Automation of Pharmacy System

By

Raja.P

Reg. No. 71204621025

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

June 2007

Kumaraguru College of Technology

Coimbatore – 641006.

Department of Computer Applications

Bonafide Certificate

Certified that this project report titled **Automation of Pharmacy System** is the bonafide work of **Mr. Raja.P** 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.

Project Guide

Head of Department

Submitted for the University Examination held on 02-07-2007

Internal Examiner

External Examiner

CARITO

Vision ➢

June 7th, 2007

To whomsoever it May Concern

This is to inform you that **P.RAJA** has successfully completed his project assignment titled **AUTOMATION OF PHARMACY SYSTEM** as a part of **MCA** curriculum.

As a Project Trainee, he started this project on **December 18, 2006** and completed it on **June 7, 2007**

Please note, as per the company's policies and practices, the company retains ownership of the intellectual property rights concerning work undertaken during projects and disclosure of the source code and any other relevant information or data out of the organization is strictly prohibited.

P.RAJA designated, as project trainee will not be delivering the respective source code pertaining to his project.

For Caritor (India) Pvt Ltd,

Mary Mitra Mazumdar

MARY MITRA MAZUMDAR
ASSISTANT MANAGER – HUMAN RESOURCES

Abstract

The project entitled **Automation of Pharmacy System** deals with Integrating software product such as Pharmacy2000 (Pharmacy workflow system), AutoLink v 3.0 and Pharmaserv (Pharmacy Management Suit) and adds new functionality based on the customer requirement. The goal of this project is to provide an integrated workflow solution that will create a more efficient pharmacy operation by reducing medication errors and increasing both patient and user satisfaction. Additionally, the integrated product will leverage existing suite of products and add value that customers can identify with and that will intuitively drive the market to purchase the new suite of products.

Global Interface Module has come into existence because of the need to ensure seamless transfer of information from the Pharmacy Data Entry Application (Host) to RA database that supports several communication methods and parsing techniques. It can be triggered only when the RxProcessing license is off and Global Interface license is on.

The information is sent by the Host to Global Interface (GI). GI will receive the transmission and process the incoming message strings based on the parser selected in the configuration screen. GI will parse the records accordingly and will update the RA database. After updating the database GI acknowledges the Host. GI will also retrieve information from the RA database to generate the outbound message strings. Workflow is the total system of dispatching of the pharmacy order that includes imaging, filling, checking etc.

This project uses VB.NET 2003 as front end and MS SQL SERVER 2000 as the database Backend.

Acknowledgement

I express my grateful thanks to our beloved principal, **Dr. Joseph V Thanikal, Kumaraguru College of Technology**, Coimbatore, for giving me an opportunity to take up this project.

I express my deep sense of gratitude to **Dr.S.Thangasamy,** Professor and Dean of Computer Science and Engineering department and **Dr.M.Gururajan,** Professor and Head of Department of Computer Applications for extending their help in providing all the facilities at college for the successful completion of the project.

I am greatly indebted to **Mr. A.MuthuKumar,** Project Coordinator, Assistant professor, Department of Computer Applications, for his valuable guidance and encouragement at every stage of this project work.

I would like to express my deepest and sincere gratitude to **Mr M.Manikandan,** Senior Lecturer Department of Computer Applications for his guidance, support, cooperation and valuable suggestions during the course of this project.

I would like to thank **Mr. Deepak Pandey**, Accounts Manager, **Caritor (India) Pvt.** Ltd., Bangalore for the valuable guidance, encouragement and creative suggestions offered during this project and also in preparing this report.

I would also wish to record my gratitude to all my friends for their immense help and support given during development of this project.

Table of Contents

List of Tables

Table Description **Page No**

List of Figures

CHAPTER 1

INTRODUCTION

1.1 SYSTEM OVERVIEW

The project entitled **Automation of Pharmacy System** deals with Integrating software product such as Pharmacy2000 (Pharmacy workflow system), AutoLink v 3.0 and Pharmaserv (Pharmacy Management Suit) and adds new functionality based on the customer requirement. The goal of this project is to provide an integrated workflow solution that will create a more efficient pharmacy operation by reducing medication errors and increasing both patient and user satisfaction. Additionally, the integrated product will leverage existing suite of products and add value that customers can identify with and that will intuitively drive the market to purchase the new suite of products.

Global Interface is an application that acts as a seamless communication layer for transferring information from the host pharmacy management system to the workflow. This interface supports several communication methods (TCP/IP, Serial and Direct Poll) and parsers (Retail, HL7 and ACS).

The information is sent by the Host to Global Interface, GI will receive the transmission and process the incoming message strings based on the parser selected in the configuration screen. GI will parse the records accordingly and will update the RA database. After updating the database GI acknowledges the Host. GI will also retrieve information from the RA database to generate the outbound message strings. Workflow is the total system of dispatching of the pharmacy order that includes imaging, filling, checking etc.

PHARMASERV and RA are both client products. Information collected by the PHARMASERV application is updated into the RA database directly without the help of any third party utility. Workflow is able to access this information from the database.

The scenario is completely different if any NON MPS host wants to send any information to the workflow. This NON MPS host will not be able to pass the information directly. The foreign host will send message strings that cannot be interpreted by the RA. This will limit the scope of MPS.

