of the WAP layers in order to accomplish peer-to-peer communication with WAP gateway.
- A Micro-Browser running on top of the WAP stack.

As said before, the Micro-Browser is a simplified web browser able to display WML contents and to execute WML Script code. WML, a lightweight version of

HTML, defines a new WAP-specific User Interface model that is based on the Card and Deck metaphor. Deck and Cards are referred by URLs.

The WML browser is equipped with a byte code-based virtual machine, so it can also execute compiled scripts on behalf of the application server. Scripts are coded on WML Scripts, a scripting language derived from JavaScript. WML Script source code need to be compiled into the WML Script byte code before it can be run on WAP Terminal.

WML Script enhances Micro-Browser with procedural logic, so it is possible to perform action as field validation or small conversion operation.

## **Layer 2: Network-Application interface**

Almost all of the WAP gateways support HTTP connections that allow to separate the application implementation by the network and terminal details. Using HTTP between the Network and Application Layers allows the easy integration of pre-existing web-based applications. On the other hand this could require additional logic to filter or adapt application outputs to WML

## **Layer 3: Application Server (Application Layer)**

The Application layer architecture employs a component-based design, using the reusable class technology to implement and deploy object-oriented distributed applications. The .Net platform supports ADO.Net compliant drivers through which it is possible to access common DBMS. The Application Programming Model (APM) has the following properties:

- Clear separation between business logic and low-level services. The former is implemented by reusable components whereas the latter are offered by the platform (network and data connections, multi threading, transactions, component persistence, distribution, security and life-cycle.).

- Server-side reusable objects – the attribute-based declarative programming model offered by .Net allows to dynamically choose the runtime behavior of a component.
- Support to distributed transactions involving multiple objects executed over different Virtual Machines.
- The Application Server is designed in a modular way and comprises the following specialized layers  
Server Side Implementation: They manage the HTTP communication with the WAP gateway, which acts as a proxy for a mobile terminal, and care of the dynamic WML/HTML code generation.
- Reusable Classes: They represent a server-side proxy of the client and are used to model the clientapplication interaction (business rules). An Object instance is bound to a single client. This layer manages the maintaining of the application state through state-less communications.

#### **Layer 4 : Application-Data interface**

To access relational databases the ADO.Net standard is used. ADO.Net allows to connect to a database, to start query/transactions and to obtain results (ResultSet) avoiding DBMS related tricks and details.

The connection to a database can be either long-term (it is maintained for the component lifetime) or short-term (it is acquired and released for each data access method invocation).

ADO.Net allows easy operation and development, so they are used especially with script-based server implementation. The Application-Data interface should also support access to different sources of data, such as semi-structured data (text, HTML, XML), datawarehouse or multi-dimensional database, legacy data.

Moreover, the navigation of the metadata repository eventually offered by the Data layer should be supported.

## 3.2. Technology Overview

---

The various technologies used in the software system are:

- ASP.NET
- ADO.NET
- WAP
- WML and WML Script



### 3.2.1. ASP.NET:

---

- **Enhanced Performance:**

ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.

- **World-Class Tool Support:**

The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.

- **Power and Flexibility:**

Because ASP.NET is based on the common language runtime, the power and flexibility of that entire platform is available to Web application developers.

The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.

- **Simplicity:**

ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

- **Manageability:**

ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications. Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools. This "zero local administration" philosophy extends to deploying ASP.NET Framework applications as well. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.

- **Scalability and Availability:**

ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor

environments. Further, processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.

- **Customizability and Extensibility:**

ASP.NET delivers a well-factored architecture that allows developers to "plug-in" their code at the appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET runtime with your own custom-written component. Implementing custom authentication or state services has never been easier.

- **Security:**

With built in Windows authentication and per-application configuration, you can be assured that your applications are secure.

### **3.2.2. ADO.NET:**

---

ADO.Net is a suite of data access technologies included in the .NET Framework class libraries that provide access to relational data and XML.ADO.NET consists of classes that make up the DataSet (such as tables, rows, columns, relations, and so on), .NET Framework data providers, and custom type definitions (such as SqlTypes for SQL Server). The following diagram illustrates the components of ADO.NET architecture.

ADO.NET provides consistent access to data sources such as Microsoft SQL Server, as well as data sources exposed through OLE DB and XML. Data-sharing consumer applications can use ADO.NET to connect to these data sources and retrieve, manipulate, and update data.

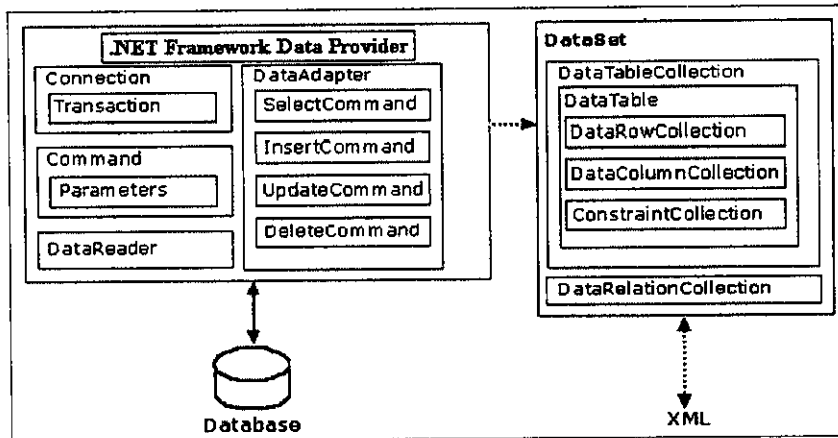
The ADO.NET classes are found in System.Data.dll, and are integrated with the XML classes found in System.Xml.dll. When compiling code that uses the System.Data namespace, reference both System.Data.dll and System.Xml.dll.

The ADO.NET components have been designed to factor data access from data manipulation. There are two central components of ADO.NET that accomplish this: the DataSet, and the .NET Framework data provider, which is a set of components including the Connection, Command, DataReader, and DataAdapter objects.

The ADO.NET DataSet is the core component of the disconnected architecture of ADO.NET. The DataSet is explicitly designed for data access independent of any data source. As a result it can be used with multiple and differing data sources, used with XML data, or used to manage data local to the application. The DataSet contains a collection of one or more DataTable objects made up of rows and columns of data, as well as primary key, foreign key, constraint, and relation information about the data in the DataTable objects. The other core element of the ADO.NET architecture is the .NET Framework data provider, whose components are explicitly designed for data manipulation and fast, forward-only, read-only access to data. The Connection object provides connectivity to a data source. The Command object enables access to database commands to return data, modify data, run stored procedures, and send or retrieve parameter information. The DataReader provides a high-performance stream of data from the data source. Finally, the DataAdapter provides the bridge between the DataSet object and the data source. The DataAdapter uses Command objects to execute SQL commands at the data source to both load the DataSet with data, and reconcile changes made to the data in the DataSet back to the data source. Users can write .NET Framework data providers for any data source. The .NET Framework ships with two .NET Framework data providers: the .NET Framework Data Provider for SQL Server and the .NET Framework Data Provider for OLE DB.



.NET Framework data providers, and custom type definitions (such as SqlTypes for SQL Server). The following diagram illustrates the components of ADO.NET architecture.



ADO.NET provides consistent access to data sources such as Microsoft SQL Server, as well as data sources

**Fig 3.2.2 Architecture of ADO.Net**

### 3.2.5. WAP:

In 1997, several wireless-phone manufacturers organized an industry group called the Wireless Application Protocol Forum. This group defined the WAP specification in the form of a long, describes technical document series that defines stirs for implementing wireless-network applications.

WAP allows Internet access through mobile devices, PDA that have constraints Like less memory, low power CPU, small-display screen.

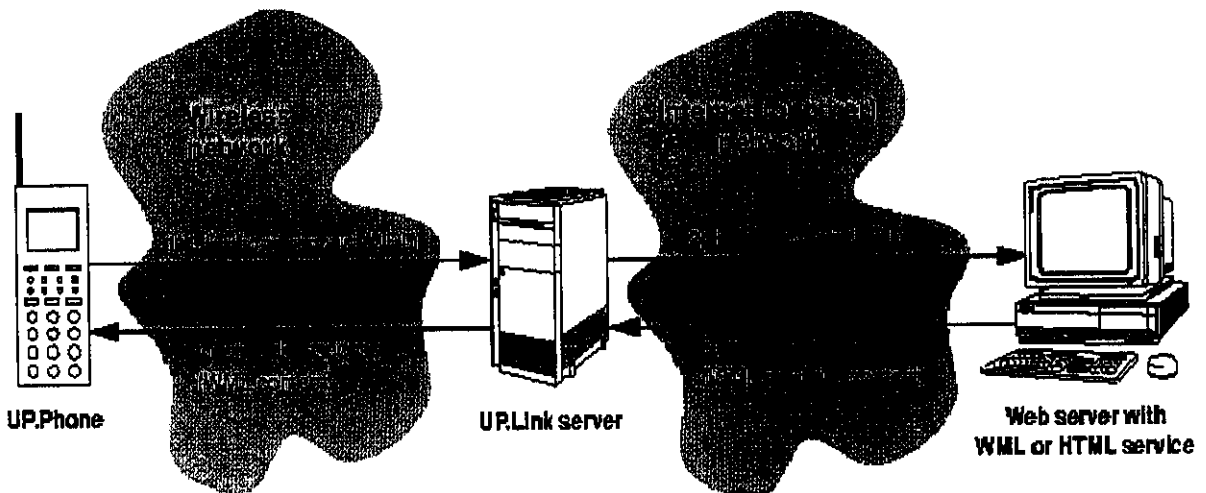
It consists of two essential elements:

- End-to-end application protocol.
- Application environment based on the browser.

### Network Structure of WAP:

A mobile phone sends WAP request to a WAP gateway. The gateway upon receiving a WAP request sends an HTTP request to a plain web server, which provides the content

through a normal HTTP response (the web server perceives the gateway as a proxy server)



**Figure 3.2.5 – Wireless Application Protocol network architecture**

response into a WAP response for the mobile device. The micro browser in the mobile terminal interprets the response and displays it appropriately.

### **3.2.6. WML and WML Scripts:**

WAP technology has 3 major supporting components. The first is the wireless markup language. WML is the WAP equivalent to HTML and is based on XML. It makes optimal use of small screens, with a built-in scalability from 2-line text displays to the full graphics screens on smart phones and communication devices. The Wireless Application environment specification defines the syntax, variables, and elements used in a valid WML file. WML employs the concept of decks and cards. Each card is a frame displayed on the screen. We refer to a logical collection of interlinked cards as a deck, usually stored in a single WML file.

The second is WML script, a client side scripting language used with WML that makes WML pages dynamic (similar to what Java Script allows with HTML) WML Script makes minimal demands on memory and CPU usage, omitting a no. of functions that are not required and that are present in other scripting languages to wireless application

regular scripting languages are recourse intensive pages on a mobile phone. WML Script is a scaled-down, simplified script, developed successfully to fit the WAP architecture of decks and cards. It lets the developer provide inter activity in WAP pages without taking the valuable wireless resources.

The third supporting component is wireless bitmaps; WBMP is the difficult picture format for WAP. WBMPs are uncompressed, monochrome black and white bitmaps for use in devices with small screens and various bandwidth connections. However, the screen size and bandwidth along with limited graphics capabilities are WBMPs major constraints. Recent developments in displays technology provide color bitmaps for the latest WAP browsers.

### 3.3 Functional Overview

---

The Business Modules of the system are

- Appointment
- Product
- Customer
- Sales Transaction
- Service
- Posting

#### **Appointment Management:**

- The employee's appointments to the customer's places in the day-to-day life are kept track in this module.
- This appointment are classified into
  - Scheduled appointments
  - Rescheduled appointments
- The purpose of this module is that the employee can easily view the appointments of that day via the web; also admin can be able to find the appointment and their status based on the date.

**Product Management:**

- Purpose of this model is to know about the companies' products and their functions.

**Customer Information Management:**

- This module allows the user to keep track of customer information such as customer name, contact person, phone number and address.

**Sales Management:**

- Once the customer comes to the stage of buying the product, the sales management can take care about the transaction between the company and the customer.
- Purpose of this module is to keep track of the sales transaction between the company and the customer.

**Support/ Service Management:**

- This module is specifically for the supports and service management.
- This module is classified into
  - Scheduled services.
  - Rescheduled services.
- Purpose is to manage the service contract information about the customers.

**Posting Management:**

- Admin people can send the comments to the employees.

## **ANALYSIS OF THE PROBLEM**

Due to the advancement in technology it became crucial to keep track of time bounded data (i.e., those data values whose characteristics will change according to time and space) to be monitored by cooperates/ individuals in order to control their business for example stock exchanges, news, etc., while on the move.

### **4.2. System Analysis**

---

#### **4.2.1. Existing System**

Existing system requires notepads/ tabletops standalone computers to view the information which is presented on the web but this solution only handset is sufficient enough, provided the websites are all WAP enabled.

On the earlier days the required information was given directly to users in the following ways:-

- User calls the company to get information
- User if a mobile user could get it through SMS
- With the help of Internet connection user can retrieve the information from the internet connection user can retrieve the information from the internet via browser like Netscape Navigator, Internet Explorer, Opera etc.

