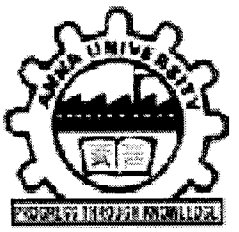
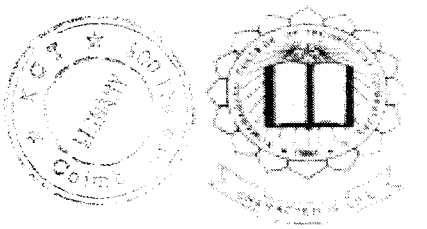




APPOINTMENT SCHEDULING SYSTEM FOR PATIENTS

By

R.KANNAN

Reg No: 71202621018

of

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641006

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 OFCOMPUTER APPLICATIONS

June, 2005

BONAFIDE CERTIFICATE

Certified that this project report titled

APPOINTMENT SCHEDULING SYSTEM FOR PATIENTS

Is Bonafide work of

Mr.R.KANNAN (Reg. No: 71202621018)

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.

18/5/05

Project Guide

Head of the Department

We examined the Candidate with University Register No. 71202621018

in the project Viva-Voce examination held on 25.05.2005

Internal Examiner

External Examiner

ABSTRACT

The Project Titled "Appointment Scheduling System for Patients" is Code Named as ASSP, It is Application Software developed for a Hospital for monitoring the Scheduling Starting from Appointments and up to the Bill Production.

Appointment Scheduling System for Patients is an integral part of daily work for healthcare professionals, from family practices to large clinics, from physician offices to hospitals. Medical office staff has to schedule patients for dentist, optometrist, ophthalmologist, surgeon, psychiatrist and other general care and specialist appointments. Scheduling quickly becomes complicated if the doctor is practicing in several clinics and travels between medical offices.

The Appointment Scheduling System for Patients is used maintains the data's of Appointment details. Appointment Scheduling System for Patients stores the records of patients, doctors, appointment details, equipment details, room details and information about the data manager and reminders about the expiry date of appointment.

The major goal of the system is used to maintain the details of appointments Information's, it gives the consultation records up to date for the Doctors and their visiting time about their Consultations and the equipments and rooms Information for Hospitals. If new patients come to hospital, Medical staffs will be registering his personal information and he will choose the type of service needed by the patient. Appointment Scheduling System for Patients is developed according to the requirements of the concern monitor their entire schedule and to provide current and accurate information to their patients.

ACKNOWLEDGEMENT

Successful completion of this study not only belongs to me, but also to a lot of people who stand behind the stage.

First, I thank my parents who gave me full support both economically and morally.

I thank **Dr.K.K.Padmanaban**, Principal, Kumaraguru College of Technology, Coimbatore, who has given an efficient infrastructure that aided me during the course of my education.

I thank **Dr. S.Thangasamy,** Head of the Department of Computer Science and Engineering for his support, valuable guidance and persistent encouragement towards the completion of this project.

I extend my sincere thanks to **Mr. A. MuthuKumar MCA, M.Phil,** Project Coordinator for rendering us all the timely helps through out the project.

I regard my heartfelt thanks and everlasting gratitude to my Project Guide **Mrs. L.Jayashree M.E,** Senior Lecturer, Department of Computer Science & Engineering for her uplifting ideas, inspiring guidance and valuable suggestions which have been very helpful in refining upon the project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Appointment Scheduling System for Patients is an integral part of daily work for healthcare professionals, from family practices to large clinics, from physician offices to hospitals. Medical office staff has to schedule patients for dentist, optometrist, ophthalmologist, surgeon, psychiatrist and other general care and specialist appointments. Scheduling quickly becomes complicated if the doctor is practicing in several clinics and travels between medical offices. Appointments need to be coordinated, medical support staffs have to be constantly aware of all new patients and doctor's schedule.

1.1 PROJECT OVERVIEW

Appointment Scheduling System for Patients stores the records of patients, doctors, appointment details, equipment details, room details and information about the data manager and reminders about the expiry date of appointment. Appointment Scheduling System for Patients is developed according to the requirements of the concern monitor their entire schedule and to provide current and accurate information to their patients.

Major Modules in Appointment Scheduling System for Patients are

- Administration
- Doctors
- Patients
- Equipments and rooms
- Appointments

1.1.1 Project Scope

- Scheduling the appointments and generating reports.
- Checking the availability of Doctors by giving date and time.
- Medical office staff can easily manage doctors' appointment schedules.
- Appointment scheduling system for patients Schedule can be used in clinics and hospitals requiring personnel appointment scheduling.
- Maintaining doctors, patients, and equipment and rooms details efficiently.
- Reduces cancellations of appointments.
- Increases Patients volumes and improves patient and doctors satisfaction.

1.2 ORGANIZATION PROFILE

Kaveri Infosys is the Indian strategic partner of the Infosys, Inc. Chicago. IL. U.S.A. Kaveri Infosys is one of the leading providers of Windows-based software solutions for the healthcare continuum. Incorporated in November 1995 and situated in Chennai.

Vision

Great product Great place Great people

Focus

The company has been totally focused on software development for the healthcare industry. It has all along been a leading provider of integrated software system for healthcare automation services. They are committed to provide error free software through continual improvement. They maintain highly competent human resources so that the customer's requirements are always met.

Services

The Company is currently providing software product development and research and development services to their Principal InfoSys, Inc., US. The future plan will include offering services across the entire spectrum of technology-based services.

They improve the relationship among Patients, Provider and insurers by delivering quality products and services to managed healthcare services continuum and enabling them to expedite the availability of high quality, cost-effective and on time medical services.

Products

Kaveri Infosys solutions are designed to eliminate fragmentation of information management through a revolutionary systems architecture that accommodates unique clinical, financial and administrative business processes found in various segments of the ambulatory and post-acute continuum. Kaveri Infosys healthcare software suites include:

a. **MedSys Ambulatory/Acute** applications for physicians, rehab/therapy facilities for small hospitals

b. **HomeSys Post-acute** solutions comprised of long-term care, home healthcare and hospice configurations.

CHAPTER 2

SYSTEM REQUIREMENT AND SPECIFICATION

The Software Requirements Specification is a technical specification of requirements for the software product. The goal of software requirements definition is to completely and consistently specifies the technical requirements for the software products in a concise and unambiguous manner.

The Software Requirements Specification is based on the system definition high-level requirements specified during initial planning are elaborated and more specific in order to characterize the features that the software product will incorporate .The requirement specification is primarily concerned with functional and a performance aspect of the software product and emphasis is placed on specifying product characteristics is placed on specifying product characteristics without implying how the product will provide those characteristics.

Desirable properties of a Software Requirement Specification

- Correct
- Complete
- Consistent
- Unambiguous
- Functional
- Verifiable
- Traceable.

2.1 HARDWARE REQUIREMENTS

- Processor : Intel Pentium
- Processor : 500 MHZ
- Memory : 96 MB RAM
- Hard Disk : 10 GB

2.2 SOFTWARE REQUIREMENTS

- Operating System : Windows 2000/xp/NT Server
- Programming Language : VB.NET
- Database Server : Microsoft SQL Server 2000

2.3 SOFTWARE OVERVIEW

What is Microsoft .NET?

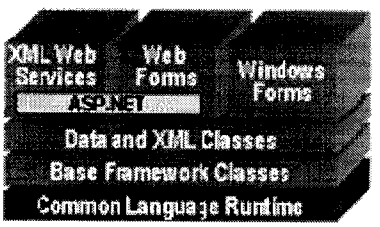
Microsoft .NET is software that connects information, people, systems, and devices. It spans clients, servers, and developer tools, and consists of:

- The .NET Framework used for building and running all kinds of software, including Web-based applications, smart client applications, and XML Web services—components that facilitate integration by sharing data and functionality over a network through standard, platform-independent protocols such as XML (Extensible Markup Language), SOAP, and HTTP.
- Developer tools, such as Microsoft Visual Studio® .NET 2003 which provides an integrated development environment (IDE) for maximizing developer productivity with the .NET Framework.

- A set of servers, including Microsoft Windows® Server 2003, Microsoft SQL Server™, and Microsoft BizTalk® Server, that integrates, runs, operates, and manages Web services and Web-based applications.
- Client software, such as Windows XP, Windows CE, and Microsoft Office XP, that helps developers deliver a deep and compelling user experience across a family of devices and existing products.

What is the .NET Framework?

The .NET Framework is a development and execution environment that allows different programming languages & libraries to work together seamlessly to create Windows-based applications that are easier to build, manage, deploy, and integrate with other networked systems.



The .NET Framework consists of:

- **The Common Language Runtime (CLR)**

The common language runtime (CLR) is responsible for run-time services such as language integration, security enforcement, and memory, process, and thread management. In addition, the CLR has a role at development time when features such as life-cycle management, strong type naming, cross-language exception handling, and dynamic binding reduce the amount of code that a developer must write to turn business logic into a reusable component.

- **The Framework Class Libraries (FCL)**

A consistent, object-oriented library of prepackaged functionality Base classes provide standard functionality such as input/output, string manipulation, security management, network communications, thread management, text management, and user interface design features.

The ADO.NET classes enable developers to interact with data accessed in the form of XML through the OLE DB, ODBC, Oracle, and SQL Server interfaces. XML classes enable XML manipulation, searching, and translations. The ASP.NET classes support the development of Web-based applications and Web services. The Windows Forms classes support the development of desktop-based smart client applications.

Together, the class libraries provide a common, consistent development interface across all languages supported by the .NET Framework.

Reasons for Using VB.NET

- **Seamless Deployment**. VB .NET promises to end "DLL Hell" and it may. I haven't experimented with it much yet but I've heard mixed reviews. Probably it will be better once we're used to it. It can't be a whole lot worse.

- **More Robust Code**. The features they list here are the real-time background compiler (to check syntax as you type) and the task list. Neither of these is a big step beyond VB 6. They also list strict type checking (which is good) and structured error handling. Structured error handling is really just more like the way it works in C/C++. It has some advantages but also some big disadvantages such as no Resume statements and no On Error Resume Next capabilities.

- **Powerful Windows-base Applications**. This isn't a real big deal. They point to the Form Designer (we've always had that), control anchoring and docking (replaces straightforward code with confusing design time properties), in-place menu editor (the old one was fine), and the tab order editor (a nice but tiny feature).