In order to alleviate this issue, the GLOBAL INTERFACE has been introduced by MPS. The Global Interface will act as an intermediate communication channel. It will accept the message strings sent by the NON MPS host, parse them according to the mutually decided parsing standard and update the information into the RA database. Depending on the result of the updating and parsing, the Global Interface will send appropriate messages back to the respective host. Thus, the information is now accessible to the workflow

1.2. OBJECTIVES OF THE PROJECT

- Meet or exceed customer's expectations in terms of their workflow needs with workflow encompassing all operational flows occurring within a retail / outpatient pharmacy.
- Provide the majority of key functionality currently available in Pharmacy2000, and Auto Link
- Scale well across customers of different sizes and market segments.
- Include some pathway for current customers to easily adopt the new solution.
- Work independent of products, or alternate vendor product.
- Interoperate with all other existing products.
- Support existing customers of Pharmaserv, P2000 and Autolink.
- Offer an intuitive experience consistent with products.

1.3 ORGANIZATION PROFILE

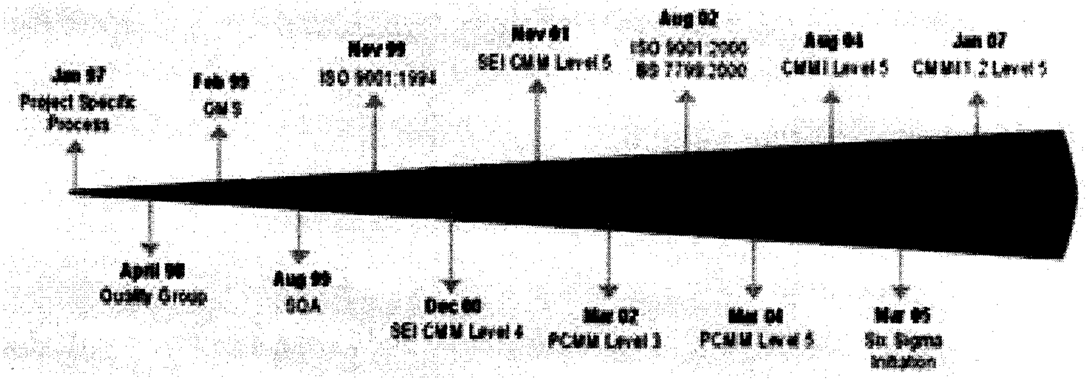
Caritor, incorporated and headquartered in USA, is a global IT consulting & Systems Integration firm that delivers high-quality IT services to leading clients around the world. We have been playing the role of a trusted IT partner to our clients since 1993 by helping them translate their IT vision into solid, measurable value. Today we have a presence that spans USA, UK, France, the Middle-East, India and Singapore with over 3684 Caritorians working across these locations to deliver winning solutions for our clients. We are long-term IT partners of our clients and help them realize measurable value from their business vision using technology.

Caritor offers cost-effective and intelligent IT solutions to clients in Financial Services, Communications, Retail, Manufacturing, High-Technology, Travel & Transportation and Public Sector industries. We offer IT services in the areas of Application Development, Application Management, Enterprise Business Solutions, Software Testing and Systems Integration through a global delivery model that ensures security, cost-effectiveness and quality for clients. We have successfully completed the BS-7799 second surveillance audit for information security practices at Caritor.Since our inception we have firmly believed in ensuring highest quality and security for the IT solutions that we deliver to clients. Our quality and security processes & certifications are a testament to this commitment - we are one of the very select companies in the world to be certified at SEI-CMM Level 5, PCMM Level 5, CMMI Level 5, ISO:9001 and the BS7799 standards. As part of our continuing quality initiatives we are also rolling out Six-Sigma processes internally

Caritor has independent teams for Quality Assurance and Process Management called SQA (Software Quality Assurance) and PMG (Process Management Group), which caters to verification, validation and process compliance to Caritor's BMS. SQA is an independent group responsible for the quality of deliverables and reports directly to the Quality Head of the organization. All project deliverables undergo SQA review before delivery to the customer. SQA conducts reviews of requirement specifications, design documents, software code as well as test plans and test cases. SQA carries out testing activities to validate the software and deliverables prior to customer release.

Caritor offers its product offerings under the brand "Certido™ ". Certido™ leverages Caritor's track record of developing commercial grade products and a decade of experience of working with world class customers, delivering best practices across industries. Certido currently offers a suite of outsourced service delivery management solutions under the brand, Certido™ Service Xcellence.

Certido™ Service Excellence provides superior customer service at lower cost and facilitates the efficient governance of outsourced service delivery operations. Certido™ Service Xcellence is currently used by four divisions of a Fortune 500 Organization to ensure reliable delivery of outsourced IT services to its employees. Certido™ Service Xcellence provides a multi-channel service framework and allows IT Organizations to track, measure and ensure adherence to pre-determined service level agreements.



LEVEL 5 PCMM 5 LEVEL CMMI 5 LEVEL ISO 9001 BS 7799

CHAPTER 2

SYSTEM STUDY AND ANALYSIS

2.1 EXISTING SYSTEM OVERVIEW

2.1.1 Pharmacy 2000

Pharmacy 2000 is a client-server proprietary software application Systems for integration into outpatient pharmacies operational environments. The use of Pharmacy 2000 allows the pharmacy to increase productivity by managing the fulfillment process while invoking the correct automated equipment. The three major components are Imaging, Filling or Checking.