#### **4.2.2. Disadvantages of Existing**

- Can't keep track of time bounded data while on the more eg. exchange, news etc.
- Cost of system is high hence all kinds of people can't buy
- User gets bulks of extra information which are not required. Thus time and cost is wasted

- Requires notepads/ tabletops/ standalone computers to view the information which presented on the web
- Over the internet, the tendency to visiting other sites/ viewing advertisements will be more

### 4.2.3. Proposed System

This project is designed for receiving real-time data from a database servers running over TCP/IP based networks using server side scripting languages. Mobile devices are now a days getting common. Most of the devices has WAP enabled micro browser embedded in them. Many network services providers are giving options for their clients to access the web services running over internet. This project is conceptualized for making use of this option.

In the server side i.e., in the web server, a aspx page will be available in the appropriate directory. The functionality of aspx page is to dynamically generate the WML pages required for the Mobile device since WAP enabled mobile devices can understand only WML tags. All database lookup will be done by ADO.NET component for the real-time data.

Over the internet, the tendency of visiting other sites or viewing advertisements will be more, but in the current solution only the necessary information only will get posted on the mobile using WAP Interfaces thus saving time Proposed System helps cooperates/ individuals to be informed about the real-time changing data available in a legacy server on their mobile or PDAs, at an effective cost. It is handy, reliable and easy to maintain because of *Thin Client Implementation (Zero Percent Installation)*.

This project is designed for receiving data from a database servers running over TCP/ Based networks using server side scripting languages. Mobile devices are now a days getting common. Day-by-day the scope of mobile technologies are getting increased. In recent advancements the various dependent factors like Network Bandwidth, GSM rates, handset device cost are available to a common man at affordable rates. The speed of

bandwidth is enabling more data to be accessible in less time even on handsets. Wide variety of websites are providing WAP enabled interfaces. Thus the current solution will be market leading solution for the current and future generation. Though mobile device have the following constraints in computing environment –

- Less memory
- Less powerful CPUs
- Different input devices
- Smaller display screen
- Lower throughput

By using new architectural designs and system platform we can cope up with the problems and facilitate the integration between the wired and wireless world.

#### **4.2.4. Advantages of Proposed System**

- Helps to keep track of time bounded data while on the move.
- No need of notepads/tabletops/standalone computers. Only handset is sufficient.
- Network bandwidth, GSM ratios, handset device cost are all available in affordable rates.
- Only necessary information will get posted on the mobile using WAP Interfaces thus saves time and cost.
- All information is entered into the database server by the user from the mobile and also the user can access the related information at anytime, anywhere.
- No additional overhead to the end-users other than internet cost (GPRS) to view the time bounded data.
- Database design is very flexible so that it can be migrated to any Major RDBMS.
- No separate additional resource implementation is required for commercial use of this solution.
- Authentication provided thus transactions are trusted.
- Messaging is enabled so that one-to-one communication can be done effectively.

## ANALYSIS TOOLS

---

### 4.3.1. Dataflow diagram

Dataflow is the movement of data in a system from a point of origin to a specified destination indicated by line or arrow. Dataflow diagram is the graphical representation of the data movements, processes and files (data stores) used in support of information systems.

#### ➤ Context Level DFD for Admin and Mobile Client

This level shows the graphical representation of the system in admin and client side. It clearly depicts the overall input and overall output the system is going to handle.

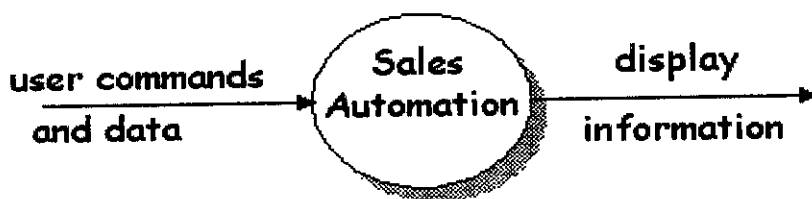


Figure 4.3.1– Level 0 DFD for Admin and Mobile Client



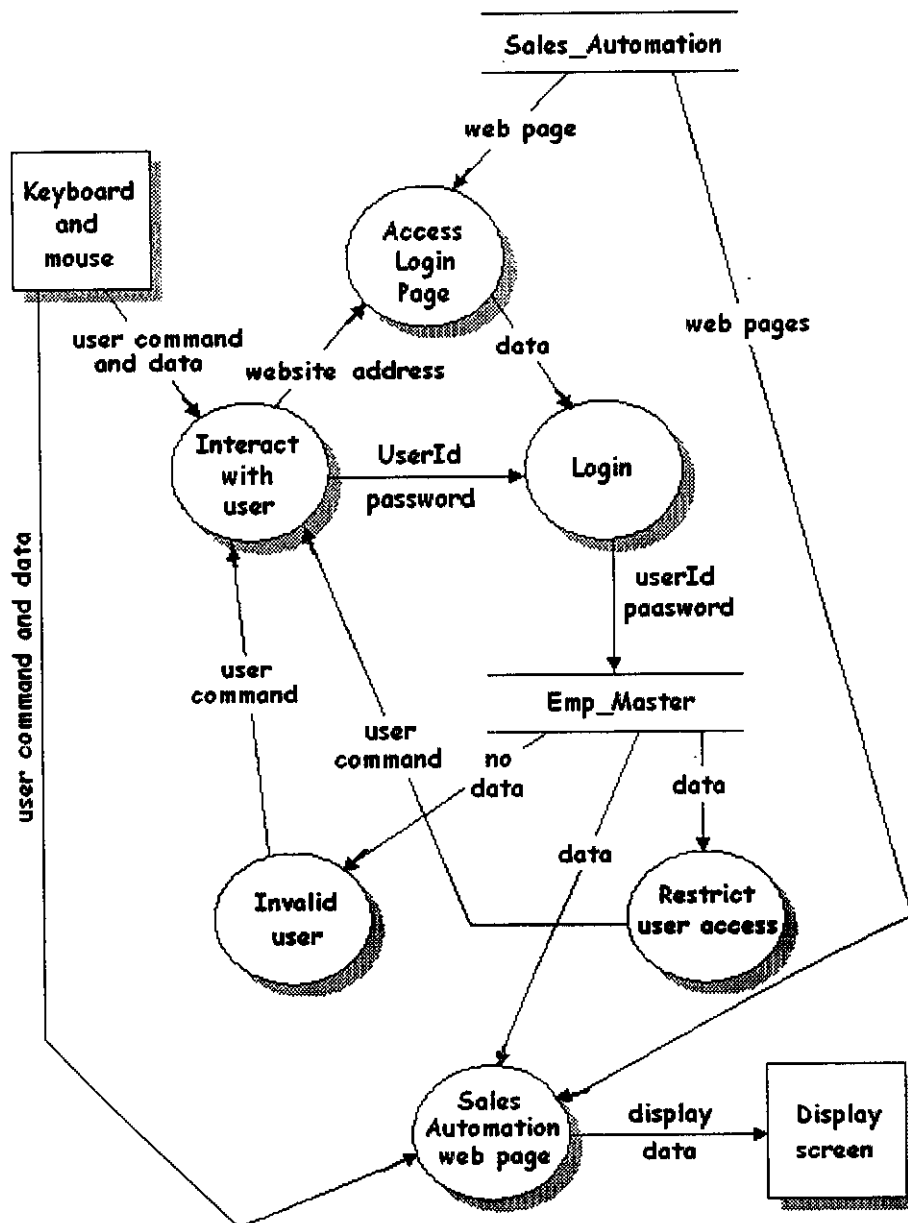
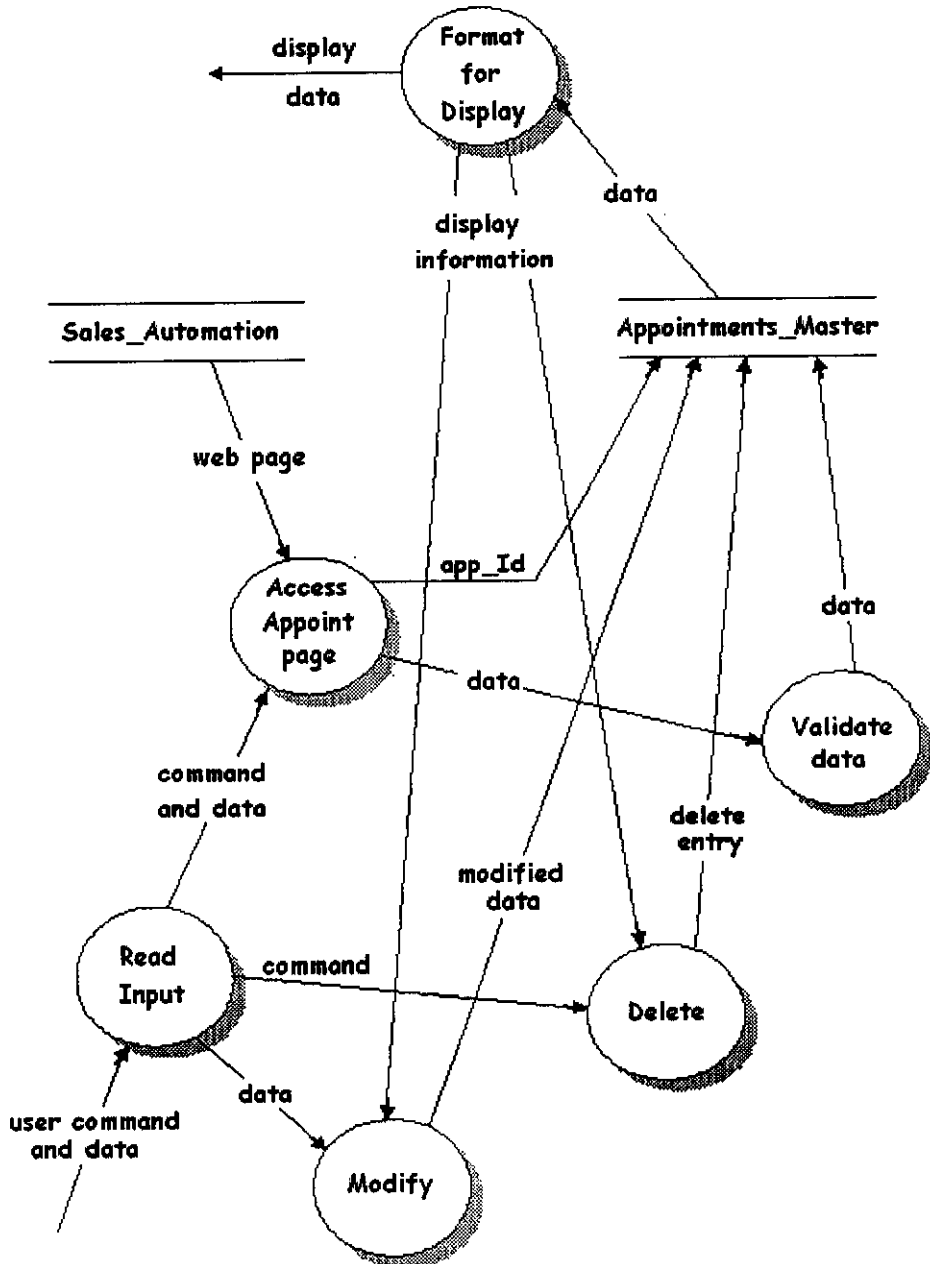


Figure 4.3.2 – Level I DFD for Admin



The above DFD is common to all modules in the admin side.