- **Powerful, Flexible Data Access**. They say VB .NET provides ADO and ADO.NET. Actually it looks like Microsoft is trying to dump ADO. ADO.NET does have some nice features, but it's aimed strongly at Web-like program structures where database users never interfere with each other. Record locking doesn't work as it does in ADO so building a traditional desktop application for multiple users requires some trickery.

- **Simplified Component Creation**. The Toolbox now includes more non-visual components (sort of like the Timer control) for things like database connection, Dataset, message queues, etc. It's not hard to create these objects in your code. Putting them on the form lets you do a bit more design-time configuration for them so this isn't a bad thing; it's just no big deal.

- **Enhanced Control Creation**. They don't say much here. It's true that VB .NET lets you build user controls. I haven't seen any big advantages over how it works in VB5/6 but I haven't spent a huge amount of time on this. One BIG disadvantage is the user control object (and the Picture Box and Form for that matter) no longer have an Auto Redraw property. If you want Auto Redraw, you need to implement it yourself with your own bitmap buffers.

- **Complete, Direct Access to the Platform**. This just means VB programmers can use the .NET Framework. That gives access to the registry, event log; performance counters, and files system. What this means in practice is you use new classes to do stuff that you would have done using API calls before. A nice feature but nothing you couldn't live without.

- **Integrated Reporting with Crystal Reports** (Professional edition and above). This lets you manipulate Crystal Reports within the IDE. I suppose you might care if you use Crystal Reports a lot.

Database Design

Appointment Scheduling System for Patients uses Microsoft SQL Server 2000 as its Backend. Microsoft Extends the Performance, reliability, quality and ease-of-use of Microsoft SQL Server version 7.0. Microsoft SQL Server 2000 includes several new features that make it an excellent database platform for large-scale online transactional processing (OLTP), data warehousing, and e-commerce Application.

Fundamentals of SQL Server 2000 Architecture

- Online Transaction Processing (OLTP) systems must be capable of handling thousands of orders placed at the same time.

- Increasing numbers of corporations are implementing large Web sites as a mechanism for their customers to enter orders, contact the service department, get information about products, and for many other tasks that previously required contact with employees. These sites require data storage that is secure, yet tightly integrated with the Web.

- Organizations are implementing off-the-shelf software packages for critical services such as human resources planning, manufacturing resources planning, and inventory control. These systems require databases capable of storing large amounts of data and supporting large numbers of users.

- Organizations have many users who must continue working when they do not have access to the network. Examples are mobile disconnected users, such as traveling sales representatives or regional inspectors. These users must synchronize the data on a notebook or laptop with the current data in the corporate system, disconnect from the network, record the results of their work while in the field, and then finally reconnect with the corporate network and merge the results of their fieldwork into the corporate data store.

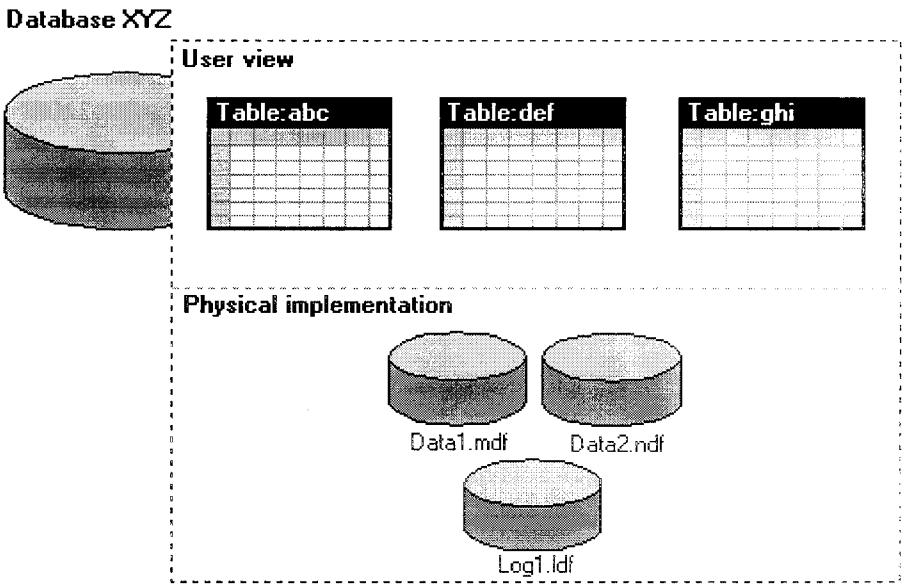
- Managers and marketing personnel need increasingly sophisticated analysis of trends recorded in corporate data. They need robust Online Analytical Processing (OLAP) systems easily built from OLTP data and support sophisticated data analysis.

- Independent Software Vendors (ISVs) must be able to distribute data storage capabilities with applications targeted at individuals or small workgroups. This means the data storage mechanism must be transparent to the users who purchase the application. This requires a data storage system that can be configured by the application and then tune itself automatically so that the users do not need to dedicate database administrators to constantly monitor and tune the application.

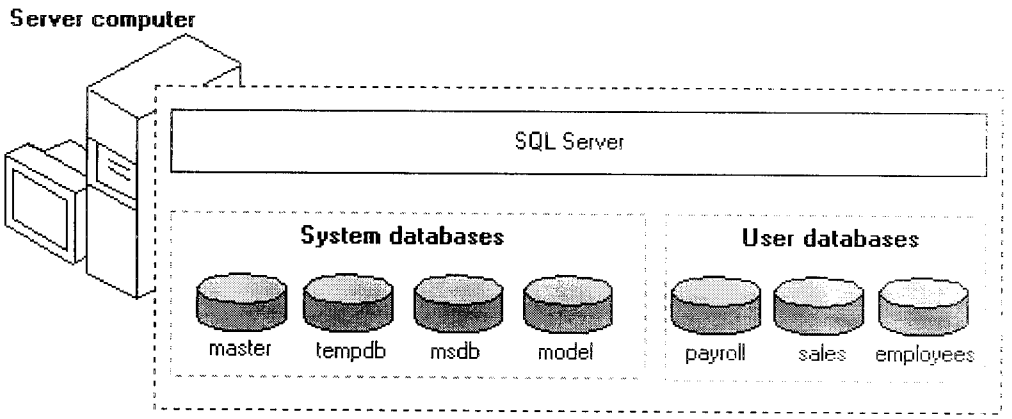
Database Architecture

Microsoft® SQL Server™ 2000 data is stored in databases. The data in a database is organized into the logical components visible to users. A database is also physically implemented as two or more files on disk.

When using a database, you work primarily with the logical components such as tables, views, procedures, and users. The physical implementation of files is largely transparent. Typically, only the database administrator needs to work with the physical Implementation.



Each instance of SQL Server has four system databases (**master**, **model**, **tempdb**, and **msdb**) and one or more user databases. Some organizations have only one user database, containing all the data for their organization. Some organizations have different databases for each group in their organization, and sometimes a database used by a single application. For example, an organization could have one database for sales, one for payroll, one for a document management application, and so on. Sometimes an application uses only one database; other applications may access several



It is not necessary to run multiple copies of the SQL Server database engine to allow multiple users to access the databases on a server. An instance of the SQL Server Standard or Enterprise Edition is capable of handling thousands of users working in multiple databases at the same time. Each instance of SQL Server makes all databases in the instance available to all users that connect to the instance, subject to the defined security permissions.

When connecting to an instance of SQL Server, your connection is associated with a particular database on the server. This database is called the current database. You are usually connected to a database defined as your default database by the system administrator, although you can use connection options in the database APIs to specify another database. You can switch from one database to another using either the Transact-SQL USE database name statement, or an API function that changes your current database context.

SQL Server 2000 allows you to detach databases from an instance of SQL Server, then reattach them to another instance, or even attach the database back to the same instance. If you have a SQL Server database file, you can tell SQL Server when you connect to attach that database file with a specific database name.

Features of SQL Server 2000

- **Internet Integration**

  The SQL Server 2000 database engine includes integrated XML support. It also has the scalability, availability, and security features required to operate as the data storage component of the largest Web sites. The SQL Server 2000 programming model is integrated with the Windows DNA architecture for developing Web applications, and SQL Server 2000 supports features such as English Query and the Microsoft Search Service to incorporate user-friendly queries and powerful search capabilities in Web applications.

- **Scalability and Availability**.

  The same database engine can be used across platforms ranging from laptop computers running Microsoft Windows® 98 through large, multiprocessor servers running Microsoft Windows 2000 Data Center Edition. SQL Server 2000 Enterprise Edition supports features such as federated servers, indexed views, and large memory support that allow it to scale to the performance levels required by the largest Web sites.

- **Enterprise-Level Database Features**.

  The SQL Server 2000 relational database engine supports the features required to support demanding data processing environments. The database engine protects data integrity while minimizing the overhead of managing thousands of users concurrently modifying the database. SQL Server 2000 distributed queries allow you to reference data from multiple sources as if it were a part of a SQL Server 2000 database, while at the same time, the distributed transaction support protects the integrity of any updates of the distributed data. Replication allows you to also maintain multiple copies of data, while ensuring that the separate copies remain synchronized. You can replicate a set of data to multiple, mobile, disconnected users, have them work autonomously, and then merge their

- **Ease of installation, deployment, and use**

  SQL Server 2000 includes a set of administrative and development tools that improve upon the process of installing, deploying, managing, and using SQL Server across several sites. SQL Server 2000 also supports a standards-based programming model integrated with the Windows DNA, making the use of SQL Server databases and data warehouses a seamless part of building powerful and scalable systems. These features allow you to rapidly deliver SQL Server applications that customers can implement with a minimum of installation and administrative overhead.

- **Data warehousing.**

  SQL Server 2000 includes tools for extracting and analyzing summary data for online analytical processing. SQL Server also includes tools for visually designing databases and analyzing data using English-based questions.

2.4 SPECIFICATION

Platform Specification

Operating System

An Operating System is a Collection of Computer program that control, how the computer works. It can also be defined as the software, which acts as a traffic cop, directing requests and information to add from the various devices within a single PC. The OS handles disk requests (read and write) keyboard translations, memory accesses, peripheral accesses and much other function.