- **Imaging:** Using the RxID bar code that was placed on the paper prescription, the data entry clerk will scan the bar code initiating the flatbed scanner to digitize the prescription. This process is repeated for every new prescription. A single prescription image can be associated to multiple related prescriptions.

- **Filling:** Functioning in parallel with the Imaging process, the building of the patient's order begins in the filling zone. Securing the laser label printed by the host system, the filling tech scans the RxID bar code alerting Pharmacy 2000 to start the fulfillment for the specific prescription. The filling station will fire the correct automation and direct the filling tech to the drug location. Upon confirming that the correct drug was retrieved, the filling tech scans a personal ID and is instructed in which packing box to place the prescription. At this point the interface may send a Fill complete message to the host with the identity of the filling person and time filled.

- **Checking:** When all prescriptions are filled within an order and placed in the packing box, an alert is provided to the Pharmacist directing the checking of the completed order. Assisting the pharmacist, the Checking station displays several sets of information allowing for a sound professional judgment. Displayed for review is data pertaining to the prescription as entered during the data entry process, the digital image of the prescription from the imaging process, Pill Image, and drug information from the database. The pharmacist scans his/her personnel ID after verifying the validity of the prescription; the Checking station will display each prescription of the order until complete. With pharmacist verification the interface will transmit a release message to the host systems when required.

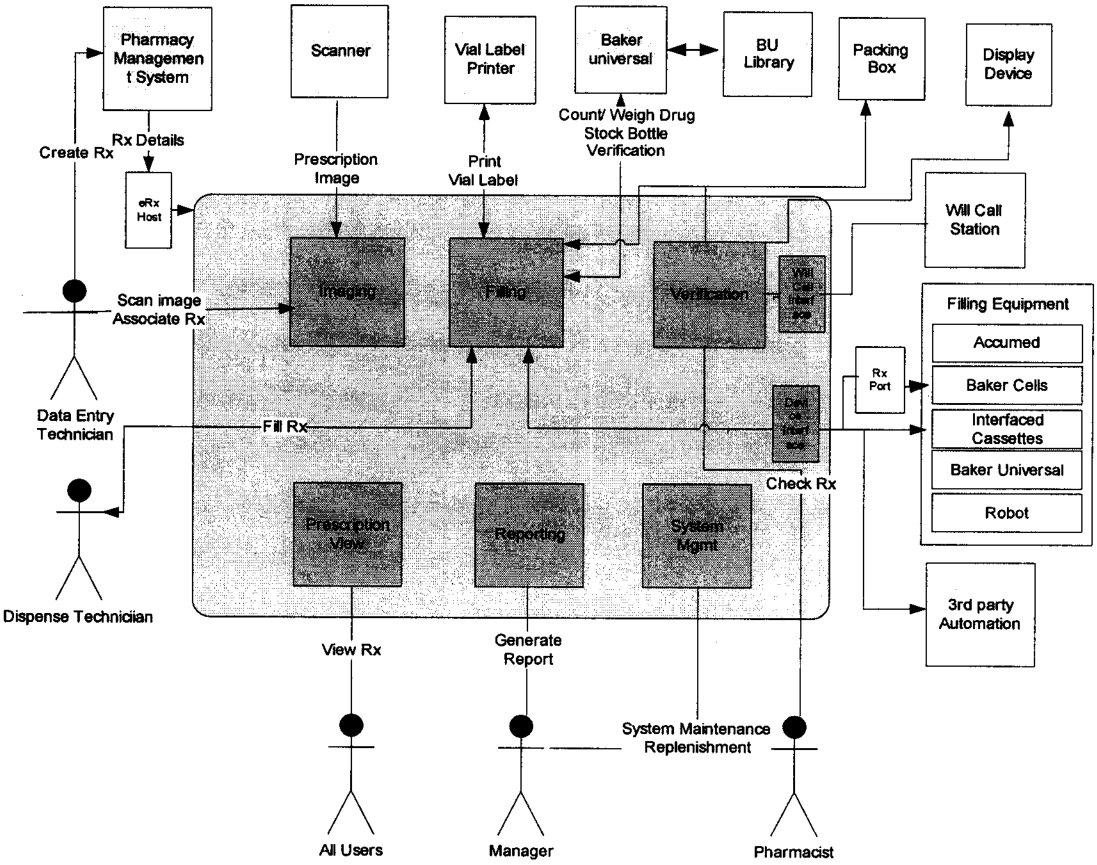


Figure 2.1.1 Pharmacy 2000

The message layout differs between parsers, but commonly contains the checking pharmacist's identification and date time stamp of verification. This concludes the Pharmacy 2000 involvement in the fulfillment process.

- **Prescription View:** This module allows viewing prescriptions and their status without disrupting the workflow in the Pharmacy 2000 system.

The View menu includes the following:

1. Status - This option allows an operator to check the status of any prescription, including current information and prescription history. With this ability, a customer's inquiry about their prescription can be quickly answered without disrupting the workflow within the pharmacy.
2. View Rx - The View Rx option allows the Pharmacist to look up the information for a prescription before it arrives at the Checking Station or after it has left.