Figure 4.3.3 – Level II DFD for Admin

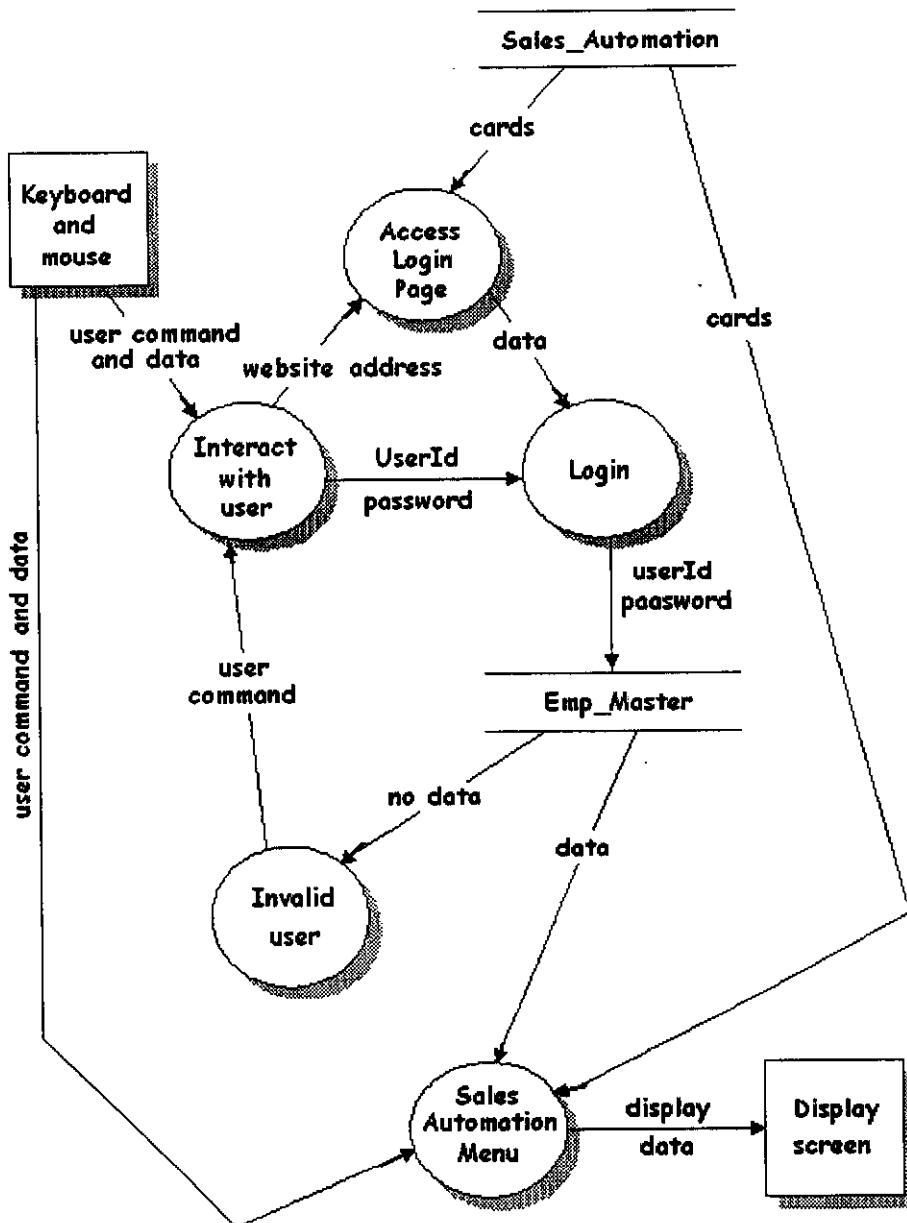
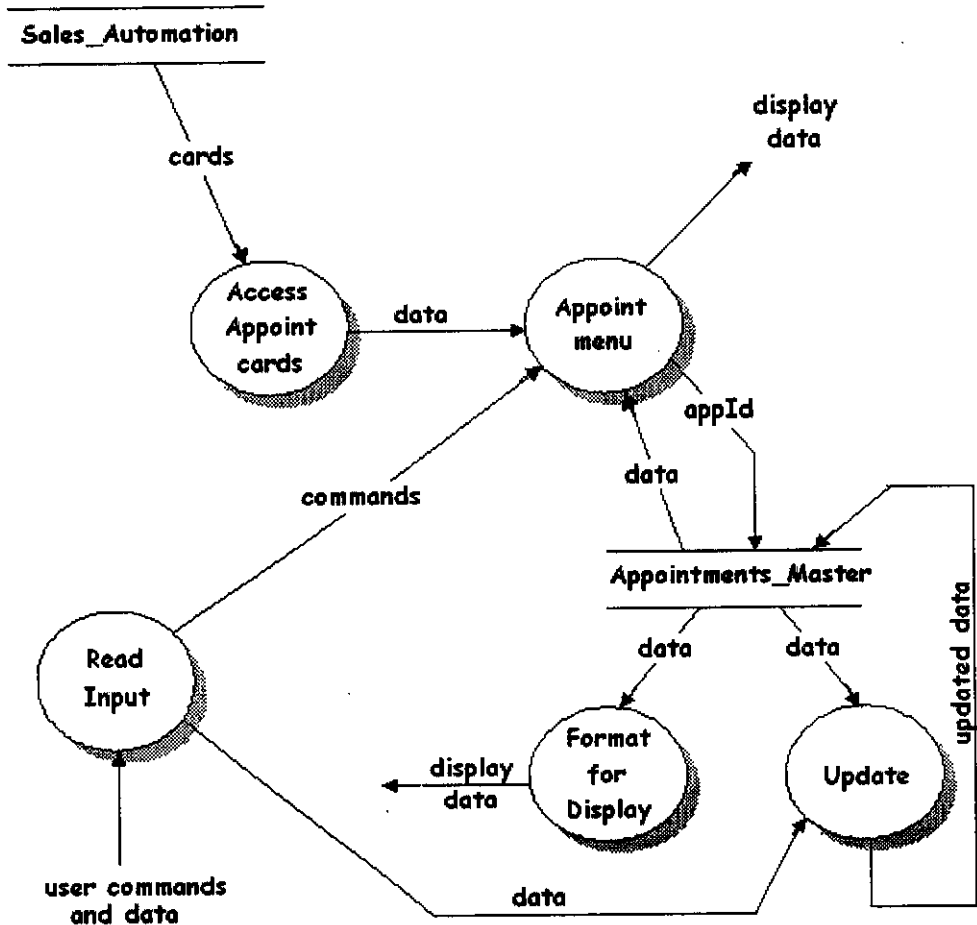


Figure 4.3.4 – Level I DFD for Mobile Client



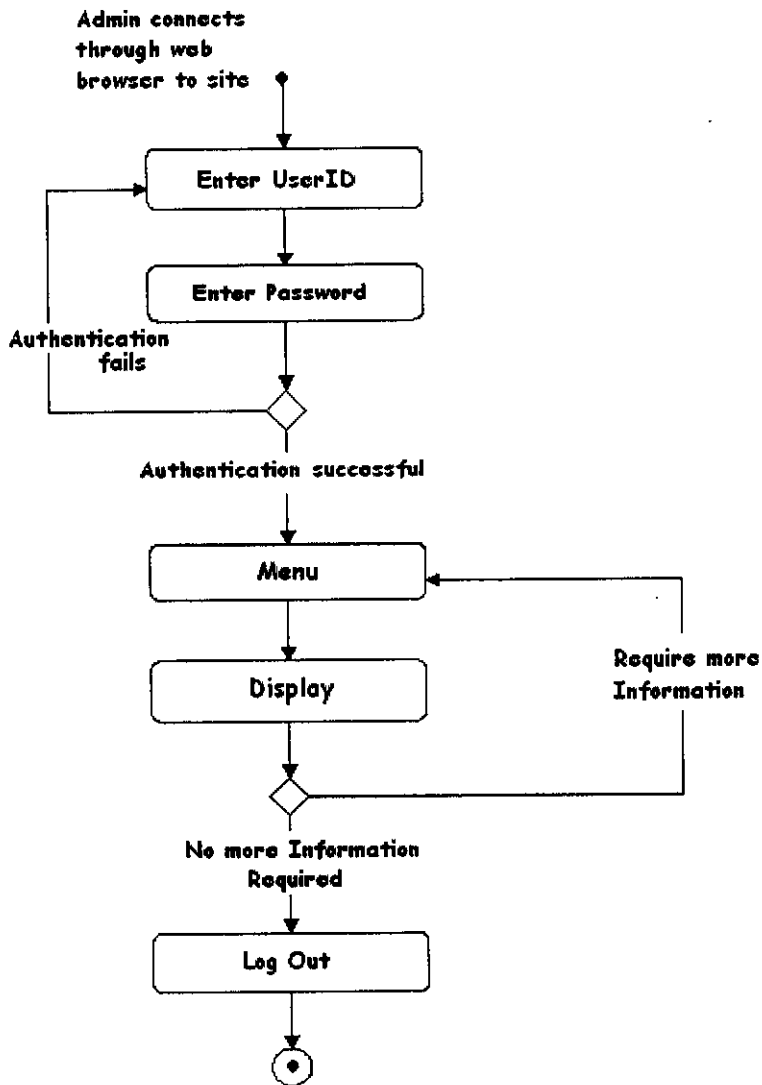
This DFD is common to **Appointment** and **Service** modules in the client side.

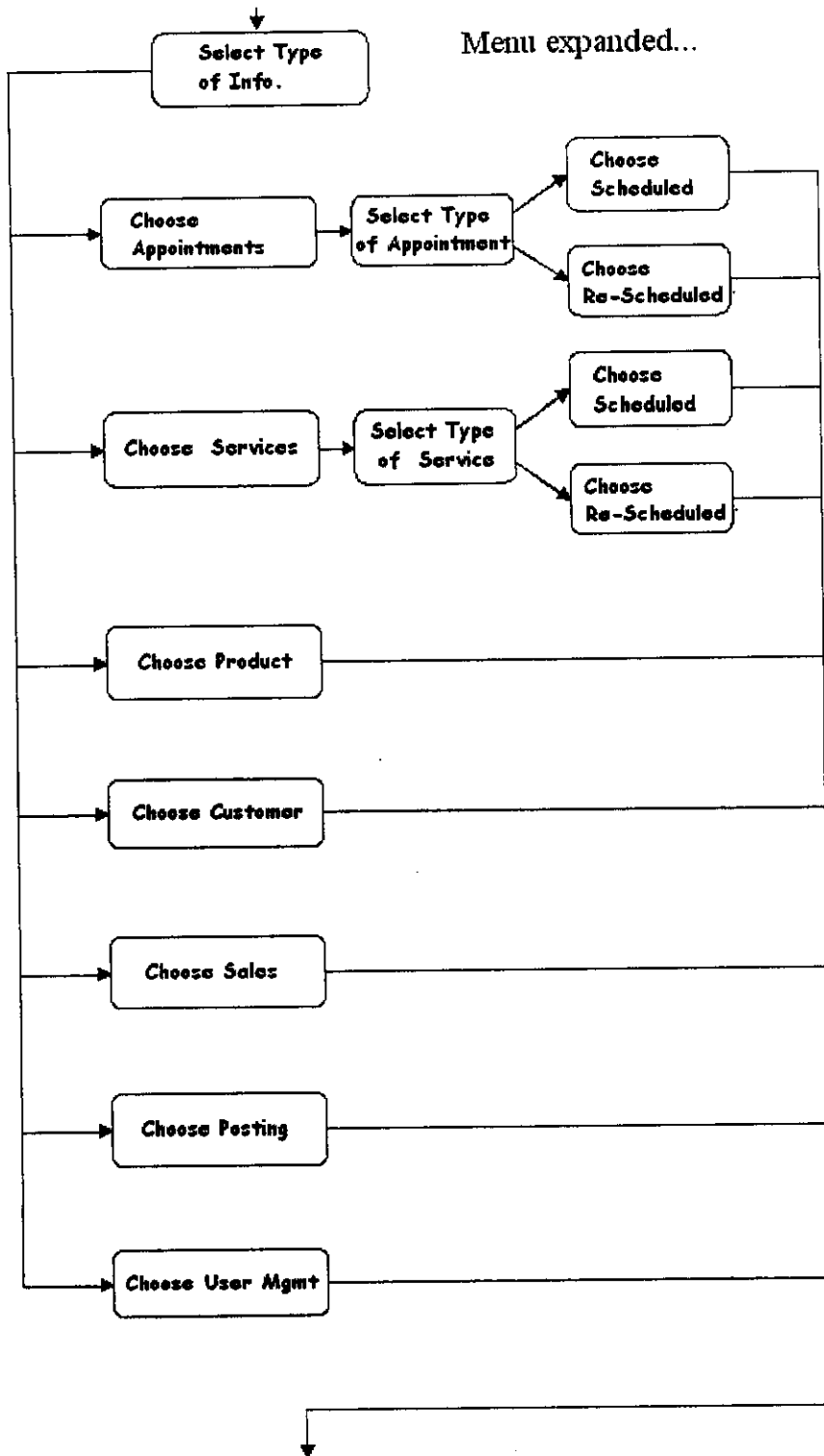
Figure 4.3.5 – Level II DFD in Mobile Client

### 4.3.2. Process flow diagram

Process flow diagram shows the sequence of the process in the system. It shows the sequential and conditional statements in the operational system.

#### ADMIN





Display expanded...