Windows 9x

Windows 9x is a desktop operating system. It runs on Intel/Cyrix/AMD processors. It support file system FAT32 (NRWF), VFAT (NRWF). Also integrated with DOS for command line interface and explore for graphical user interface. It supports multimedia applications and Internet applications. Generally it is used as the operating system.

Windows NT

This is a network operating system. This is based on the Client-Server architecture. Benefits of Windows NT are;

- More intuitive interface.
- Better multitasking and multithreading.
- Clients can be attached to workstations.
- Plug-and-play technology.
- Higher level of security.
- NTFS-a powerful NT File System.

Windows 2000/NT Server

Microsoft Windows 2000/NT server 4.0 is a multipurpose server operating system. A multipurpose operating system integrates a variety of network services.

The services it provides are designed to address customer requirements and are managed in a single way. It offers a complete end-to-end solution. It gives the ability to exploit the popularity of compatible third party products and such application as Active Server pages, for dynamic content web pages or Server for specialized database tasks and web server for web services.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The existing system maintains the information about the patients, doctors, appointments and equipments, rooms and the invoice details manually. This may lead to manual errors; high time consumption, low performance and the data cannot be quickly retrieved.

Disadvantages

- The categories are not separated.
- All users have same access privileges.
- More manual works is done.
- Modification and updating is difficult and complex.
- Data manipulation not secure and efficient.
- A doctor feels uneasy to maintain the Appointments scheduling system.
- Low performance.

3.2 PROPOSED SYSTEM

The proposed system automates the information maintenance, which will be of great advantages in terms of efficiency, accuracy and consistency. This also avoids repetition of the same data for different purposes.

This project aims at developing the information maintenance tool specially suited for hospitals that handle Patient, Doctor, Appointments, Equipments, rooms and invoice details.

The Proposed System is developed using VB.NET and MICROSOFT SQL SERVER 2000 as database.

Advantages of Proposed System

- The System is Categorized as follows
  - Administration
  - Doctors and Patient information
  - Appointments System
  - Data Manager
  - Medical Equipments and room information
  - Invoice details
- Almost all the processes are automated to implement the concept of paperless office.
- Maintain the archived appointments of whole scheduling system.
- Status about the appointment and Doctors unavailability are maintained.
- Doctors are categorized to schedule the appointments for patients effectively.
- Checking the availability of time and scheduling is done in a fraction of second.

3.3 PROTOTYPING

Planning the software development process involves several important considerations. The most important consideration is to define a product life cycle.

The software life cycle encompasses all activities required to define, develop, test, deliver, operate, and maintain a software product. Different

models emphasize different aspect of the life cycle, and no single life-cycle model is appropriate for all software products. Life-cycle models used are the phased model, the cost model, the prototype model and the successive versions model.

The Prototype Life-cycle Model

Prototype is a mock-up or model of a software product. A prototype incorporates components of the actual product. Typically, a prototype exhibits limited function capabilities, low reliability, and inefficient performance.

There are several reasons for developing a prototype:

- To illustrate input date formats, message, reports, and interactive.
- Dialogues for the customer. This is used to explain various processing option to the customer and to gain better understanding of the customer's needs.
- To explore technical issues in the proposed product.

The nature and extent of prototype to be performed on particular software is dependent on the nature of the product. New versions of existing products can most likely be developed using the phased life-cycle model with little or no prototyping. Development of a totally new product will probably involve some prototyping during the planning and analysis phase or iterating through a series of successive designs and implementations may develop the product.

CHAPTER 4

SYSTEM DESIGN

4.1 DESIGN PRINCIPLES

The process of design involves "conceiving and planning out in mind" and "making a drawing, pattern, or sketch of". In software design, there are three distinct types of activities: external design, architectural design detailed design. Architectural design and detailed design are collectively called internal design.

External design of software involves conceiving, planning out, and specifying the externally observable characteristics of a software product. These characteristics include user displays and report formats, external data sources and data sinks, and the functional characteristics, performance requirements and high-level process structure for the product. External design begins during the analysis phase and continues into the design phase. Requirements definition is concerned with specifying the external, functional and performance requirements for a system. External design is concerned with refining those requirements and establishing the high level structural view of the system.

Internal design involves conceiving, planning out and specifying the internal structure and processing details of the software product. The goals of internal design are to specify internal structure and processing details, to record design decisions and indicate why certain alternatives and trade-offs were

chosen, to elaborate the test plan, and to provide a blueprint for implementation. testing and maintenance activities. The work products of internal design include a specification of architectural structure, the details of algorithms and data structures, and data structures, and the test plan.

Design of Appointment Scheduling System for Patients

Appointment scheduling system for Patients is designed in the way that maintains the information of Appointments and makes better use of that information. This system is an integral part of daily work for healthcare professionals, from family practices to large clinics, from physician offices to hospitals. Medical office staff has to schedule patients for dentist, optometrist, ophthalmologist, surgeon, psychiatrist and other general care and specialist appointments.

Appointments need to be coordinated, medical support staffs have to be constantly aware of all new patients and doctor's schedule. This system is designed for information specially suited for hospitals that handle Patient, Doctor, Appointment, Equipment, room and invoice details.

Requirements Definition

What is Appointments scheduling System for Patients?

Appointment scheduling system for Patients provides high quality, multidisciplinary health and hospice care to individuals of all patients, to maintain doctors' appointments and improves patients and doctors satisfaction.

Appointments Scheduling System for Patients may be considered for patients who are:

- Reducing appointments cancellations
- Undergoing treatment.
- Waiting for appointments

✓ **Doctors' care**

Following consultation with Doctors perform specific tasks including:

- Providing appointments efficiently.
- Completing physical assessments.
- Instructing families on the care of acutely ill patients.
- Supervising treatment and diet.
- Instructing in and supervising the use of medication.
- Administering intravenous medications.
- Counseling on health issues.
- Assisting patients and their families with crisis intervention.

✓ **Providing medical equipments**

✓ **Special care and room facilities**

4.2 ARCHITECTURAL DESIGN

Architectural Design is concerned with refining the conceptual design of the system, identifying internal processing functions, decomposing high level functions into sub functions, defining internal data streams and data stores and establishing relationships and interconnections among functions, data streams and data stores.

4.2.1 Appointment scheduling System for Patients – Module overview

ASSP consists of six modules

- Administrator Module
- Patient Module
- Doctor Module
- Equipment and Room Module
- Appointment Module
- Data Manager Module

Administration module

Administration module is the security level concerns to allow the users with their mode. It deals with all the modes like administrator mode, scheduler mode and user mode.

Patient Module

Patient module allows maintain the details of patients and gets the type of service the patient needs and sends it to the particular module or department.

- Doctors Care
- Equipments and Rooms needs
- Treatments

Doctors Module

The doctor deals with details about the doctors in the hospital and helps us to know whether they are specialist or general doctors and if they are

specialists means, in which field such surgery, cardio, neuro etc. It also helps us to know the availability of doctors and their visiting time also.

Equipments and Rooms module

Equipments and Rooms module maintains the details of the equipments and rooms in home health center and scheduling for patients needs.

Data Manager Module

In this module scheduler or administrator maintains and displays information of the

- Appointment details
- Details of archived appointments
- Doctors and Patients details
- Categorization of doctors
- Equipments and rooms details
- Invoices details
- Doctors unavailability durations
- Status of the appointment
- Deleted appointments and
- Security of the users

Also maintains the waiting list of the appointment for particular doctors.

4.3 DATA FLOW DIAGRAM

Data flow diagram is commonly used during problem analysis and design. A DFD shows the flow of data through a system. It views the system as a function that transforms the inputs into desired outputs. A DFD aims to capture the transformation that takes place within a system into output data so that eventually the output data is produced. The agent that performs the transformation of data from one state to another is called a process (Bubble). Named circles show the processes and dataflow are represented by named arrows. A square defines a source or destination of system data. An open rectangle is a data source.