- **Reporting:** The purpose of the reporting function is to generate pharmacy analysis on different aspects of the pharmacy process. The Pharmacy 2000 system can generate the following reports:
  - Prescription Volume
  - Personnel Activity
  - Dispensing Quantities
  - Partially Filled
  - Drug Locations
  - Rx Information

- **System Management/ Maintenance**: The purpose of the Maintenance function is to provide a method of configuring the Pharmacy 2000 system data

It includes the following:

- Personnel Setup

- Pharmacy 2000 Setup
- Drug Mapping and Replenishment
- Rejection Messages Setup
- Vial Sizes Setup
- System Clean Up
- Drug Substitution and Stock Bottle Association
- Backup
- Pill Images Update

2.1.2. AutoLink v3.0

AutoLink v3.0 is a configurable pharmacy filling solution that powers the AccuMed technology. By combining a software program with bar code technology and state-of-the-art counting and dispensing technology, AutoLink enhances the efficiency of pharmacy procedures while greatly reducing the risk of medication errors and improving the security of your pharmacy. AutoLink provides the most on-site user configurable product for maximum flexibility.

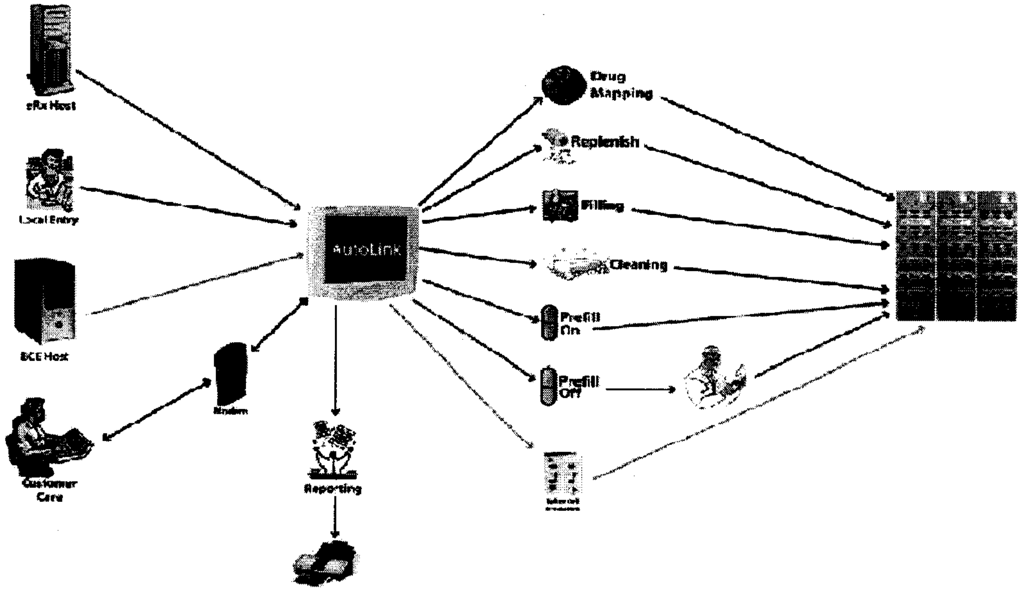


Figure 2.1.2 AutoLink v3.0

The software allows for multiple dispensing modes such as fill by drug, stock bottle, location, and Rx. Pharmacies have the option to use an interface that automatically dispenses the medication after adjudication. A prescription level interface, which provides robust dispensing tracking, or a traditional baker cell interface can be used. The security features are configurable to allow security not only by user, but also by procedure.

AutoLink software provides enhanced maintenance and inventory features that ensure the pharmacist have the ability to maintain optimal performance of the system. Color-coded graphics automatically alerts the user for replenishment and cleaning of the cells. Additionally, AutoLink tracks inventory, expiration dates, and lots numbers by cell. AutoLink prevents any drug that has been expired from being dispensed.