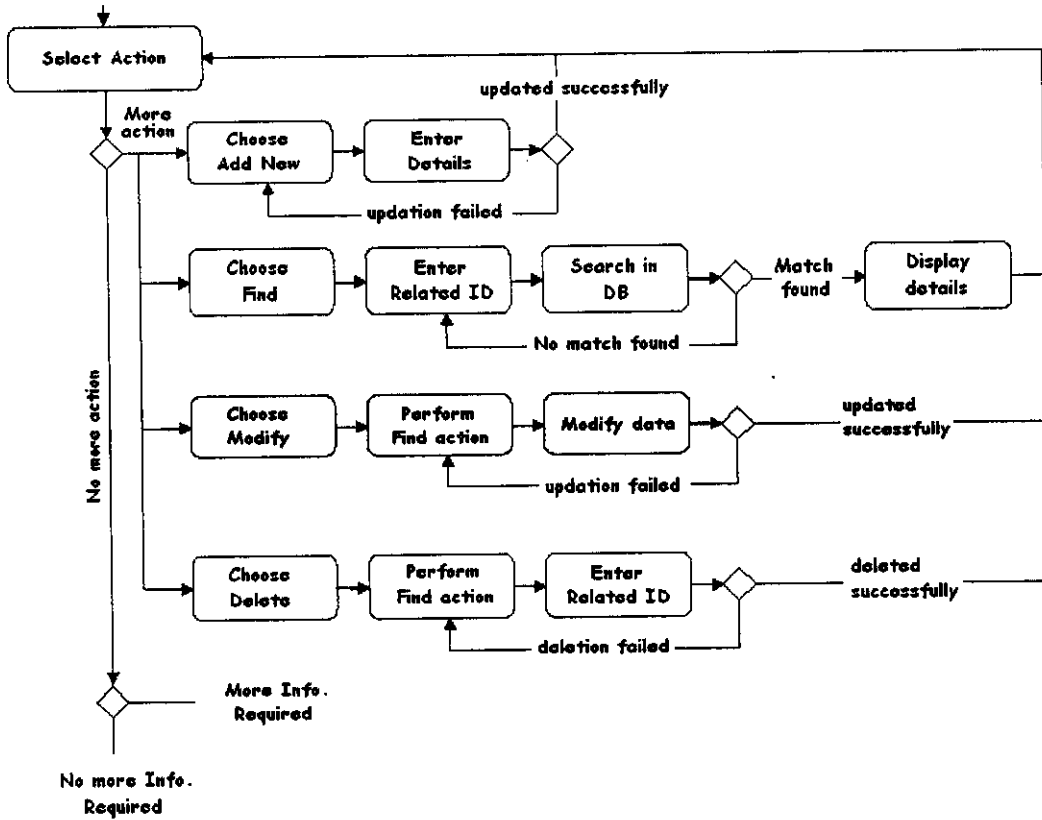
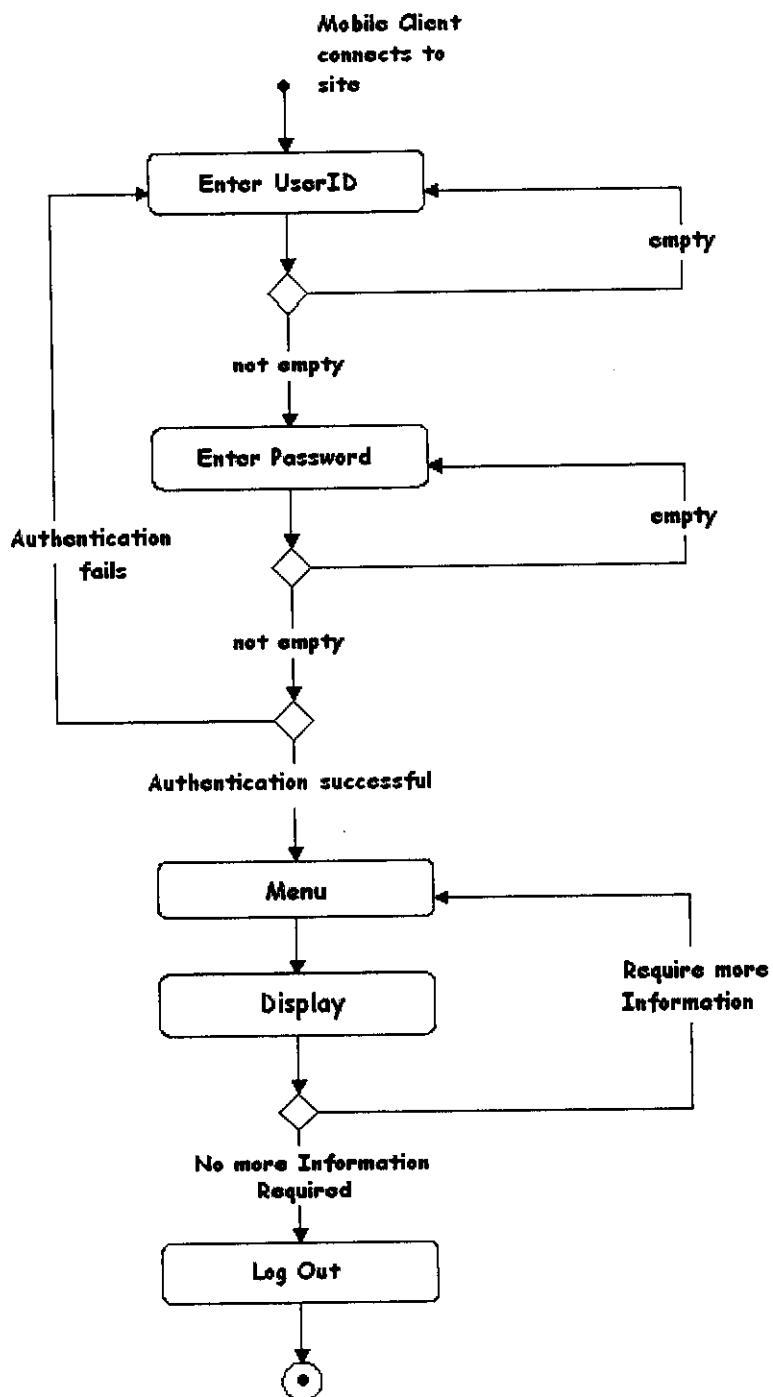


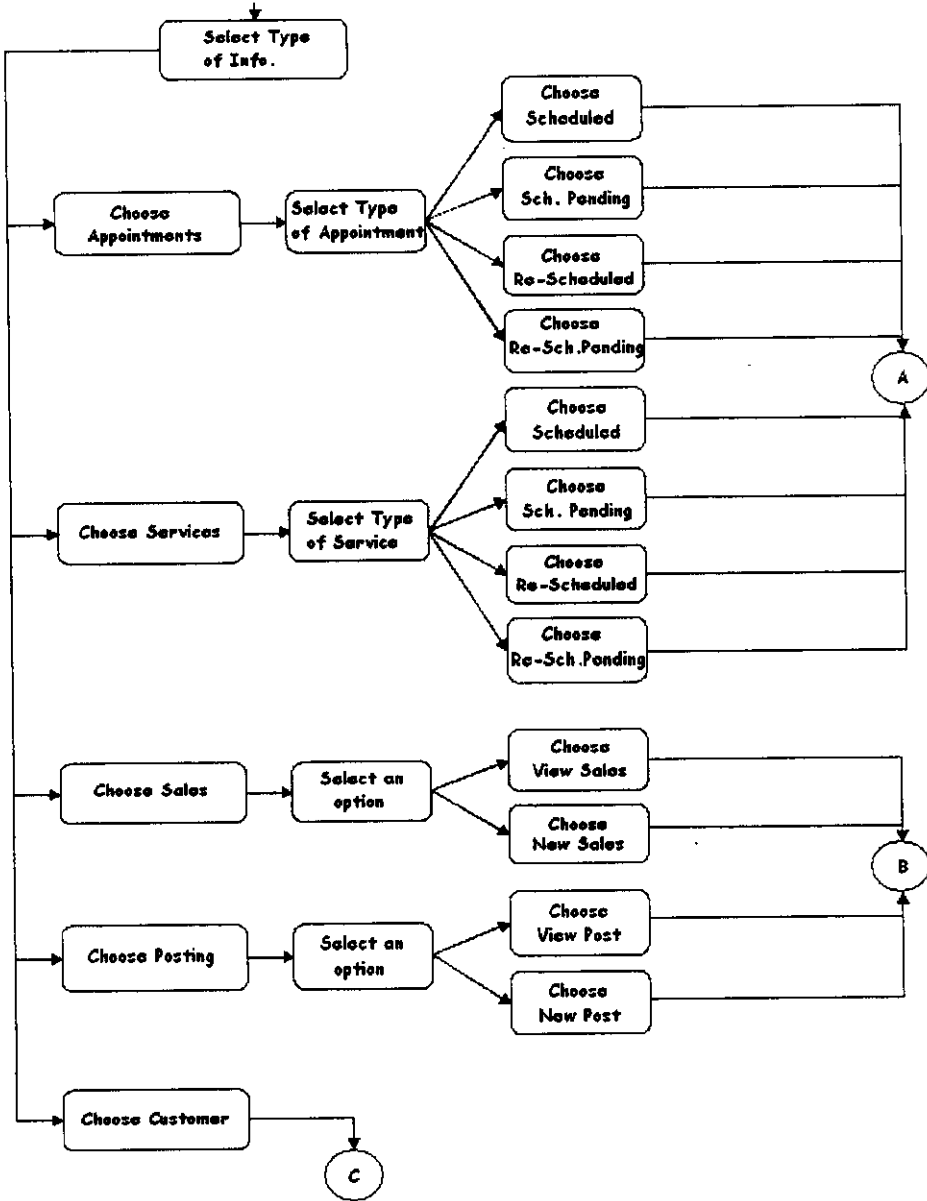
Figure 4.3.6 – Process Flow Diagram For Admin

## MOBILE CLIENT





Menu expanded...



Display expanded...