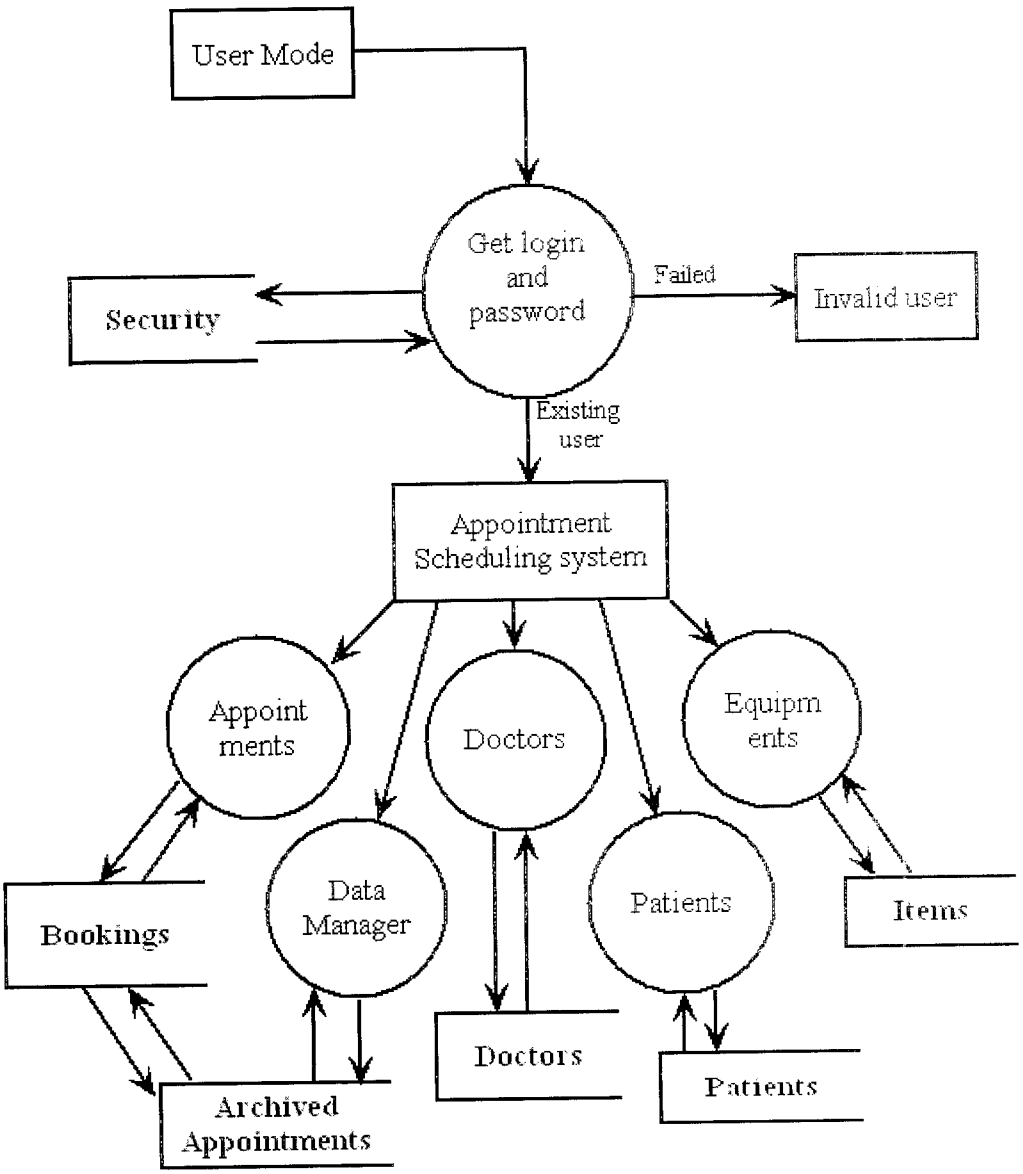


Figure 4.3.1 SYSTEM LEVEL FLOW DETAILS

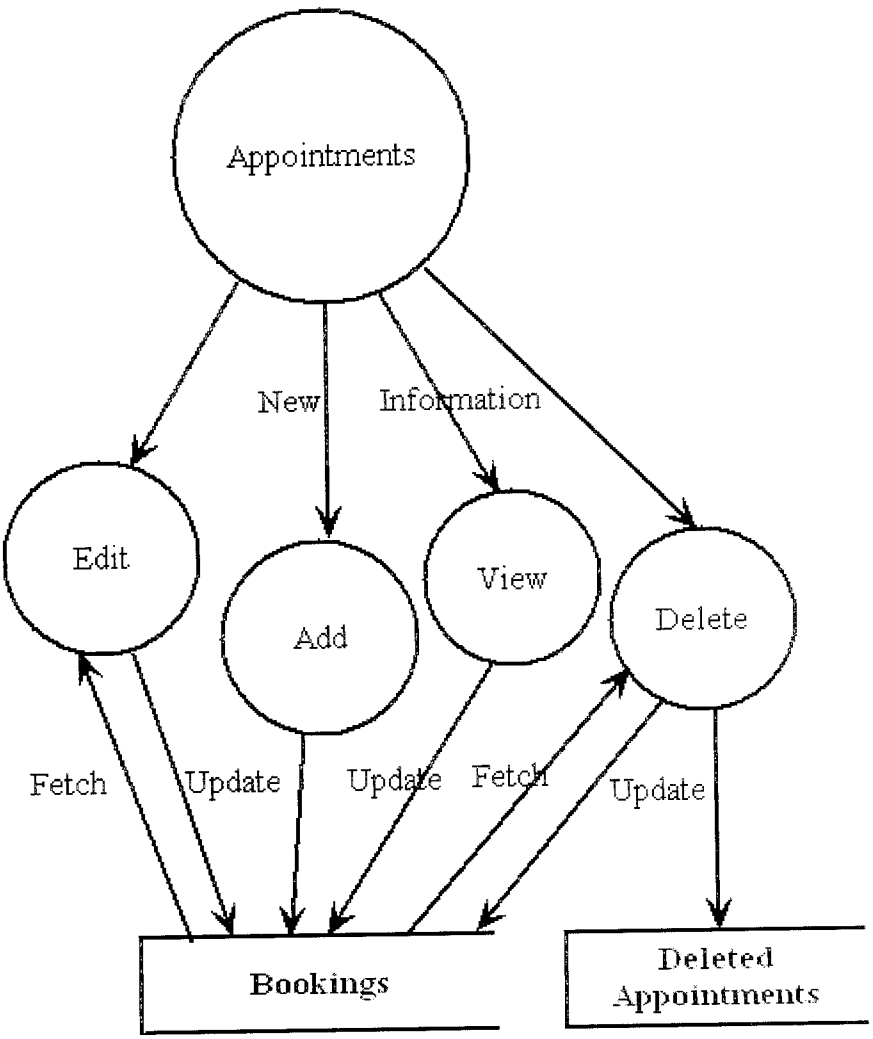


Figure 4.3.2 APPOINTMENT DETAILS

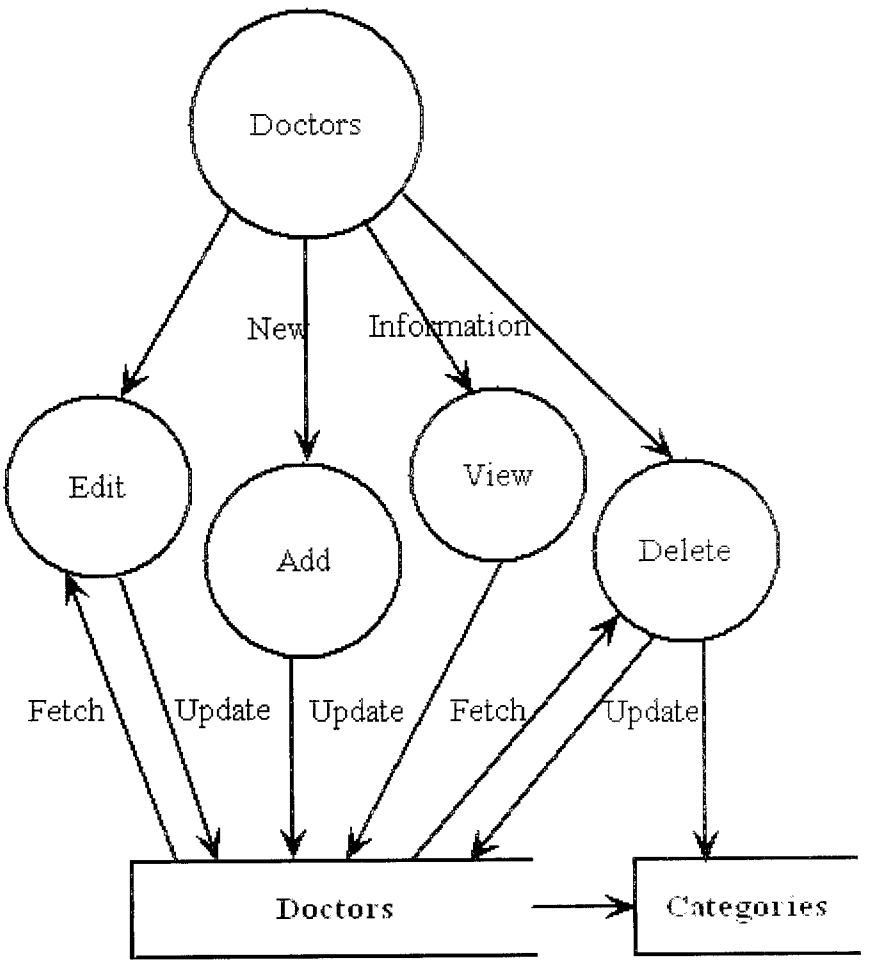


Figure 4.3.3 DOCTORS DETAILS

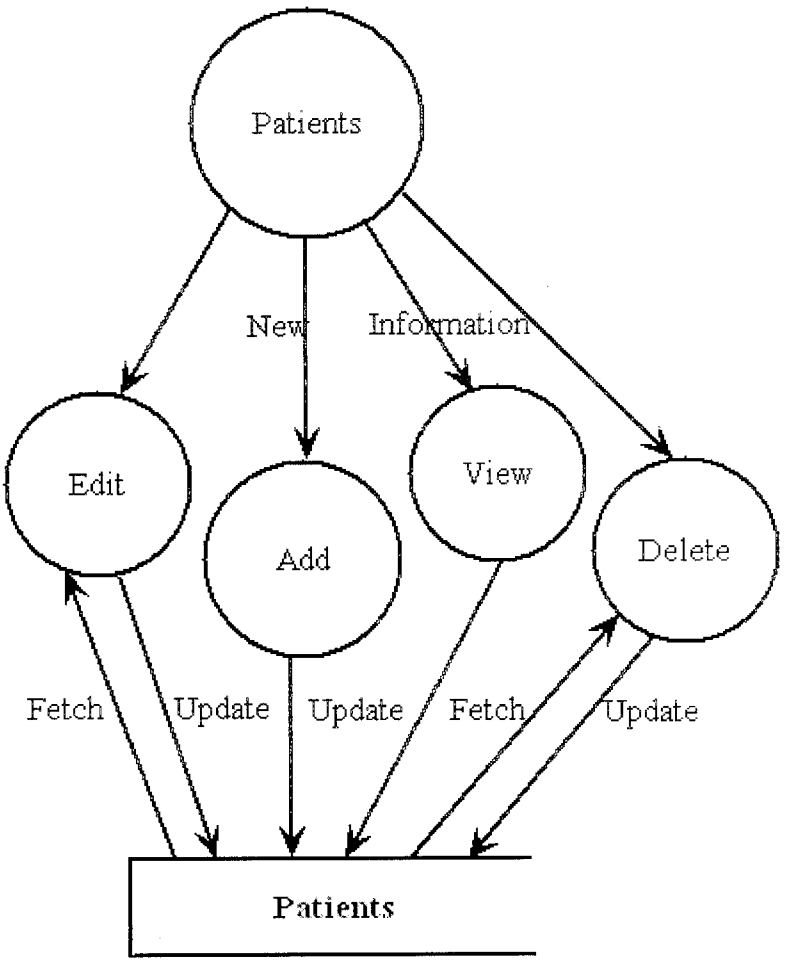


Figure 4.3.4 PATIENT DETAILS

4.4 TABLE DESIGN

4.4.1 Table Name: Archived Appointments

COLUMN NAME	DATA TYPE	LENGTH	DESCRIPTION
ID	Number	9	Identification code
Date	Date/time	8	Appointment date
StartTime	Date/time	8	Starting time
EndTime	Date/time	8	Ending time
Person	Varchar	30	Doctor Name
Room	Varchar	15	Room detail
Description	Varchar	250	Description
Freq	Varchar	10	Frequency code
LoginID	Varchar	30	User Name
Notes	Varchar	250	Notes
EndDate	Date/time	8	Ending date
ClientID	Varchar	25	Patient Name
Status	Char	1	Status
Modified	Varchar	25	Modification check
Invoice	Varchar	10	Invoice detail
Color	Char	1	Color field
Items	Varchar	250	Equipments detail