2.1.3. Pharmaserv (Pharmacy Management Suite)

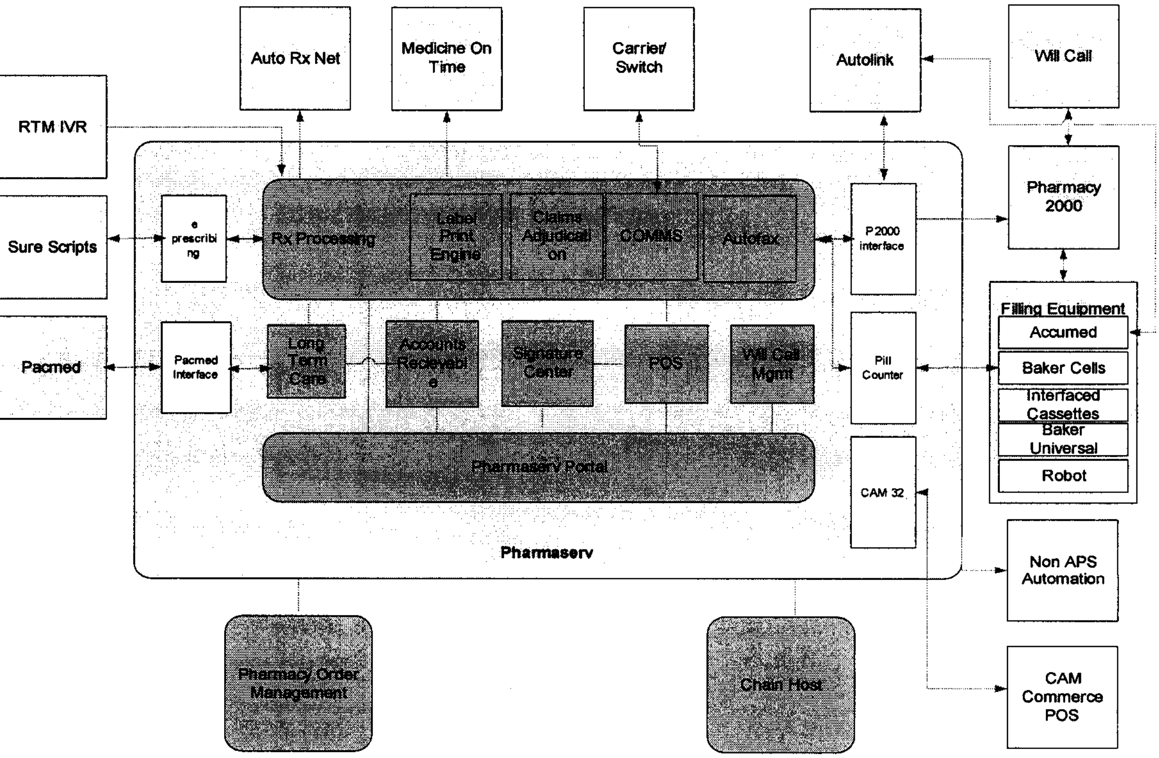


Figure 2.1.3 Pharmaserv (Pharmacy Management Suite)

Pharmaserv portal: This is the common location from where the various modules/programs in the Pharmacy management suite of products can be invoked. It is the all encompassing system, which serves as the entry point to the various Pharmaserv modules like Rx Processing, Long Term Care, Accounts Receivable, and Signature Center.

Rx Processing: This is the module where Rx records (prescriptions) are captured. Data such as the Rx number/ID, entered date and other pertinent information is entered by the clerk and is stored in the database. This information is then processed, and claims adjudication takes place. A label is printed for the Rx with a bar code containing the Rx information on it.

Label print engine: The label print engine prints data relevant to the Rx, such as the Rx number/ID, number of Rx's, etc in the form of a bar code and also a label to be pasted on the vial which will eventually be filled with the dispensed medication.

Claims Adjudication: This module takes care of sending patient information to the insurance company for claiming payment for the medication about to be dispensed. Co-pay, or complete payment of medication, or any other ways of payment for the medication are determined in this step. CAM uses COMMS as the communication interface module.

Long-term care: This module allows for scheduling of recurring dispensing of medication for healthcare institutions or customers/patients who are associated with or do business with the pharmacy on a long term/regular basis.

Accounts receivable: This module is responsible for handling information where records pertaining to credit are maintained. Patients who hold accounts in the pharmacy who are allowed credit, and who pay the bills after accumulating credit are maintained in accounts receivable module.

Signature center: This is the module where patient signatures are collected and stored before or during delivery of medication to them.

POS: Point of sale is the final step in the life cycle of a prescription. This is the point where the medication is finally delivered to the patient. The Rx /Order are marked as being 'completed'. The status of the Rx record is updated accordingly.

P2000 Interface: This serves as the interface between the PMS and the hardware devices that dispense the medication. P2000 interface also provides the information to Global interface.

Chain host: This is a system that contains data belonging to all the pharmacies that are part of a chain of pharmacies. Data from each of these stores is replicated on the Chain Host. Data pertaining to an Rx / order is made available to the other local stores if the need arises for the order to be recalled or completed from any one of the other stores that are part of the chain.

2.2 NEED FOR PROPOSED SYSTEM

The project deals with Integrating software product such as Pharmacy2000 (Pharmacy workflow system), AutoLink v 3.0 and Pharmaserv (Pharmacy Management Suit) and adds new functionality based on the customer requirement. The goal of this project is to provide an integrated workflow solution that will create a more efficient pharmacy operation by reducing medication errors and increasing both patient and user satisfaction. Additionally, the integrated product will leverage existing suite of products and add value that customers can identify with and that will intuitively drive the market to purchase the new suite of products.

- Meet or exceed customer's expectations in terms of their workflow needs with workflow encompassing all operational flows occurring within a retail / outpatient pharmacy.

- ➢ Provide the majority of key functionality currently available in Pharmacy2000, and AutoLink
- ➢ Scale well across customers of different sizes and market segments.
- ➢ Include some pathway for current customers to easily adopt the new solution.
- ➢ Work independent of products, or alternate vendor product.
- ➢ Interoperate with all other existing products.

2.3 PROPOSED SYSTEM

The Proposed System integrates software product such as Pharmacy2000 (Pharmacy workflow system), AutoLink v 3.0 and Pharmaserv (Pharmacy Management Suit) and adds new functionality based on the customer requirement. It provides majority of key functionality currently available in Pharmacy 2000, and Autolink. Leverages existing strengths of current product. Includes pathway for current customers to easily adopt the new solution.