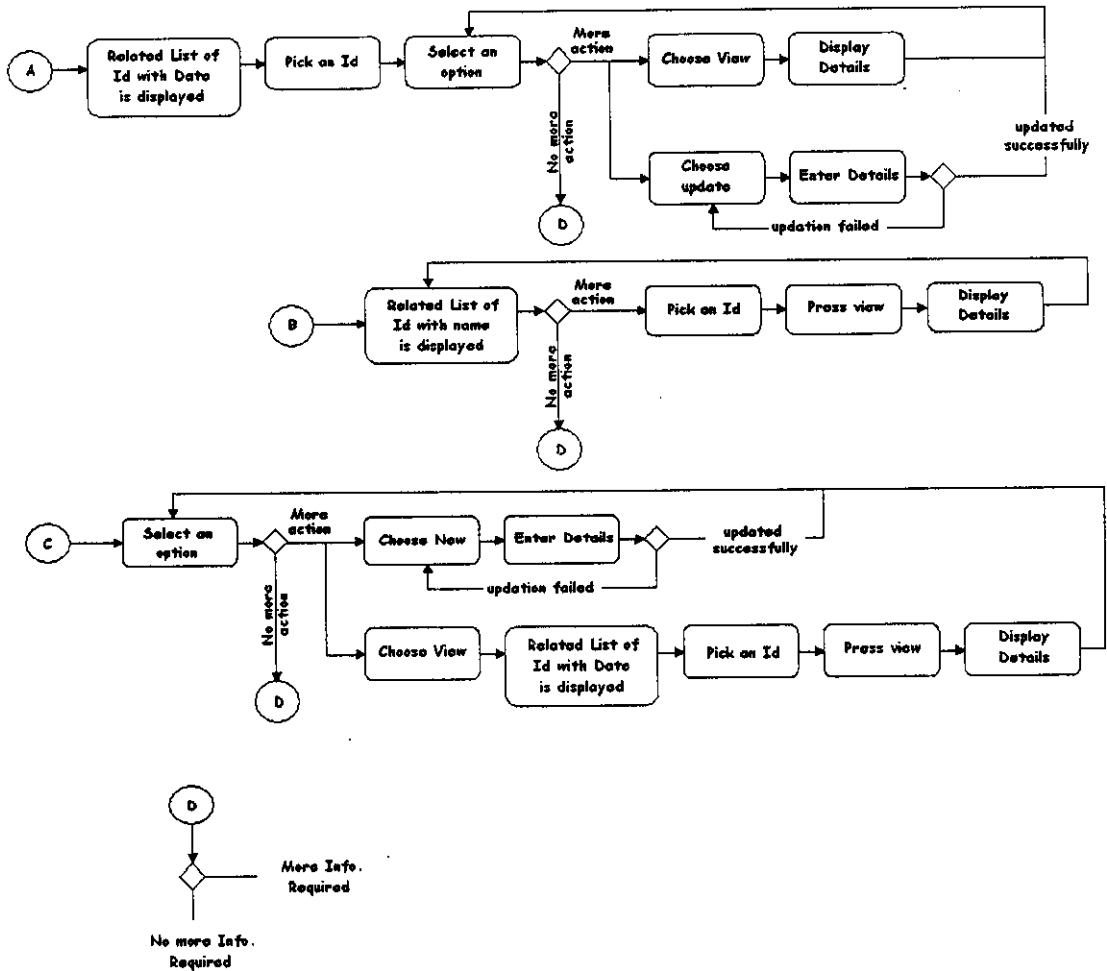


Figure 4.3.7 – Process Flow Diagram For Mobile Client

### 4.2.3 ER Diagram

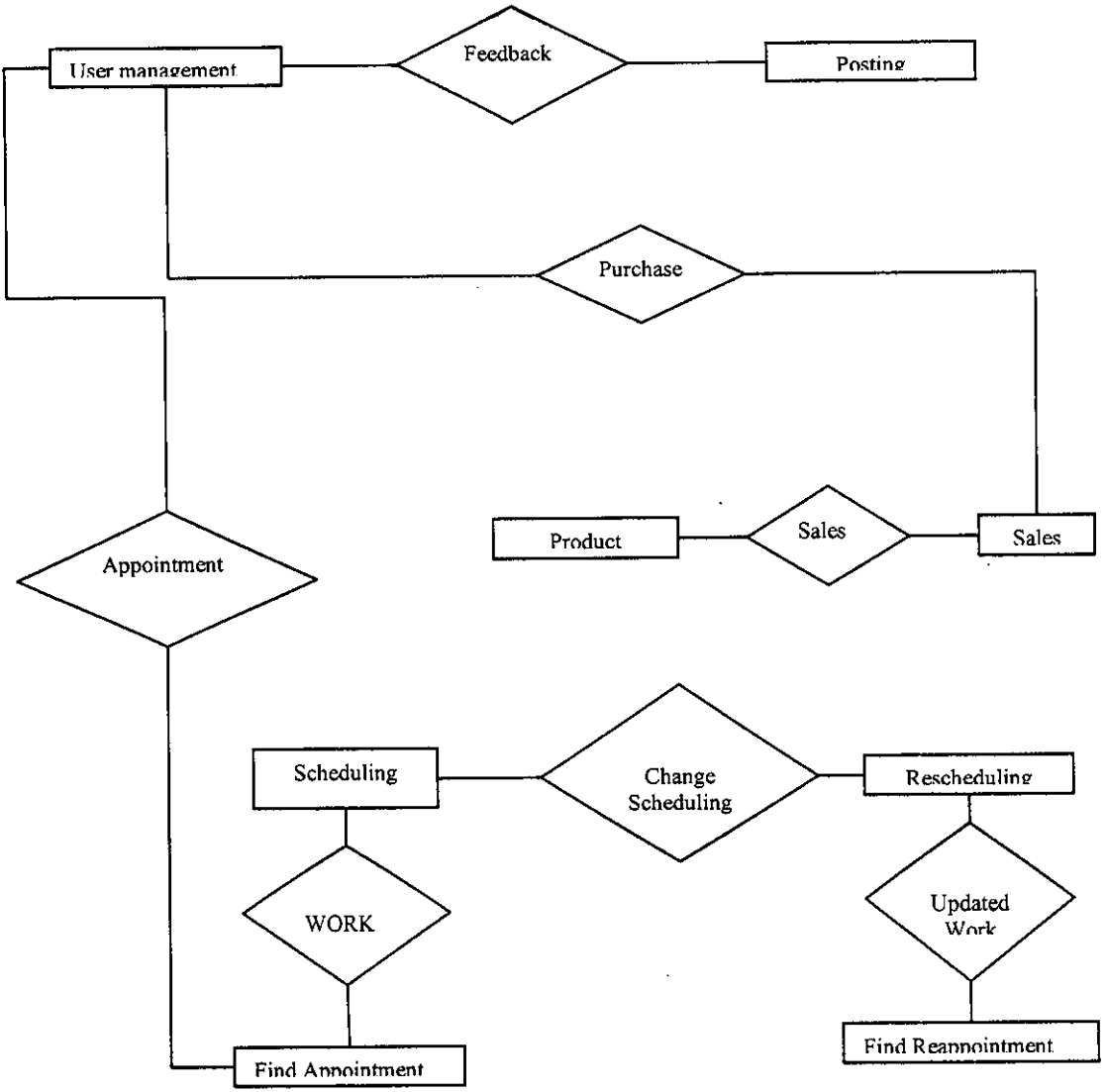


Figure 4.3.8 – ER Diagram

## 4.3. Feasibility System

---

### Introduction:

For any project to be successful there is a need for an effective feasibility study.

The purpose of feasibility study is not to solve the problem but to determine if the problem is worth solving. There are many feasibility studies to be conducted. But there are three main feasibility tests to be performed. They are:

- ◆ Operational Feasibility
- ◆ Technical Feasibility
- ◆ Economic Feasibility

### 4.3.1. Operational Feasibility

During the feasibility analysis, operational feasibility study is a must because according to software Engineering principles, a system should have a high usability.

Taking this project into consideration, proposed system is beneficial only if it can be turned into information system that will meet organization's operating requirements. Here the level of operational feasibility was found to be very high because this reaches a larger group of people.

### 4.3.2. Technical Feasibility

Technical Feasibility is the most difficult area to access at this stage. It is studied to determine if the existing hardware and software has the capacity to run the proposed system. This of course is not sufficient, new hardware and software needs to be brought for the implementation of the proposed system.

### **4.3.3 Economic Feasibility: -**

Economic analysis is the most frequently used method for evaluating effectiveness of the candidate system. More commonly known as cost / benefit analysis, the procedure is to determine the benefits and the savings that are expected to form the candidate system and compare them with the cost and benefits and then a decision is made to design and implement the system. Otherwise further justification of alternations in the proposed system will have to be made if it is to have a chance of being approved.

### **4.4 Justification**

---

The proposed system has been developed with an eye to future developments. It can easily be updated with new modules when required. Data transparency has been maintained in order to cope with the future developments. The proposed system is user friendly and has better interface with the people working with it. The system is judged operationally feasible since

- ✓ Previous system is not accessible in all places but current system is more accessible everywhere.
- ✓ Most of the development work is done using free software's available in the market hence it is economically very feasible.
- ✓ The system designed to run with a new concept using WAP technology and thus in line with technological revolution.

As this system is technically, economically and operationally feasible, thus this system is judged feasible.

## SYSTEM REQUIREMENTS

### 5.1. Software Requirements:

- **Operating System** : Windows XP/2000
- **Tools** : Visual Studio .Net 2003
- **.Net Version** : .Net 1.1/ 2
- **Browser** : Internet Explorer 5.0 or higher
- **Web Server** : IIS 5
- **Database** : SQL Server.

### 5.2. Hardware Requirements:

- WAP enabled mobile device
- GSM facility in mobile device.
- Service provider must provide web access capabilities.
- **Monitors:** 800\*600 minimum resolution at 256 colors minimum.
- **Memory:** Approximately 256MB of a board memory.
- **I/O:** Two or three button mouse and standard 101 key keyboard.
- **Processor:** Pentium IV processor with at least 1.3GHz processor.

### 5.3. Third party Interface / S/W:

Wide variety of websites that provide WAP enabled interfaces.

## **DESIGN AND IMPLEMENTATION**

### **6.1. Input / Output Form Design**

---

There are few possible inputs and outputs in each of the project module existing. They may be briefly explained as,

#### **1) Database Design and Login Screen**

This module involves in designing the database model for a real time system and performing the initial user authentication. If authenticated user, a session is established between the user and remote server (web server) for data transmission.

#### **2) Appointment and Service Screen**

This module maintains information about the Scheduled and Rescheduled Appointments and Services. The admin is given the right to add/modify/delete data while the client can only view data and update status.

#### **3) Product Information and Customer Information Screen**

Here product and customer information can be added / deleted / viewed by admin. The software also allows client to view the product and customer information.

#### **4) Sales Transaction and Posting Screen**

This module allows the admin to add/ modify/ delete the orders placed previously. The admin and client can place new orders. The client is also allowed to view the data. Posting allows the client and admin to pass messages.

## 5) Programming server side using ASP.NET and DB connection maintenance

This involves the establishment of database connection using ADO.NET. It involves in designing of ASP.NET pages which will listen to the WML pages and generate WML tags and response the appropriate WML pages back to the mobile device.

## 6.2. Database Design

---

The database design is used to group data into a number of tables. The tables are organized to

- ◆ Reduce duplication of data
- ◆ Simplicity functions like adding, deleting, modifying data etc.,
- ◆ Retrieving data

In the project we use SQL SERVER to build the necessary tables and also to make the relations. There are a total of 10 tables one for each module. They are: -

1. appointments\_master
2. appointment\_reschedule
3. services\_master
4. service\_reschedule
5. product\_master
6. customer\_master
7. sales\_register
8. post\_master
9. emp\_master
10. uid\_sequence

Each of the above mentioned modules have their own fields and among which there is a primary key with the help of which we can identify and call the table. The database table plays an important role in our project as each and every transaction and



function that will be carried out or done by the user will be updated in the database dynamically with the help of ASP.NET. We use a DSN-less connection.

The database tables have been designed and related to one another in the most convenient manner, which is suitable for our project. The following diagram depicts the table relationship that exists between all the five pivot database tables in the project:

### **Appointments\_master**

<b>Name</b>	<b>Type</b>	<b>Primary Key</b>	<b>Mandatory Field</b>	<b>Sample Data</b>
appointment_id	varchar(15)	Yes	Yes	APP0001
appointment_status	varchar(10)	No	No	Completed
assignedby	varchar(50)	No	No	admin
assignedto	varchar(50)	No	No	ram
completed_date	date	No	No	26/12/2006
completed_desc	varchar(150)	No	No	Appointment Completed
customer_id	varchar(10)	No	No	C0001N
schedule_date	date	No	No	26/12/2006
schedule_desc	varchar(150)	No	No	Appt. scheduled for C0001N

**Appointments\_reschedule**

<b>Name</b>	<b>Type</b>	<b>Primary Key</b>	<b>Mandatory Field</b>	<b>Sample Data</b>
appointment_id	varchar(15)	No	No	APP0002
completed_date	date	No	No	27/12/2006
completed_desc	varchar(150)	No	No	Re-Scheduled appt comp
reschedule_date	date	No	No	27/12/2006
reschedule_desc	varchar(150)	No	No	Rescheduled due to rain
reschedule_id	varchar(10)	YES	YES	RAPP0001
reschedule_status	varchar(50)	No	No	Completed
rescheduled_by	varchar(50)	No	No	admin
rescheduled_to	varchar(150)	No	No	ram

**customer\_master**

<b>Name</b>	<b>Type</b>	<b>Primary Key</b>	<b>Mandatory Field</b>	<b>Sample Data</b>
customer_address1	varchar(50)	No	No	4, Shankar St,
customer_address2	varchar(50)	No	No	Shanthi Nagar,
customer_address3	varchar(50)	No	No	Padi
customer_contactperson	varchar(50)	No	No	Mr. Mohan
customer_id	varchar(15)	Yes	Yes	C0001N
customer_name	varchar(50)	No	No	Suresh Computers
customer_phone	varchar(50)	No	No	26356456

**Emp\_master**

<b>Name</b>	<b>Type</b>	<b>Primary Key</b>	<b>Mandatory Field</b>	<b>Sample Data</b>
emp_id	varchar(50)	Yes	Yes	E0001
emp_name	varchar(100)	No	No	Ram
emp_password	varchar(50)	No	No	Ram
emp_type	varchar(10)	No	No	User

**post master**

<b>Name</b>	<b>Type</b>	<b>Primary Key</b>	<b>Mandatory Field</b>	<b>Sample Data</b>
post_by	varchar(50)	No	No	ram
post_date	Date	No	No	27/12/2006
post_details	varchar(150)	No	No	Re-Appt schedule completed
post_id	varchar(10)	Yes	Yes	P0001
post_status	varchar(10)	No	No	New
post_subject	varchar(50)	No	No	Reg Re-Appt
post_to	varchar(50)	No	No	admin

**product master**

<b>Name</b>	<b>Type</b>	<b>Primary Key</b>	<b>Mandatory Field</b>	<b>Sample Data</b>
product_description	varchar(30)	No	No	Windows XP
product_id	varchar(10)	Yes	Yes	P0001
product_rate	double	No	No	3900.00
product_stock	numeric	No	No	82
product_unit	varchar(15)	No	No	OS

**sales\_register**

<b>Name</b>	<b>Type</b>	<b>Primary Key</b>	<b>Mandatory Field</b>	<b>Sample Data</b>
customer_id	varchar(10)	No	No	C0001N
dateofsale	date	No	No	26/12/2006
product_id	varchar(10)	No	No	P0001
qtysold	int	No	No	20
sale_empid	varchar(50)	No	No	E0001
sale_id	varchar(15)	Yes	Yes	S0001
saledesc	varchar(150)	No	No	Windows XP 20 nos. sold

**service\_master**

<b>Name</b>	<b>Type</b>	<b>Primary Key</b>	<b>Mandatory Field</b>	<b>Sample Data</b>
completed_date	date	No	No	27/12/2006
completed_desc	varchar(150)	No	No	Schedule completed
customer_id	varchar(10)	No	No	C0001N
service_assignedby	varchar(50)	No	No	admin
service_assignedto	varchar(50)	No	No	ram
service_desc	varchar(150)	No	No	Service scheduled for ram
service_duedate	date	No	No	26/12/2006
service_id	varchar(15)	YES	YES	S0001
service_status	varchar(50)	No	No	Completed

**service\_reschedule**

<b>Name</b>	<b>Type</b>	<b>Primary Key</b>	<b>Mandatory Field</b>	<b>Sample Data</b>
completed_date	date	No	No	27/12/2006
completed_desc	varchar(150)	No	No	Re service comp
reschedule_date reschedule_desc	date varchar(150)	No No	No No	27/12/2006 service comp
reschedule_id	varchar(15)	<i>Yes</i>	<i>Yes</i>	RS0001
reschedule_status	varchar(10)	No	No	Completed
rescheduled_by	varchar(50)	No	No	Completed
rescheduled_to	varchar(50)	No	No	ram
service_id	varchar(15)	No	No	S0001

**uid\_sequence**

<b>Name</b>	<b>Type</b>	<b>Primary Key</b>	<b>Mandatory Field</b>	<b>Sample Data</b>
uid_active	char(1)	No	No	Y
uid_currentval	numeric	No	No	0
uid_desc	varchar(30)	No	No	Customer Info
uid_maxno	numeric	No	No	9999
uid_numlen	numeric	No	No	5
uid_prefix	varchar(5)	No	No	c
uid_sequence	varchar(10)	Yes	Yes	Customer
uid_suffix	varchar(5)	No	No	N



## CONCLUSION

The project has been implemented to retrieve time-bounded, business-related data through a mobile phone. A sample inventory of stocks and their respective details such as rate and quantity was created. An UP.SDK was used for modeling the entire process of the WAP implementation. Various software engineering techniques were made use of for the successful completion of WAP-Real Time Business Analysis Using WAP.