4.4.2 Table Name: Bookings

COLUMN NAME	DATA TYPE	LENGTH	DESCRIPTION
ID	Number	9	Identification code
Date	Date/time	8	Date
StartTime	Date/time	8	Starting date
EndTime	Date/time	8	Ending Date
Person	Varchar	30	Doctor Name
Room	Varchar	15	Room detail
Description	Varchar	250	Description
Freq	Varchar	10	Frequency code
LoginID	Varchar	30	User Name
Notes	Varchar	250	Notes
EndDate	Date/time	8	Ending date
ClientID	Varchar	25	Person Name
Status	Char	1	Status
Modified	Varchar	25	Modified details
Invoice	Varchar	10	Invoice details
Color	Char	1	Color field
Items	Varchar	250	Equipments detail

4.4.3 Table Name: Patients

COLUMN NAME	DATA TYPE	LENGTH	DESCRIPTION
ClientID	Varchar	25	Patient code
FirstName	Varchar	20	First Name
LastName	Varchar	20	Last Name
HomePhone	Varchar	20	Home phone
WorkPhone	Varchar	20	Working phone
Cell	Varchar	20	Mobile number
Address1	Varchar	30	Address
Address2	Varchar	30	Address
City	Varchar	20	City Name
Postal	Varchar	15	Postal code
Notes	Varchar	250	Notes
E-Mail	Varchar	40	E-mail
State	Varchar	10	State Name
Rate	Varchar	8	Rate details
Deposit	Money	8	Deposit Amount
Person	Varchar	25	Doctor Name
BirthDate	Date/Time	8	Birthday
Age	Number	3	Age
Gender	Varchar	10	Gender
Regular	Varchar	20	Visiting type
Referral	Varchar	25	Referrals

4.4.4 Table Name: Doctors

COLUMN NAME	DATA TYPE	LENGTH	DESCRIPTION
Person	Varchar	30	Doctor Name
Title	Varchar	50	Category
Phone	Varchar	20	Phone number
Cell	Varchar	20	Mobile number
Pager	Varchar	20	Pager
Address1	Varchar	25	Address
Address2	Varchar	25	Address
Notes	Varchar	250	Notes
EMail	Varchar	40	E-mail
Category	Varchar	15	Category Name
Fax	Varchar	20	Fax number
Color	Char	1	Color field
Rate	Money	8	Total cost

4.4.5 Table Name: Rooms

COLUMN NAME	DATA TYPE	LENGTH	DESCRIPTION
Room	Varchar	15	Room code
Description	Varchar	50	Description
Capacity	Number	9	Capacity
Equipment	Varchar	250	Equipment detail
Cost	Money	8	Charging amount

4.4.6 Table Name: Invoices

COLUMN NAME	DATA TYPE	LENGTH	DESCRIPTION
Invoice	Varchar	10	Invoice code
ClientID	Varchar	25	Patient code
Paid	Varchar	10	Paid detail
InvoiceDate	Date/Time	8	Invoice date
PayDate	Date/Time	8	Pay date

4.4.7 Table Name: Appointment status

COLUMN NAME	DATA TYPE	LENGTH	DESCRIPTION
Status	Char	1	Status code
Description	Varchar	15	Description
Color	Char	1	Color field
Rate	Money	8	Total cost

4.4.8 Table Name: Categories

COLUMN NAME	DATA TYPE	LENGTH	DESCRIPTION
Category	Varchar	15	Category Name
Description	Varchar	20	Description

CHAPTER 5

SYSTEM TESTING AND IMPLEMENTATION

The system testing deals with the process of testing the system as a whole. This is done after the integration process. Moving through each module from top to bottom tests the entire system. The verification and validation processes are then carried out. The errors that occur at testing phase are eliminated and a well functioning system is developed.

Test case design focuses on a set of techniques, which meets all testing objectives, which are mentioned below.

1. Testing is a process of executing a program with the intent of finding an error.
2. A successful test is one that uncovers an as yet undiscovered error.

Testing demonstrates that software functions work according to specifications. In addition data collected from testing provides a good indication of software reliability and some indication of software quality as a whole.

Testing results are used for detecting errors. Critical modules are tested as early as possible. The following tests are carried out.

5.1 TESTING METHODS

5.1.1 Unit Testing

It focuses verification efforts on the smallest unit of software design, the module. This is also known as **Module Testing.** The modules are tested separately. This testing is carried out during programming stage itself.

5.1.2 Validation Testing

Validation testing can be defined in many ways but a simple definition is that validation succeeds when the software functions in a manner that can be reasonably expected by the users.

After validation test has been conducted one of the two possible conditions exist

1. The function or the performance characteristics confirm to specification and are accepted
2. A derivation from specification is uncovered and a deficiency list is created.

5.1.3 Output Testing

After performing the validation testing the next step is output testing of the proposed system since no system is useful if it does not produce the required output in the specific format. Asking the users about the formats required by them tests the outputs generated or displayed by the system under consideration.

5.1.4 User Acceptance Testing

User acceptance of a system is a key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system users at the time development and making changes whenever required.

5.2 IMPLEMENTATION

The implementation phase of software development is concerned with translating design specification into source code. The primary goal of implementation is to raise source code and internal documentation so that conformance of the code to its specification can be easily verified, and so that debugging, testing and modification are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity, clarity and elegance are the hallmarks of good programs; obscurity, cleverness and complexity are indication inadequate design and misdirected thinking.

Source code clarity is enhanced by structured coding techniques, by good coding style, by appropriate supporting documents, by good internal comments and by the features provided in modern programming languages.

The goal structured coding is to liberalize control flow through a computed program so that the execution sequence follows the sequence in which the code is written. The dynamic structure of a program as it executes then resembles the static structure of the written text. This enhances readability of code, which eases understanding, debugging, testing, documentation and modification of programs. It also facilitates formal verification of programs. The structure coding techniques are as follows:

- Single entry, Single exit constructs
- Efficiency considerations
- Data Encapsulation
- Recursion

5.3 MAINTENANCE

Maintenance is the enigma of system development. It holds the software industry captive typing up programming resources. It could be described as the symmetric process of changing the software that is already in operation iq. order to prevent system failures and to improve the performance. Software maintenance involves keeping software interfaces simple and standard, paying particular attention to troublesome modules, replacing faulty components and generally planning to replace components that are ole, obsolete, faulty, or at risk for imminent failure.

There are several factors that require to be maintained. They are

- Hardware platforms change or become obsolete.
- Operating system change.
- Compiler change
- Language standard's change.
- Communication standard's change
- Graphical user interface change.
- Related application software package change.

Maintenance can be classified into

- Adaptive maintenance
- Perceptive maintenance
- Preventive maintenance
- Corrective maintenance

Adaptive Maintenance

It deals with adapting software change in the environment. It does not lead to changes in the system functionality.

Perceptive Maintenance

It mainly deals with accommodating new or changed users requirements. It also includes activities to increase the system performance or to enhance its user interface. The objective of perceptive maintenance should be to prevent failures and optimize the software.

Preventive Maintenance

Preventive maintenance concerns activities aimed at increasing the system's maintainability such as updating documentation adding comments, improving modular structure of the system.

Corrective Maintenance

This deals with the repair of faults found. Some of the major causes of maintenance problems are:

- Unstructured code
- Maintenance programmers having insufficient knowledge of the system and on application domain.

Chapter 6

CONCLUSION

Appointment Scheduling System for Patients is successful Software with all the requirements being satisfied. This software has successfully archived the functionality that was expected.

It contain three major packages

- Appointments
- Data Manager
- Doctor details
- Patient details

It provides a user friendly environment and reduces the Appointment Scheduling System medical staffs difficulties

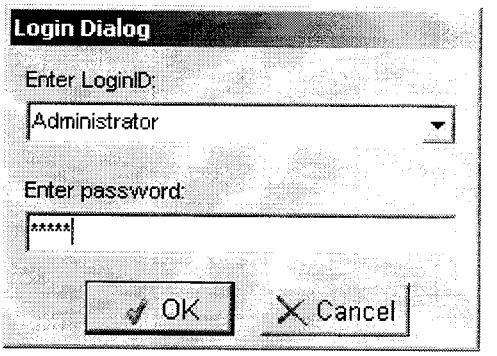
Further Enhancements

- Provide facilities for accepting the cash in credit cards.
- Provide facilities for online scheduling system.
- Can be Extend the over all the branches.