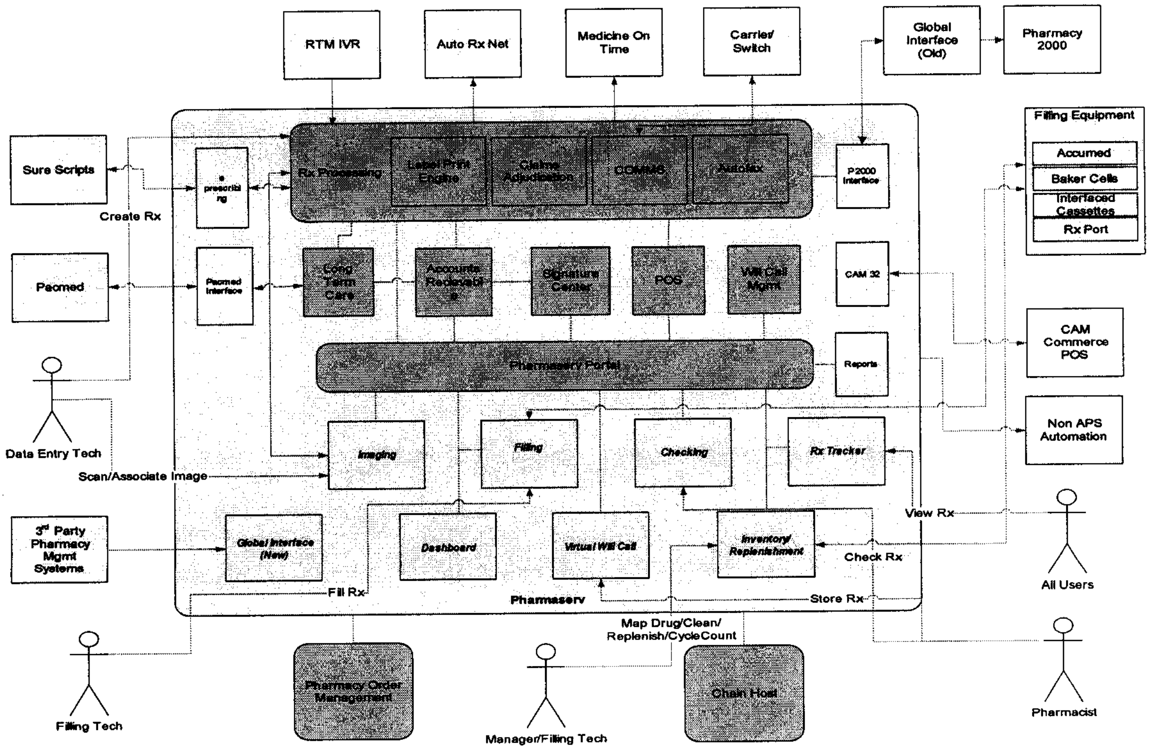


Figure 2.3 Proposed System

2.4 FEASIBILITY ANALYSIS

Feasibility analysis is the measure of how beneficial or practical the development of Information System will be to the Organization. Once the scope of the problem has been identified and initial analysis of the system has been completed, the feasibility of the project and the software to be developed is put under the test. This is done at this point to basically verify whether the software, which is to be developed, meets the scope and whether the project is feasible to do. Preliminary investigations would reveal whether the entire project would be possible to carry out in the current environment. The three stages of feasibility analysis are described below.

2.4.1 Technical Feasibility

Technical Feasibility is the measure of practicality of a specific technical solution and the availability of technical resources and expertise. It centers on the existing computer system (hardware, software, etc.) and to what extent it can support the new addition. The technical issues that have to dealt with in this phase are regarding the technology the system uses and its modifiability. It be verified that the system would be within the state of the art and defects, if any can be reduced to a level matching the application needs. Other considerations include:

- Existing of the necessary technologies.
- Capacity of the proposed system to hold the required data.
- Ability of the system to respond accurately to all inquiries.
- Scope of expandability of the system
- Technical guarantee of accuracy, reliability, ease of access and data security.

When all of the above mentioned features have been considered and the result is found to be positive, then the system development is said to be technically feasible.

2.4.2 Operational Feasibility

Operational Feasibility asks if the system will work when it is developed and installed. It checks for the support of the management, the current business methods, user's involvement and their attitude towards the proposed system, etc.

The proposed system is said to be operationally feasible only it meets all the requirements of the client. The proposed system has found encouraging from all type users as the system is said to flexible to the users and its operations are transparent as it the makes use of the loosely coupled architecture called as the service oriented architecture (SOA). The operations performed in the system are transparent as it provides for the supports variety of technologies besides the proper handshaking is followed.

2.4.3 Economic Feasibility

Economic feasibility is the measure of the cost-effectiveness of the proposed system. The investment to be made in the proposed system must prove a good investment to the organization by returning benefits equal to or exceeding the costs incurred in developing the system.

The proposed benefits of the system will outweigh the costs to be incurred during system developed since the system does not require procurement of additional hardware facilities it is economically feasible. In addition capability of the system to incorporate future enhancement will improve the performance to suit the future need of the client organization.