Further enhancements can be conveniently induced into the current project such as:

- Implementing the present code in a real mobile device by buying the appropriate services from the mobile phone service providers.
- The project can be used to access other dynamic information related to areas of sports, city event updates, train and flight timings, matrimonial, weather reports etc. with ease.
- We can also incorporate MMS(Multimedia Messaging Services).
- All web-enabled sites can be accessed through the device.
- Traffic monitoring can be done through the mobile phone.

## Sample Code

### Appointment.wml

```
<wml>
  <head>
    <meta forua="true" http-equiv="Cache-Control" content="max-age=0"/>
  </head>
  <card id="APPID">
    <big>CONNECTING TO WEB SERVER</big>
    
    
    
    
  </card>
</wml>
```

```

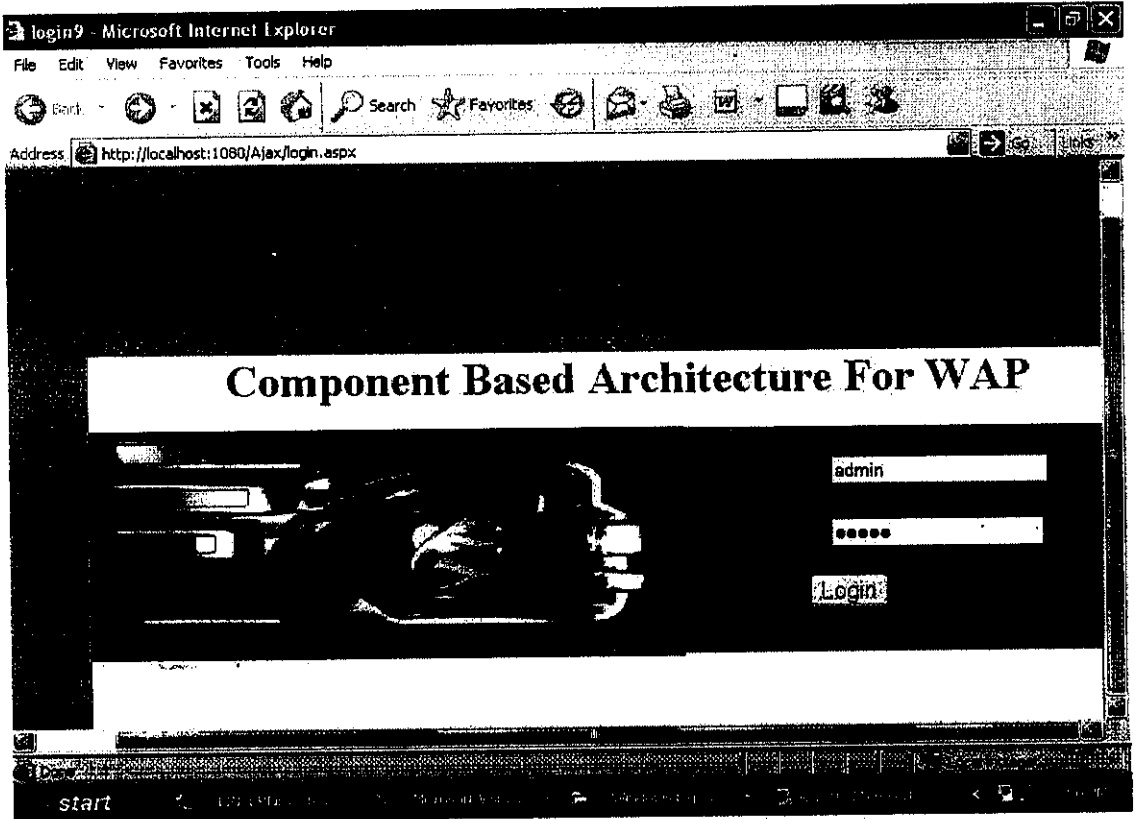
<do type="accept" label="ENTER">
  <go href="../Appointment.jsp" method="post">
    <postfield name="ACTION" value="1"/>
    <postfield name="appId" value="$(appId)"/>
    <setvar name="appId" value=""/>
  </go>
</do>
</card>
<card id="GETAPPCOMPLETEDDATE">
<br/>
  <big>Completed Date : (dd/mm/yyyy) : </big>
  <br/>
  <input type="name" name="completed_Date"/>
  <do type="accept" label="VALIDATE">
    <go href="../wmls/Appointments.wmls#verifyCompletedDate()"/>
  </do>
</card>
<card id="GETAPPCOMPLETEDDESC">
  <big>Completed Desc :</big>
  <br/>
  <input type="name" name="completed_Desc"/>
  <do type="accept" label="VALIDATE">
    <go href="../wmls/Appointments.wmls#verifyCompletedDesc()"/>
  </do>
</card>
<card id="UPDATE">
  <do type="accept" label="ENTER">
    <go href="../AppointmentBean.jsp" method="post">
      <postfield name="ACTION" value="2"/>
      <postfield name="appId" value="$(appId)"/>
      <postfield name="appstatus" value="$(appstatus)"/>
    </go>
  </do>
</card>

```

```
<postfield name="appCompletedDate" value="$(completedDate)"/>
<postfield name="appCompletedDesc" value="$(completedDesc)"/>
<setvar name="appId" value=""/>
<setvar name="completedDate" value=""/>
<setvar name="completedDesc" value=""/>
<setvar name="completed_Date" value=""/>
<setvar name="completed_Desc" value=""/>
<setvar name="appstatus" value=""/>
</go>
</do>
<big>Select Appointment Status:</big>
<select name="appstatus">
  <option value="pending">Pending</option>
  <option value="completed">Completed</option>
</select>
</card>
</wml>
```

# Screen Shots

## Admin Side Screen Shots.....



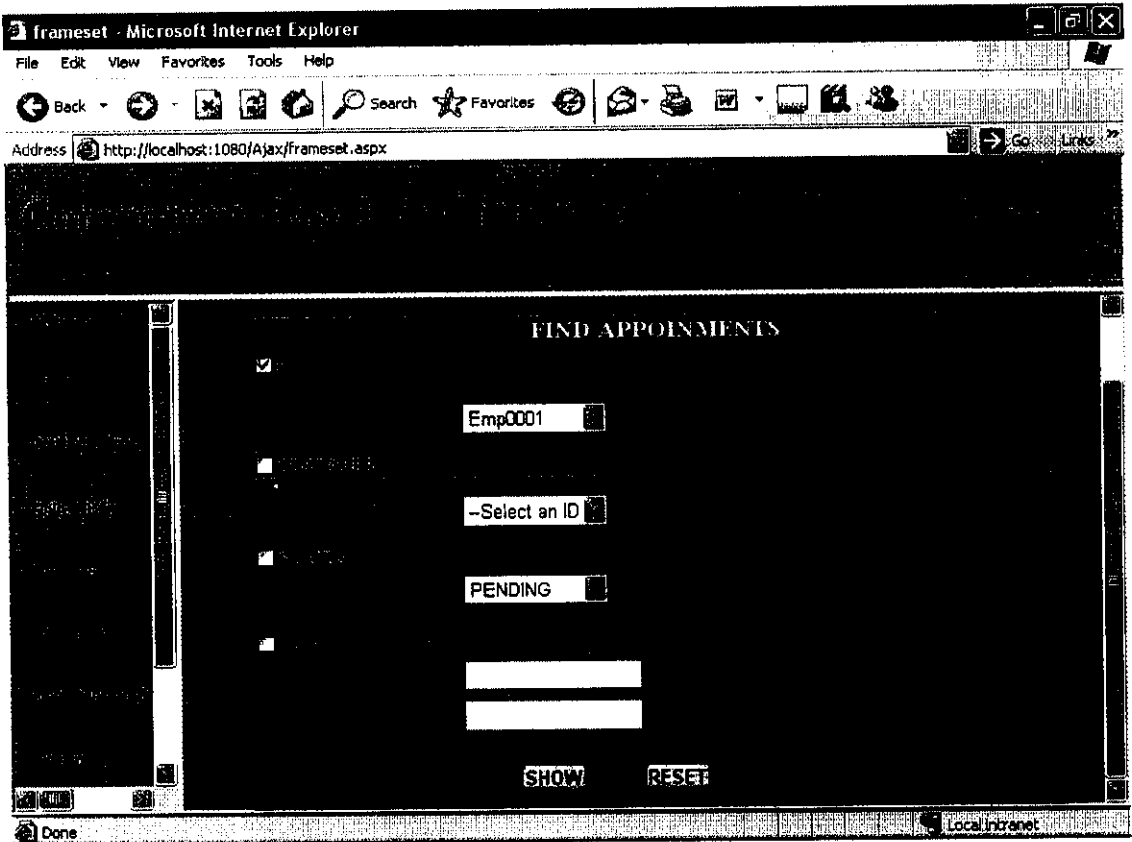
## Scheduled Appointments

The screenshot shows a Microsoft Internet Explorer browser window displaying a web application titled "frameset - Microsoft Internet Explorer". The address bar shows the URL "http://localhost:1080/Ajax/frameset.aspx". The main content area is titled "SCHEDULED APPOINTMENTS" and contains several input fields and buttons. The fields are arranged in a form that appears to be for editing or creating an appointment. The fields contain the following values:

- SA00003
- Emp0001
- Emp0002
- C00001
- 2009-03-30T00:00:00
- Give details about product
- PENDING

The browser's status bar at the bottom shows "Done" and "Local intranet". The Windows taskbar at the very bottom shows the "start" button and several open applications, including "CT\INTERN...", "Microsoft...", "Window...", "Home...", "DocView...", and "USB PM".

## Find Appointments



The screenshot shows a Microsoft Internet Explorer browser window with the address bar displaying `http://localhost:1080/Ajax/frameset.aspx`. The main content area features a form titled "FIND APPOINTMENTS". The form includes several input fields and buttons:

- A checkbox that is checked.
- A text input field containing "Emp0001".
- A dropdown menu with the text "--Select an ID".
- A text input field containing "PENDING".
- Two empty text input fields stacked vertically.
- Two buttons labeled "SHOW" and "RESET" at the bottom of the form.

The browser's status bar at the bottom shows "Done" and "Local intranet".

frameset - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites

Address <http://localhost:1080/Ajax/frameset.aspx>

appointment_id	assigned_to	assigned_by	customer_id	schedule_date	appointment_status	appointment_c

Done Local intranet

## Rescheduled Appointments

frameset - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites

Address http://localhost:1080/Ajax/frameset.aspx

---

### RE SCHEDULED APPOINTMENTS

RA00001	RA00003
SA00001	<input checked="" type="checkbox"/>
Emp0001	
Emp0002	
2009-12-09T00:00:	
vzbm	
pending	