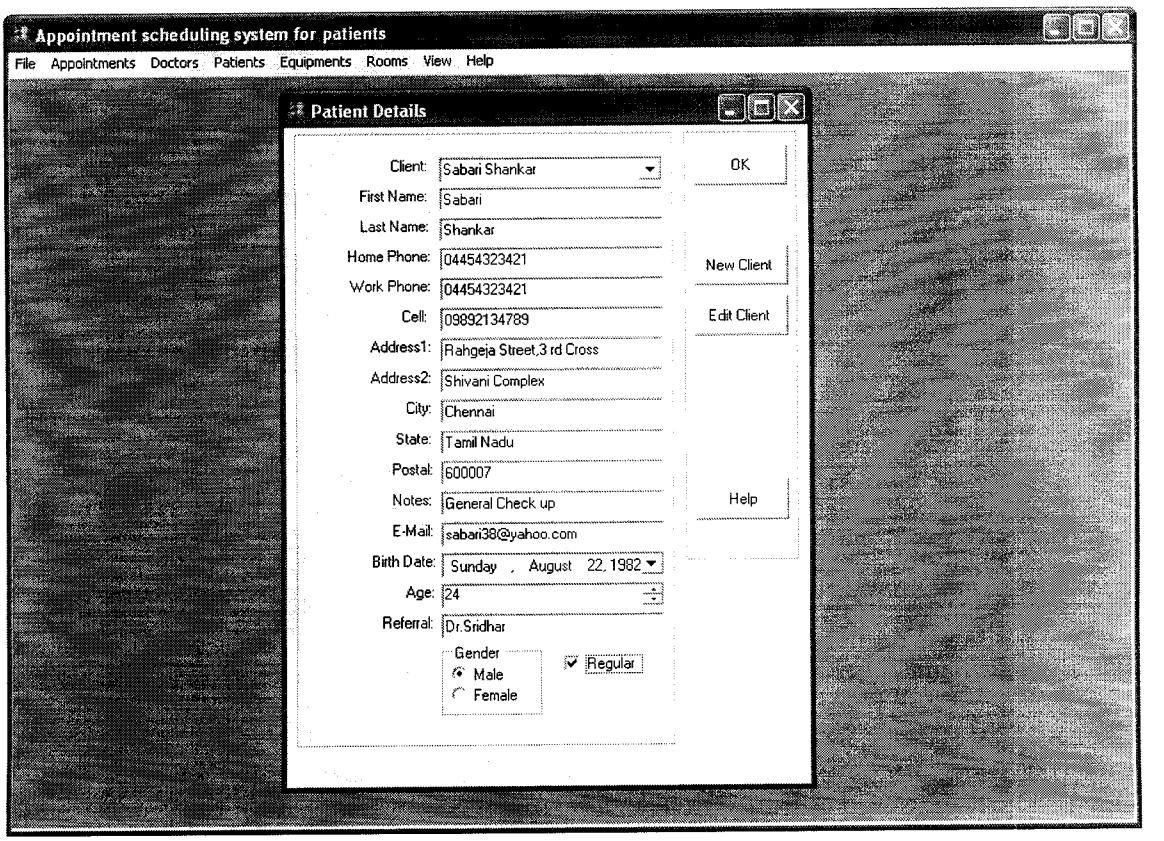
APPENDIX

SCREEN SHOTS

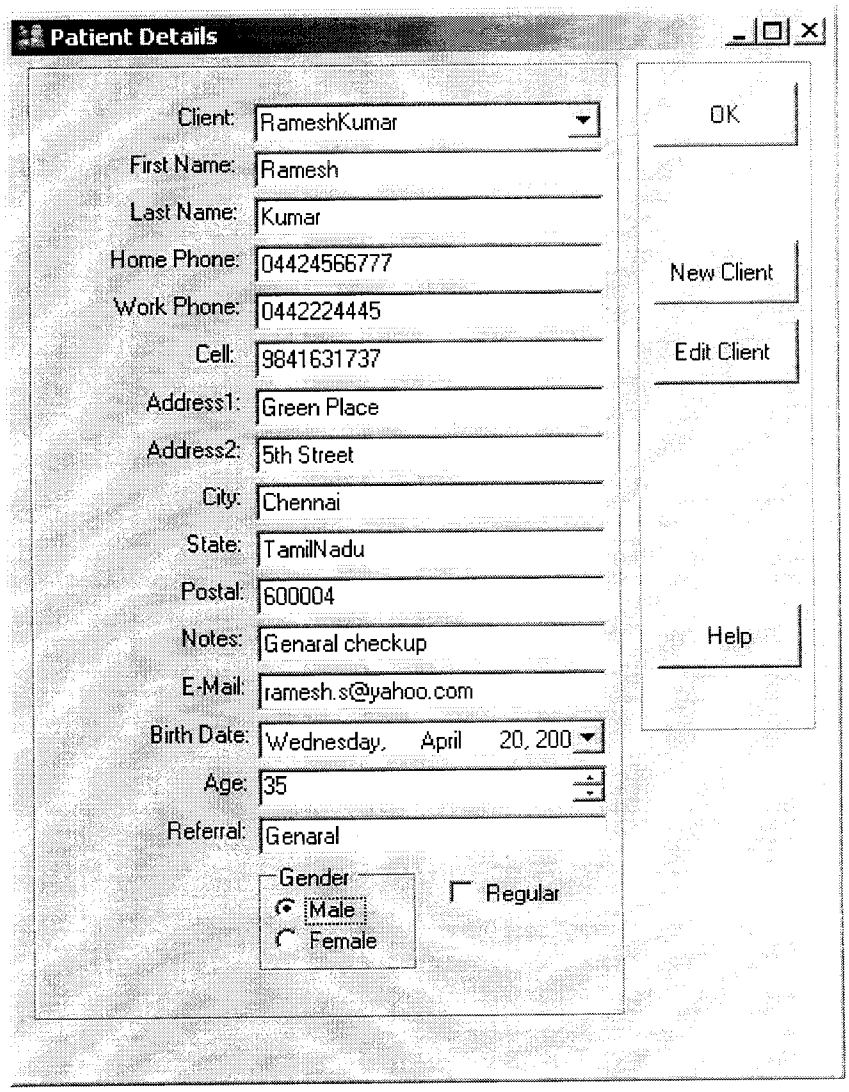
Login interface form



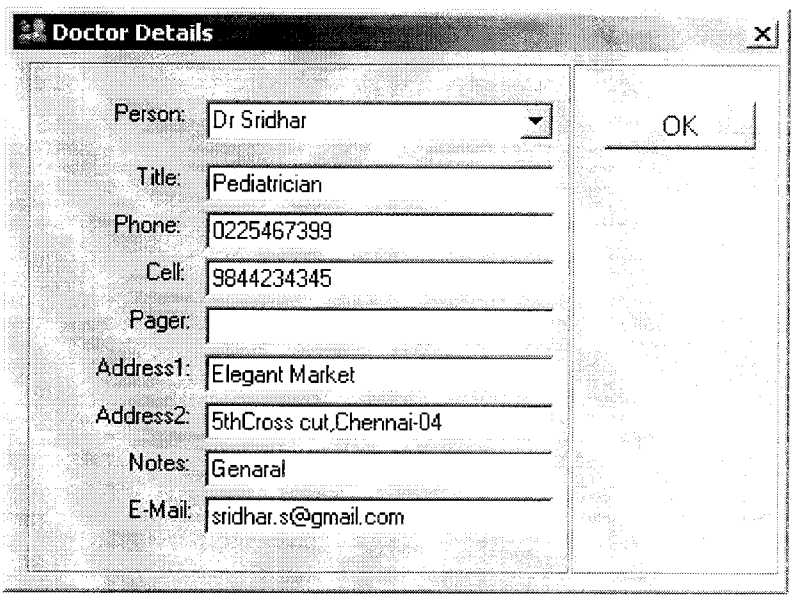
Main interface form



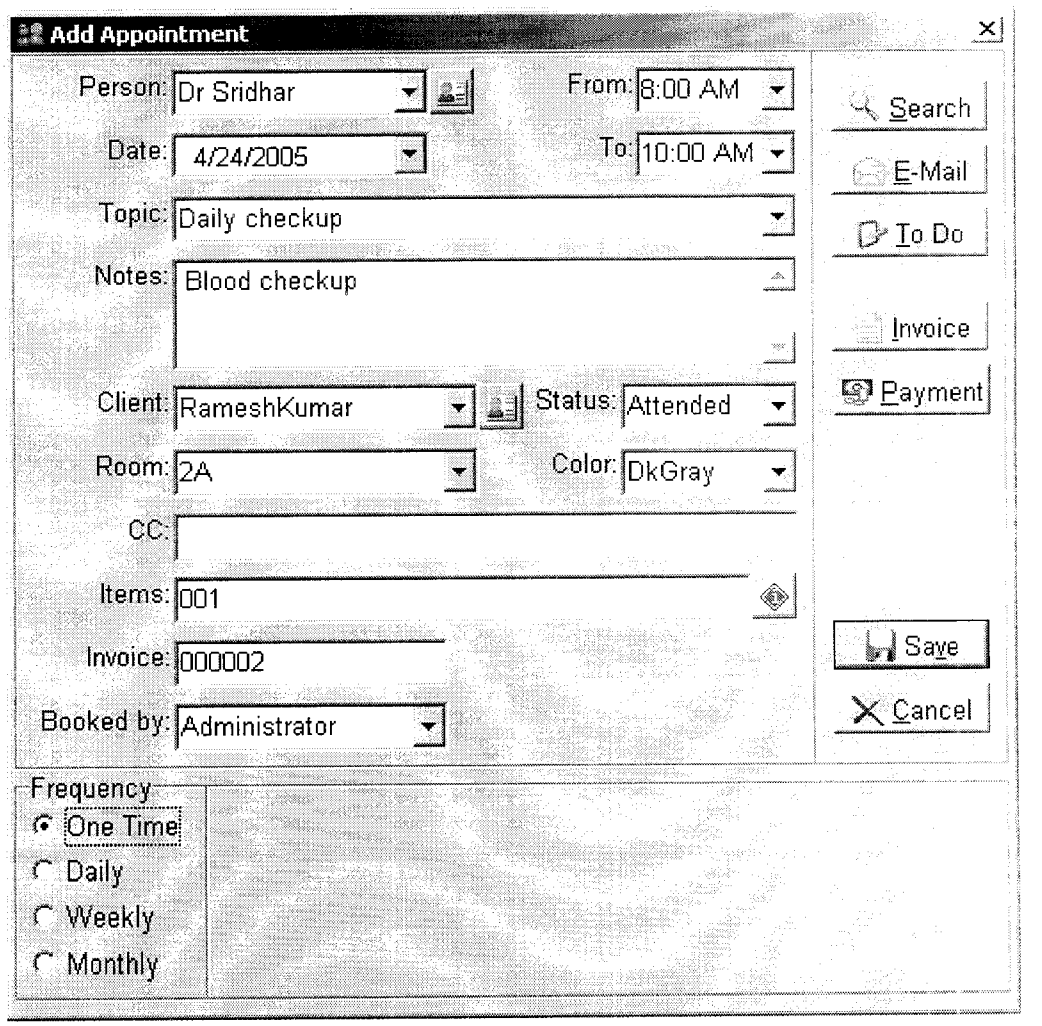
Patient details



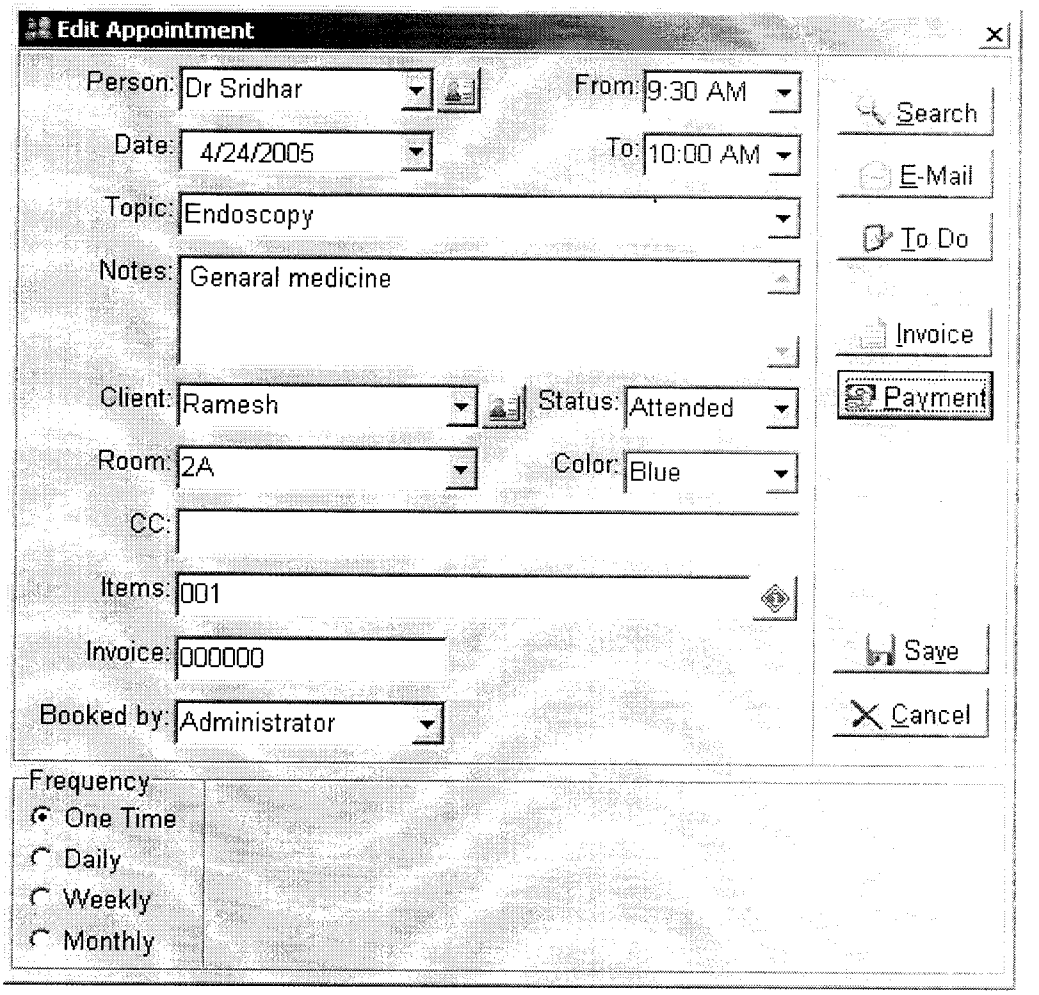
Doctor details



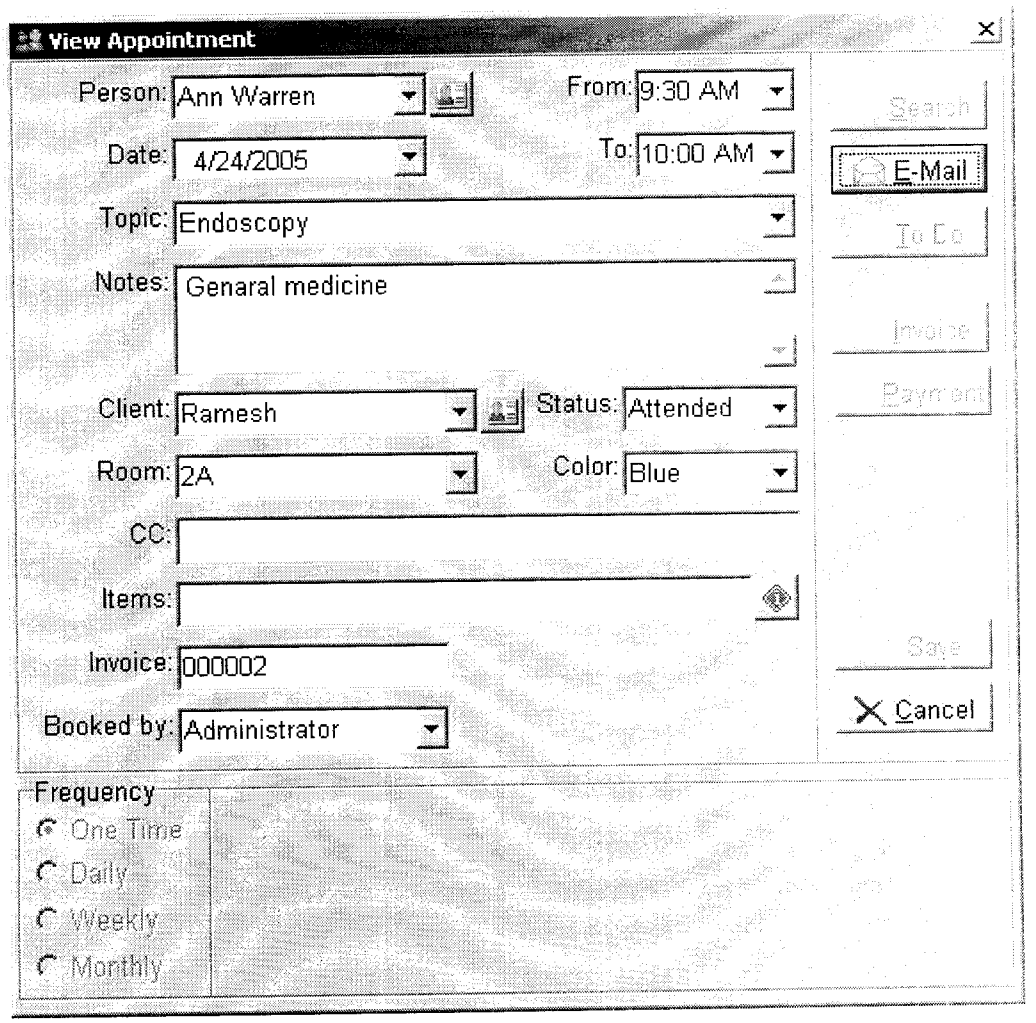