CHAPTER 3

DEVELOPMENT ENVIRONMENT

3.1 HARDWARE REQUIREMENTS

Server Requirements

Processor	Pentium 4
RAM	1.0 GB RAM (2x512) DDR2
CPU Clock	2.8 GHz or Above
Keyboard	102 Keys
Display	17 " SVGA 1024 X 768 Resolution or Above
Mouse	Standard Serial Mouse
Hard Disk	80GB SATA
Connectors	Edge port, Lava Link (Used to connect Automation equipment's to server)

Client Requirements

Processor	2.8 GHZ Pentium 4
RAM	512 MB RAM
CPU Clock	533 MHz or Above
Keyboard	102 Keys
Display	17 " SVGA 1024 X 768 Resolution or Above
Mouse	Standard Serial Mouse
Hard Disk	40GB SATA

3.2 SOFTWARE REQUIREMENTS

Operating System	Microsoft Windows Server 2003/NT 4.0/ XP
Front End	Microsoft Visual Basic.Net 2003
Back End	Microsoft SQL Server 2000

3.3 SOFTWARE DESCRIPTION

3.3.1 Visual Basic .Net 2003 (Front End Tool)

Visual Studio .NET is a complete set of development tools for building ASP Web applications, XML Web services, desktop applications, and mobile applications. Visual Basic .NET, Visual C++ .NET, Visual C# .NET, and Visual J# .NET all use the same integrated development environment (IDE), which allows them to share tools and facilitates in the creation of mixed-language solutions. In addition, these languages leverage the functionality of the .NET Framework, which provides access to key technologies that simplify the development of ASP Web applications and XML Web services.

Extensible Markup Language (XML) provides a method for describing structured data. XML is a subset of SGML that is optimized for delivery over the Web. The World Wide Web Consortium (W3C) defines XML standards so that structured data will be uniform and independent of applications. Visual Studio .NET fully supports XML, providing the XML Designer to make it easier to edit XML and create XML schemas.

The Visual Studio .NET integrated development environment now includes tools for developing applications for smart devices, such as the Pocket PC. Using the tools and the .NET Compact Framework, a subset of the .NET Framework, you can create, build, debug, and deploy applications that run on the .NET Compact

Framework in personal digital assistants (PDAs), mobile phones, and other resource-constrained devices.

ASP.NET Mobile Designer extends ASP.NET and the .NET Framework, allowing you to build Web applications for mobile phones, PDAs, and pagers. This designer is integrated into the Visual Studio IDE. You can create mobile Web applications, use the Mobile Designer to modify a mobile Web form, and then build and run the application, all from within Visual Studio.

Visual Studio .NET provides several project templates that you can use to get started on developing distributed applications. Enterprise templates define the initial structure of a distributed application and also provide architectural and technological guidance for application design. In addition to pre-defined enterprise templates, you can also create custom templates that can be used by developers in a team environment.

.NET Framework

The .NET Framework is a multi-language environment for building, deploying, and running XML Web services and applications. It consists of three main parts:

Common Language Runtime: Despite its name, the runtime actually has a role in both a component's runtime and development time experiences. While the component is running, the runtime is responsible for managing memory allocation, starting up and stopping threads and processes, and enforcing security policy, as well as satisfying any dependencies that the component might have on other components. At development time, the runtime's role changes slightly; because it automates so much (for example, memory management); the runtime makes the developer's experience very simple, especially when compared to COM as it is today. In particular, features such as reflection dramatically reduce the amount of code a developer must write in order to turn business logic into a reusable component.

Unified programming classes: The framework provides developers with a unified, object-oriented, hierarchical, and extensible set of class libraries (APIs). Currently, C++ developers use the Microsoft Foundation Classes and Java developers use the Windows Foundation Classes. The framework unifies these disparate models and gives Visual Basic programmer's access to class libraries as well. By creating a common set of APIs across all programming languages, the common language runtime enables cross-language inheritance, error handling, and debugging.

ASP.NET: ASP.NET builds on the programming classes of the .NET Framework, providing a Web application model with a set of controls and infrastructure that make it simple to build ASP Web applications. ASP.NET includes a set of controls that encapsulate common HTML user interface elements, such as text boxes and drop-down menus. These controls run on the Web server, however, and push their user interface as HTML to the browser. On the server, the controls expose an object-oriented programming model that brings the richness of object-oriented programming to the Web developer. ASP.NET also provides infrastructure services, such as session state management and processes recycling that further reduce the amount of code a developer must write and increase application reliability. In addition, ASP.NET uses these same concepts to enable developers to deliver software as a service. Using XML Web services features, ASP.NET developers can write their business logic and use the ASP.NET infrastructure to deliver that service via SOAP.

Visual Studio supports the Microsoft .NET Framework, which provides the common language runtime and unified programming classes; ASP.NET uses these components to create ASP.NET Web applications and XML Web services. Also included is the MSDN Library, which contains all the documentation for these development tools.

Visual Basic Language

Visual Basic .NET, the next generation of the Visual Basic language, is a fast and easy way to create .NET-based applications, including XML Web services and Web applications. Visual Basic .NET has many new and improved features that make it a powerful object-oriented programming language, including inheritance, interfaces, and overloading. Other new language features include free threading and structured exception handling. Visual Basic .NET also fully integrates the .NET Framework and the Common Language Runtime, which provide language interoperability, garbage collection, enhanced security, and improved versioning support.

Inheritance

Visual Basic .NET supports inheritance by allowing you to define classes that serve as the basis for derived classes. Derived classes inherit and can extend the properties and methods of the base class. They can also override inherited methods with new implementations. All classes created with Visual Basic .NET are inheritable by default. Because the forms you design are really classes, you can use inheritance to define new forms based on existing ones.