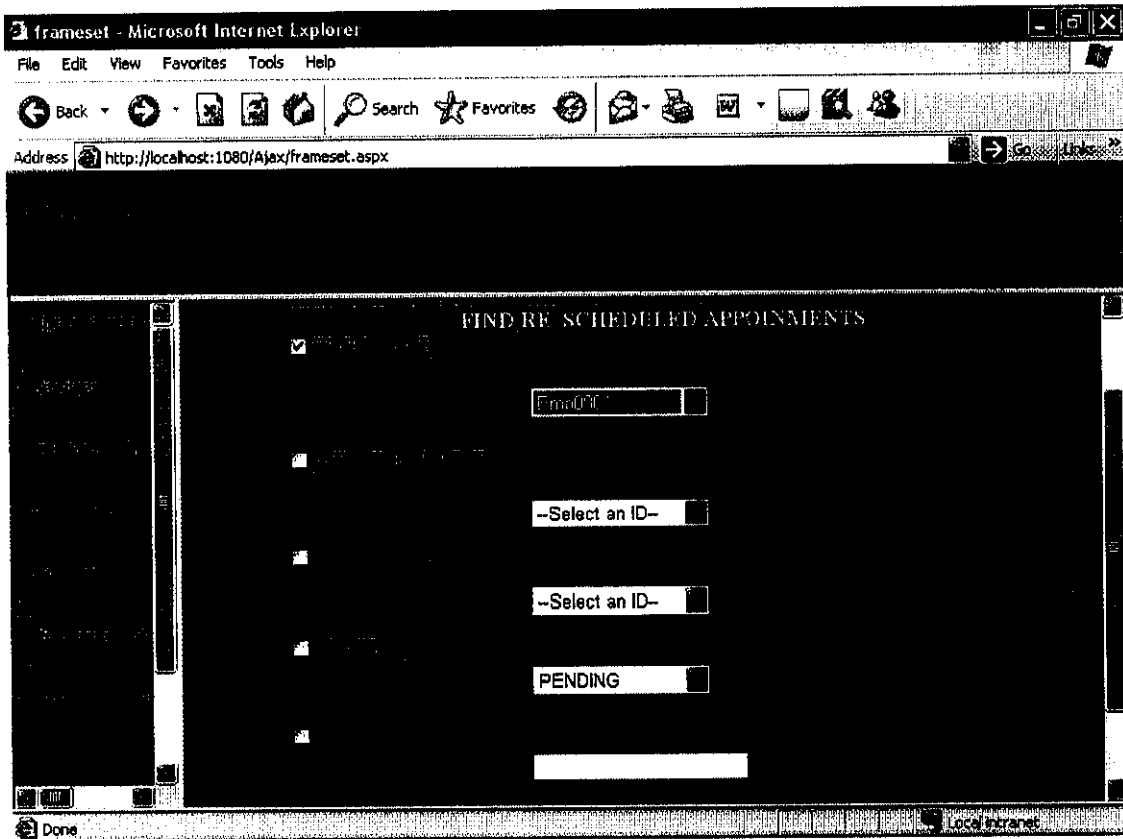
ACTION

ADD NEW MODIFY DELETE FIND RESET

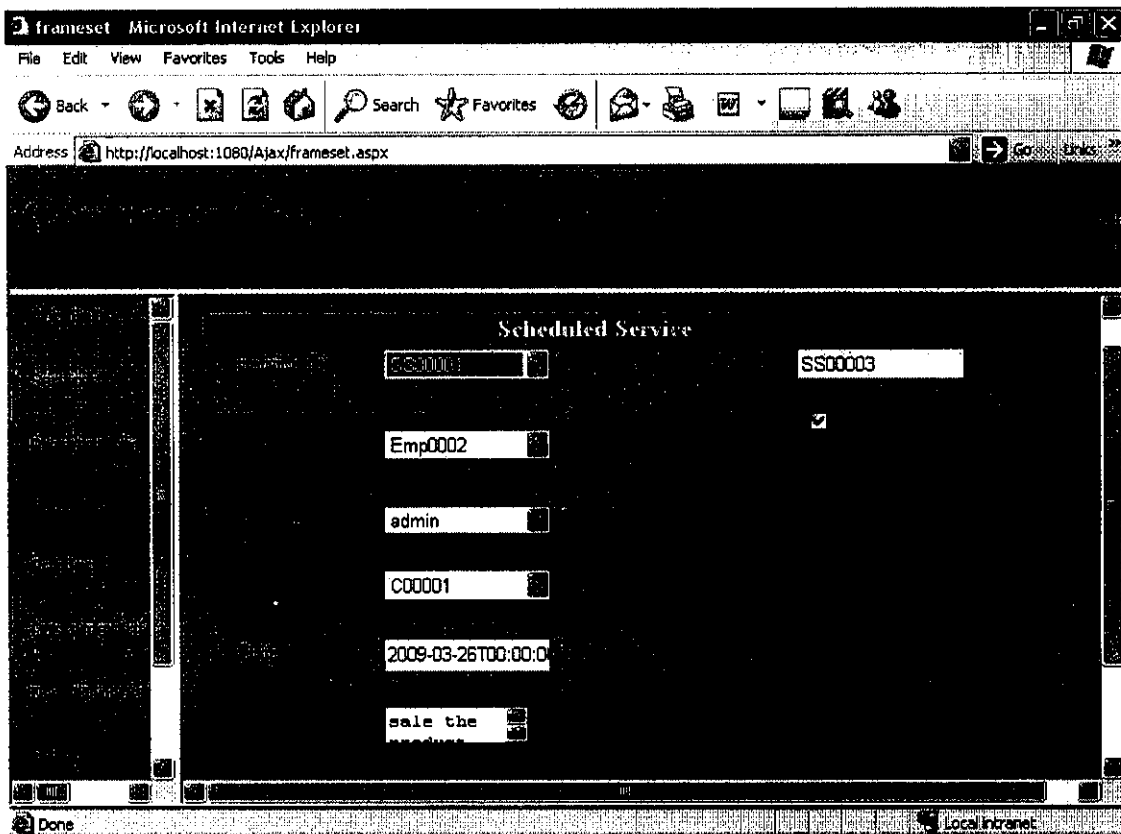
Done Local Internet



## Find Rescheduled Appointments



## Scheduled service



frameset - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites

Address <http://localhost:1080/Ajax/frameset.aspx>

service_id	customer_id	service_assignedto	service_assignedby	service_moderate	service_status	serv

Done

## Rescheduled Services

frameset - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Search Favorites

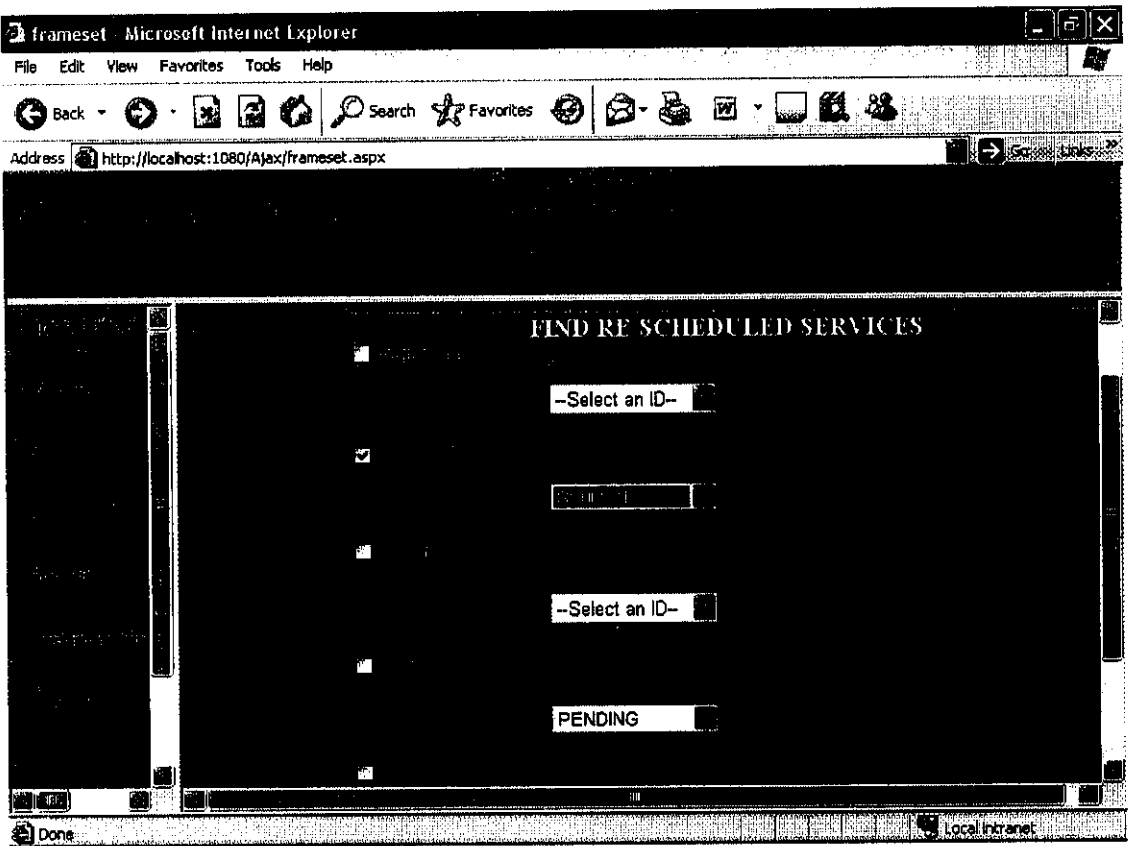
Address http://localhost:1080/Ajax/frameset.aspx

RF SCHEDULED SERVICES

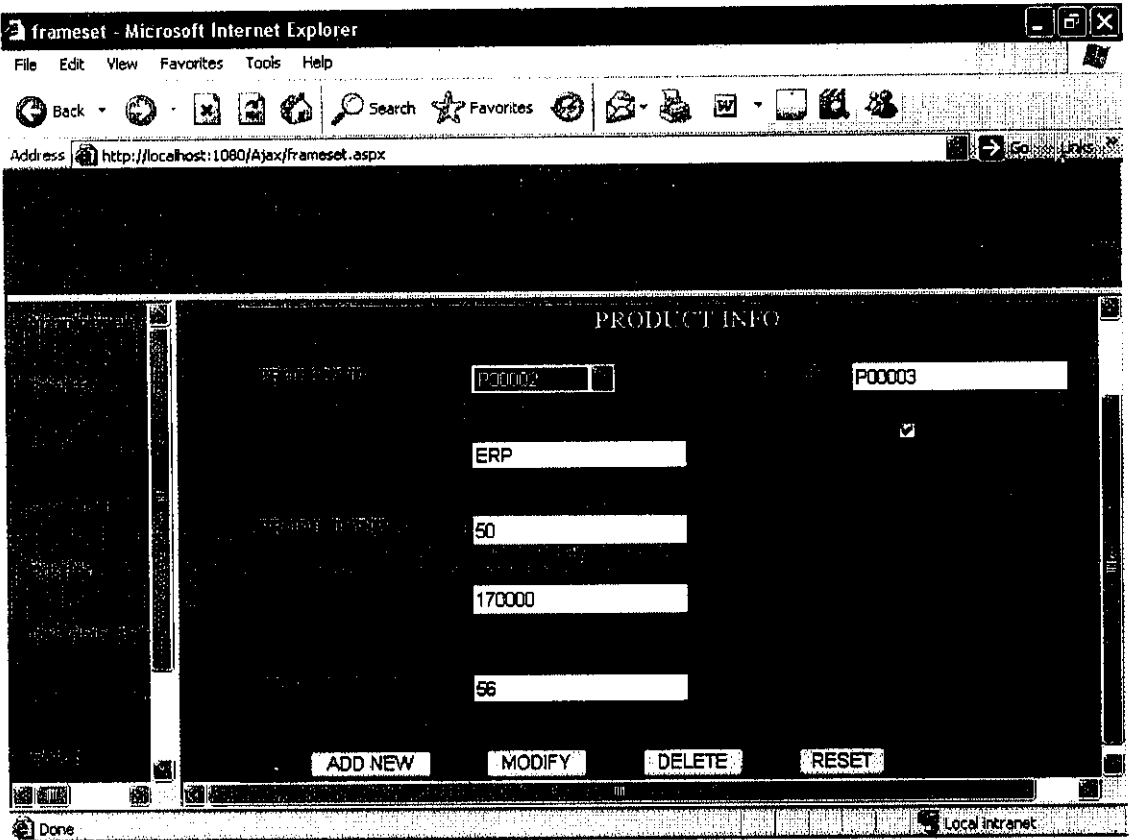
EMPLOYEE ID	--Select an ID--	RS00001
SS00001		
EMPLOYEE ID	Emp0001	
Emp0002		
DATE	12/03/09	
	meet the customer	
	pending	