Add Appointment Details



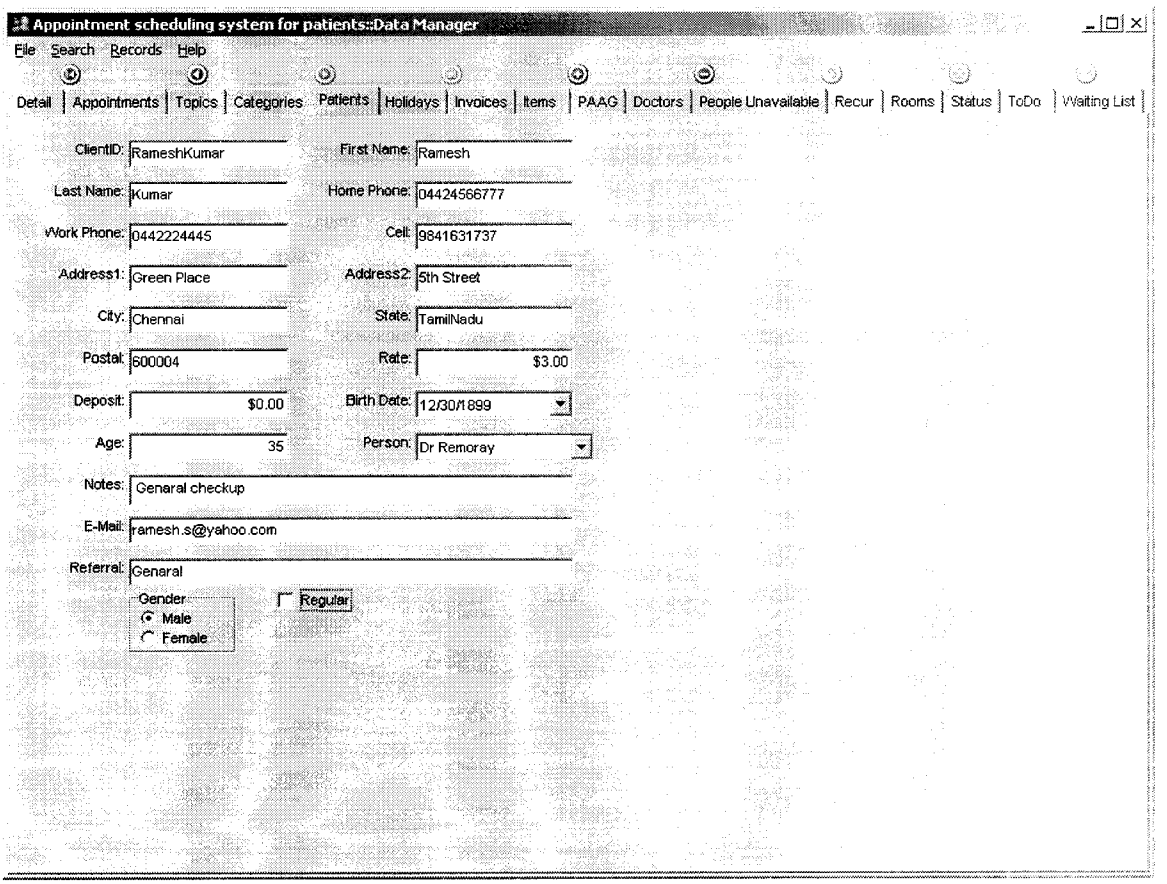
Edit Appointment Details



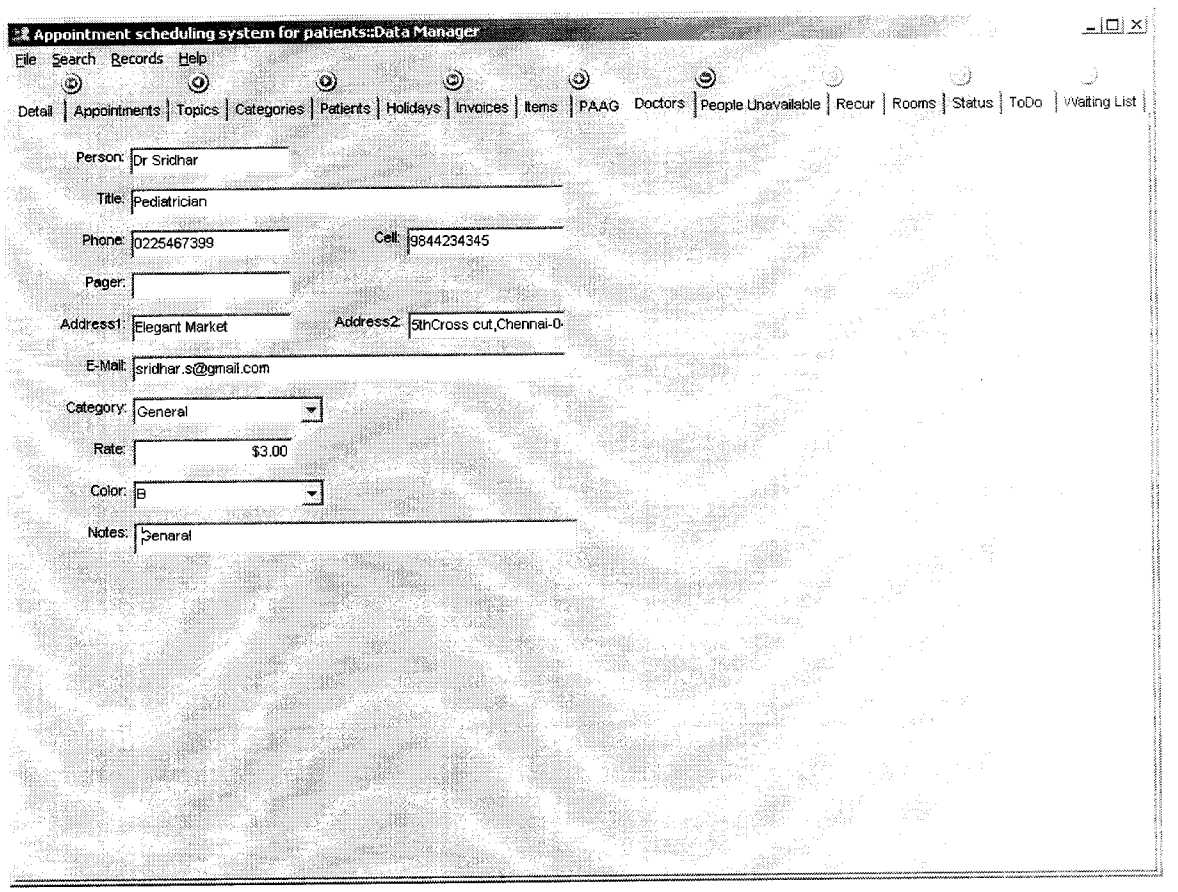
View Appointment Details



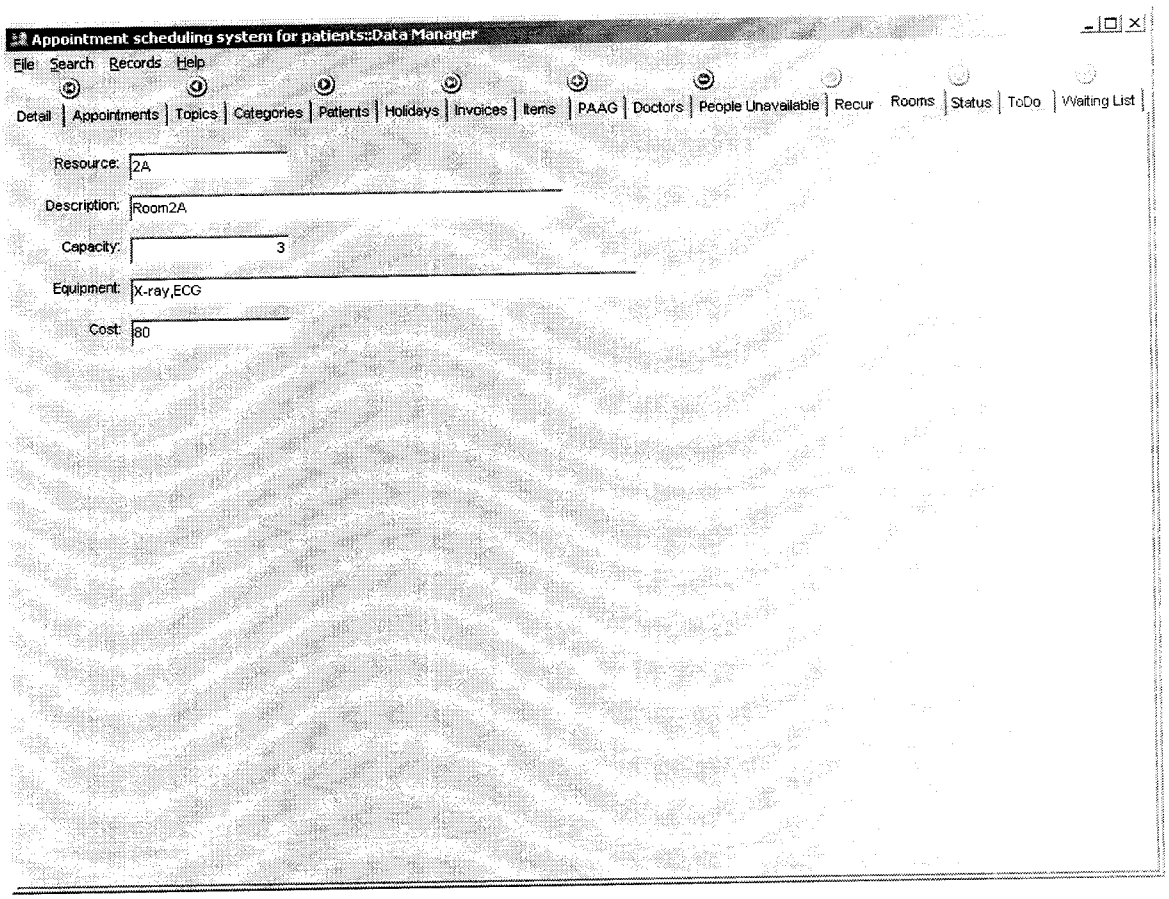
Data Manager – Patients information



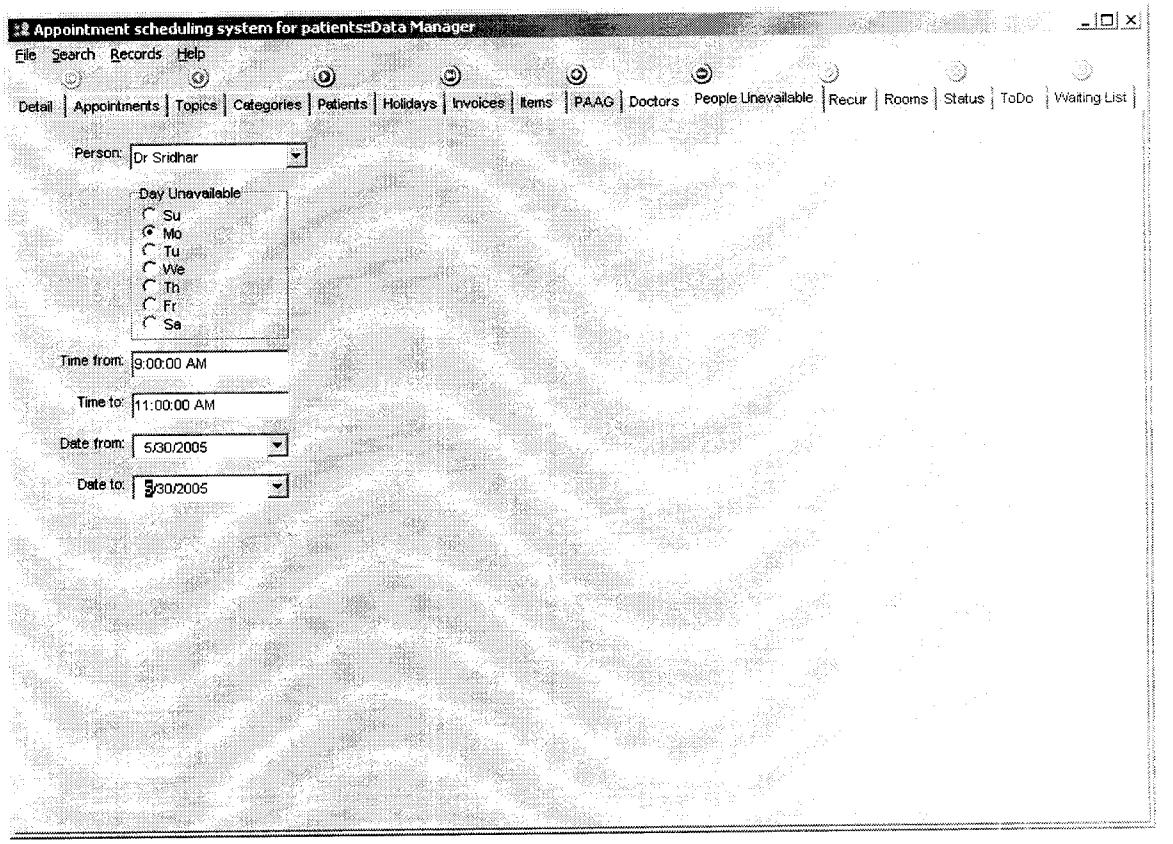
Data Manager – Doctors information



Data Manager - Rooms information



Data Manager – Doctor Unavailable information



REFERENCES

Books

[1] Ron Sonkup and Kalen Delaney "Inside Microsoft SQL Server 2000", 2000 Microsoft Press

[2] Damien Watkins "Programming in the .NET Environment ". 2003 BPB Publications

Online References

- www.DotNetSpider.com.
- www.microsoft.com.
- www.DotNetHeaven.com.