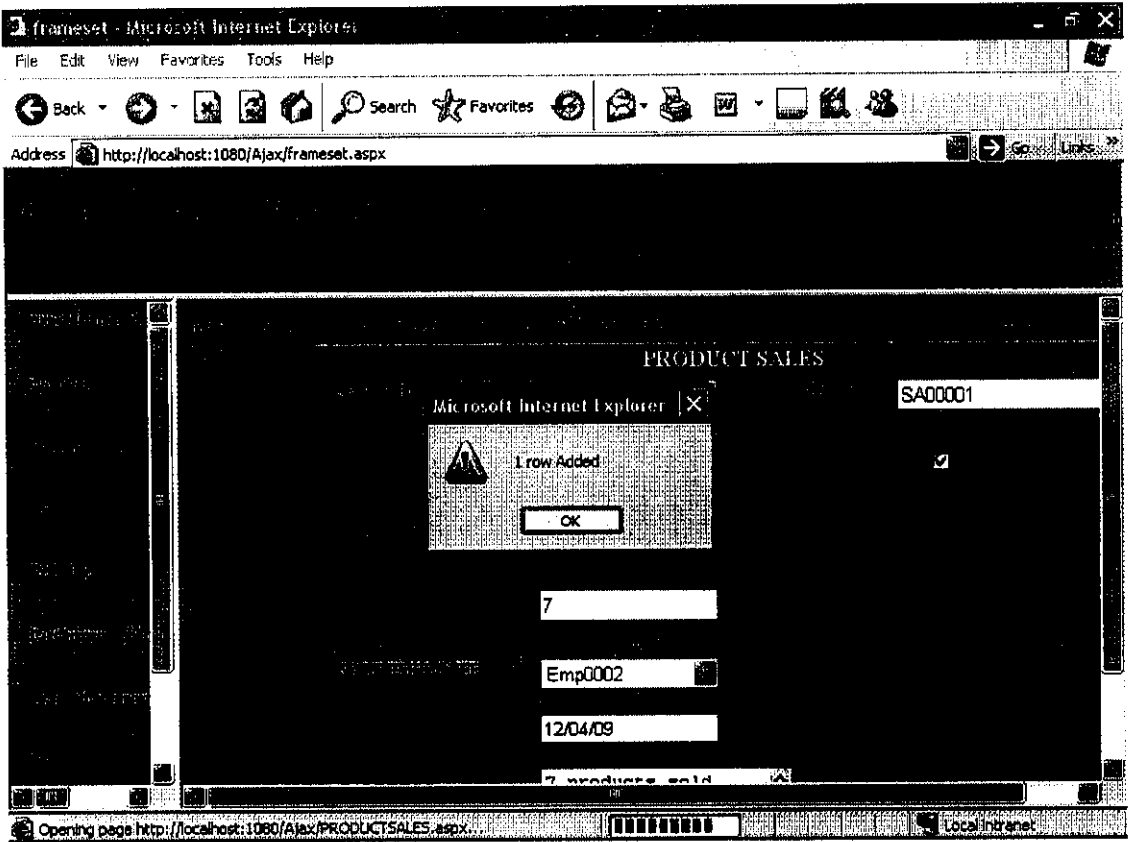
Done Local intranet

## Find Rescheduled Services



# Product Information





## View Post

frameset - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites

Address http://localhost:1080/Ajax/frameset.aspx

---

### View Post

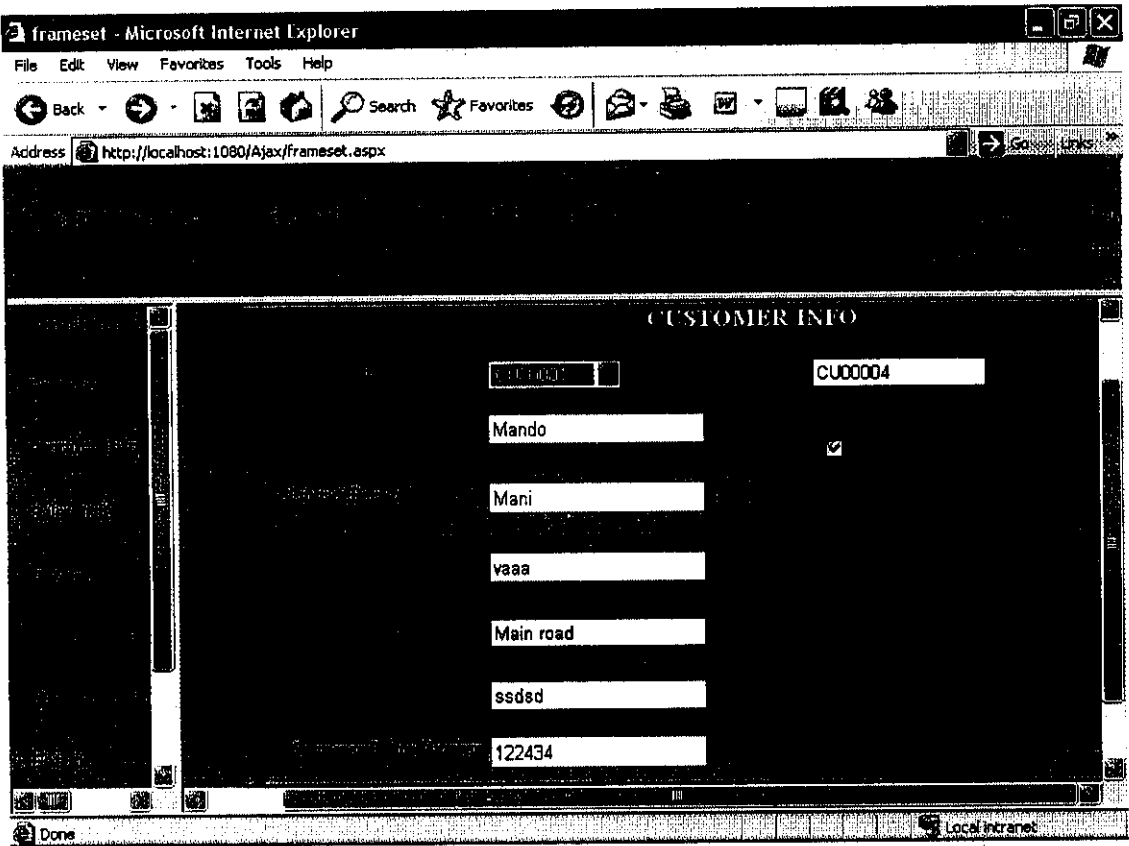
<input type="text" value="Emp001"/>	<input type="text" value="P00002"/>
<input type="text" value="admin"/>	<input checked="" type="checkbox"/>
<input type="text" value="Emp0001"/>	
<input type="text" value="2009-03-30T00:00:00"/>	
<input type="text" value="about project"/>	
<input type="text" value="sold project"/>	

ADD NEW    MODIFY    FIND    RESET

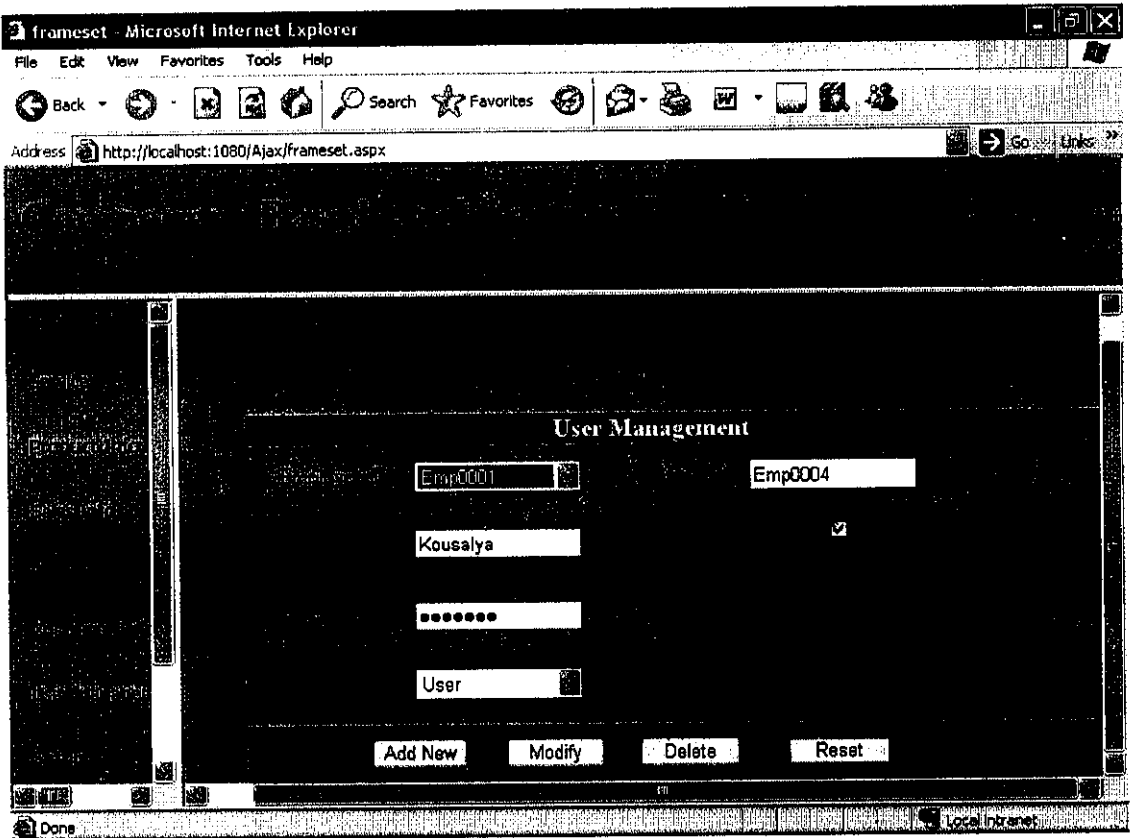
Done Local Intranet



# Customer Information



## User Management



## UId sequence

frameset - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites

Address http://localhost:1080/Ajax/frameset.aspx

UId Sequence

Appointments Scheduled Appointments

SA

2

99999

7

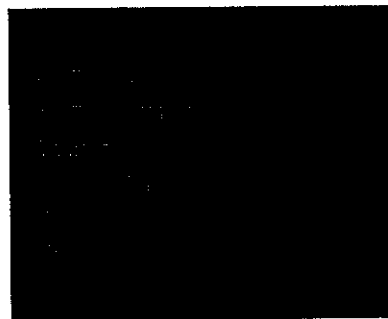
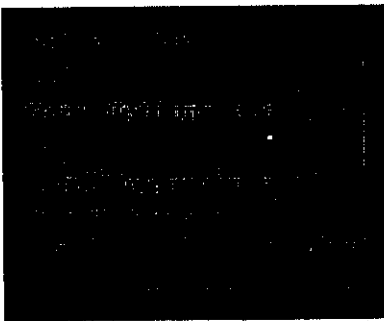
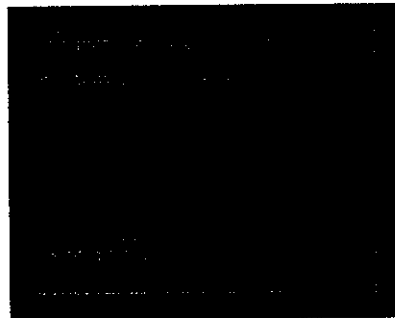
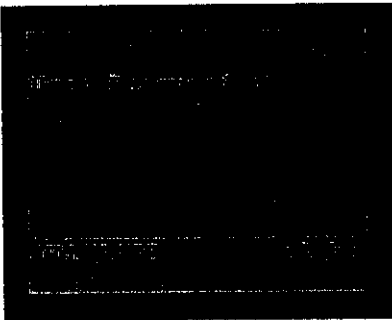
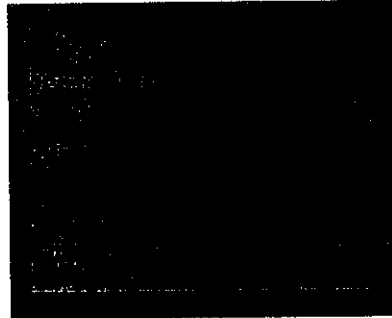
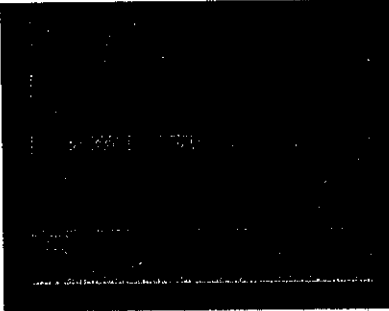
INSERT UPDATE DELETE CLEAR

Select	uid sequence	uid desc	uid prefix	uid suffix	uid current val	uid max no	uid num

Local intranet

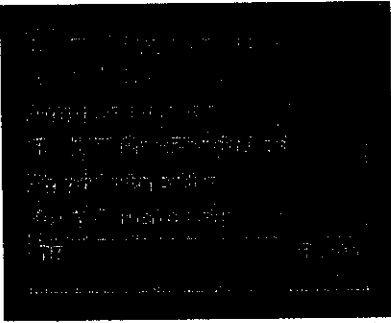
## Client Side Snap Shots.....

### Login Process

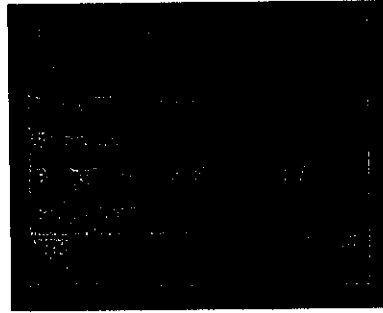


Appointment snap

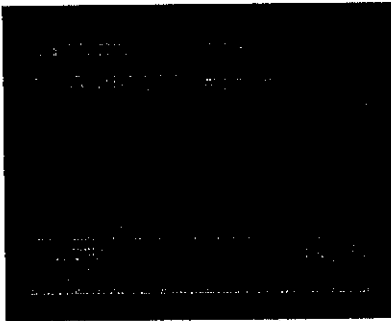
Service snap



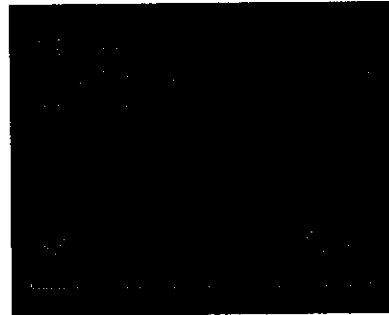
Customer Information



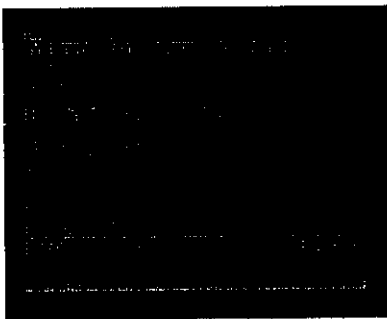
Sales Transaction Information



Posting Management



Logging Out...



## REFERENCES

- Component-Based Architecture for the Development and Deployment of WAP-Complaint Transactional services, IEEE 2001.
- MSDN for Visual Studio .NET 2003
- Ctles06m08, Software Requirement Specification (SRS), Cosmosoft Technologies Limited.
- WAP: Present and Future, Vijay Kumar, Srinivas Parims and Dharma P. Agarwal, IEEE 2003.
- Cosmosoft Online Library.
- <http://www.developershome.com/wap/wapServerSetup/tutorial.asp>
- [www.wapforum.org](http://www.wapforum.org)
- [www.computer.org/portal/site/ieeecs](http://www.computer.org/portal/site/ieeecs)
- WAP 2.0 Technical White Paper, 2002.
- [www.w3schools.com/wap/](http://www.w3schools.com/wap/)
- <http://www.wirelessdevnet.com/channels/wap/training/wml.html>
- <http://notess.com/write/archive/200008ww.html>