Exception Handling

Visual Basic .NET supports structured exception handling, using an enhanced version of the Try...Catch...Finally syntax supported by other languages such as C++. Structured exception handling combines a modern control structure (similar to Select Case or While) with exceptions, protected blocks of code, and filters. Structured exception handling makes it easy to create and maintain programs with robust, comprehensive error handlers.

Overloading

Overloading is the ability to define properties, methods, or procedures that have the same name but use different data types. Overloaded procedures allow you to provide as many implementations as necessary to handle different kinds of data, while giving the appearance of a single, versatile procedure.

Overriding Properties and Methods

The Overrides keyword allows derived objects to override characteristics inherited from parent objects. Overridden members have the same arguments as the members inherited from the base class, but different implementations. A member's new implementation can call the original implementation in the parent class by preceding the member name with MyBase.

Constructors and Destructors

Constructors are procedures that control initialization of new instances of a class. Conversely, destructors are methods that free system resources when a class leaves scope or is set to Nothing. Visual Basic .NET supports constructors and destructors using the Sub New and Sub Finalize procedures.

Data Types

Visual Basic .NET introduces three new data types. The Char data type is an unsigned 16-bit quantity used to store Unicode characters. It is equivalent to the .NET Framework System.Char data type. The Short data type, a signed 16-bit integer, was named Integer in earlier versions of Visual Basic. The Decimal data type is a 96-bit signed integer scaled by a variable power of 10. In earlier versions of Visual Basic, it was available only within a Variant.

Interfaces

Interfaces describe the properties and methods of classes, but unlike classes, do not provide implementations. The Interface statement allows you to declare interfaces, while the Implements statement lets you write code that puts the items described in the interface into practice.

Delegates

Delegates — objects that can call the methods of objects on your behalf — are sometimes described as type-safe, object-oriented function pointers. You can use delegates to let procedures specify an event handler method that runs when an event occurs. You can also use delegates with multithreaded applications.

Shared Members

Shared members are properties, procedures, and fields that are shared by all instances of a class. Shared data members are useful when multiple objects need to use information that is common to all. Shared class methods can be used without first creating an object from a class.

References

References allow you to use objects defined in other assemblies. In Visual Basic .NET, references point to assemblies instead of type libraries.

Namespaces

Namespaces prevent naming conflicts by organizing classes, interfaces, and methods into hierarchies.

Assemblies

Assemblies replace and extend the capabilities of type libraries by, describing all the required files for a particular component or application. An assembly can contain one or more namespaces.

Attributes

Attributes enable you to provide additional information about program elements. For example, you can use an attribute to specify which methods in a class should be exposed when the class is used as a XML Web service.

Multithreading

Visual Basic .NET allows you to write applications that can perform multiple tasks independently. A task that has the potential of holding up other tasks can execute on a separate thread, a process known as multithreading. By causing complicated tasks to run on threads that are separate from your user interface, multithreading makes your applications more responsive to user input.

3.3.2 Microsoft SQL Server 2000 (Back End Tool)

Microsoft SQL Server 2000 is capable of supplying the database services needed by extremely large systems. Large servers may have thousands of users connected to an instance of SQL Server 2000 at the same time. SQL Server 2000 has full protection for these environments, with safeguards that prevent problems, such as having multiple users trying to update the same piece of data at the same time. SQL Server 2000 also allocates the available resources effectively, such as memory, network bandwidth, and disk I/O, among the multiple users.

Extremely large Internet sites can partition their data across multiple servers, spreading the processing load across many computers, and allowing the site to serve thousands of concurrent users. Multiple instances of SQL Server 2000 can be run on a single computer.

Features of SQL Server 2000

The database component of Microsoft SQL Server2000 is a Structured Query Language (SQL)–based, scalable, relational database with integrated Extensible Markup Language (XML) support for Internet applications.

Microsoft SQL Server 2000 features include:

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 Windows98 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 modifications back to the publisher.

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.

CHAPTER 4

SYSTEM DESIGN AND DEVELOPMENT

4.1 ARCHITECTURAL DESIGN

This System is implemented using the various layers in the software architecture. It comprises of the five layers to achieve the architectural flexibility and reuse. The following architecture is followed for migration.

- UI [User Interface]
- UIP [User Interface Process]
- Service
- Business Conditions
- Database

User Interface: This is used to design the forms and controls that are used for the migration. All the front-end logic is taken care by this layer.

User Interface Process: This layer holds the validation logic. All the front-end validations are done in this layer. The user name, password, source server, target server validations are done through this layer.

Service Layer: This layer holds the type of service that the utility is about to provide. There are two different types of services provided, the window service and the web service. When the type of service is to be changed from the window to web, this layer can be modified.

Business Conditions: This layer encapsulates the various automated services based on the Business rules. This layer provides the standard based service interface for the database layer.

Database Layer: This layer is used to communicate directly with the database. Whenever a process needs to communicate with the database it is done through the DB Layer.

DTO: The DTO are the Data Transfer Objects. They are the class which holds the objects that can be passed through the different layers. This DTO can be accessed from any part of the layers.

The flow of control moves from the UI layer towards the UIP, Service, Business and the Database Layer and vice versa. At any cost the UI layer cannot directly communicate with the business or DB layer. The communication flows in a top-down and bottom-up approach.

4.2 ACTVITY DIAGRAM

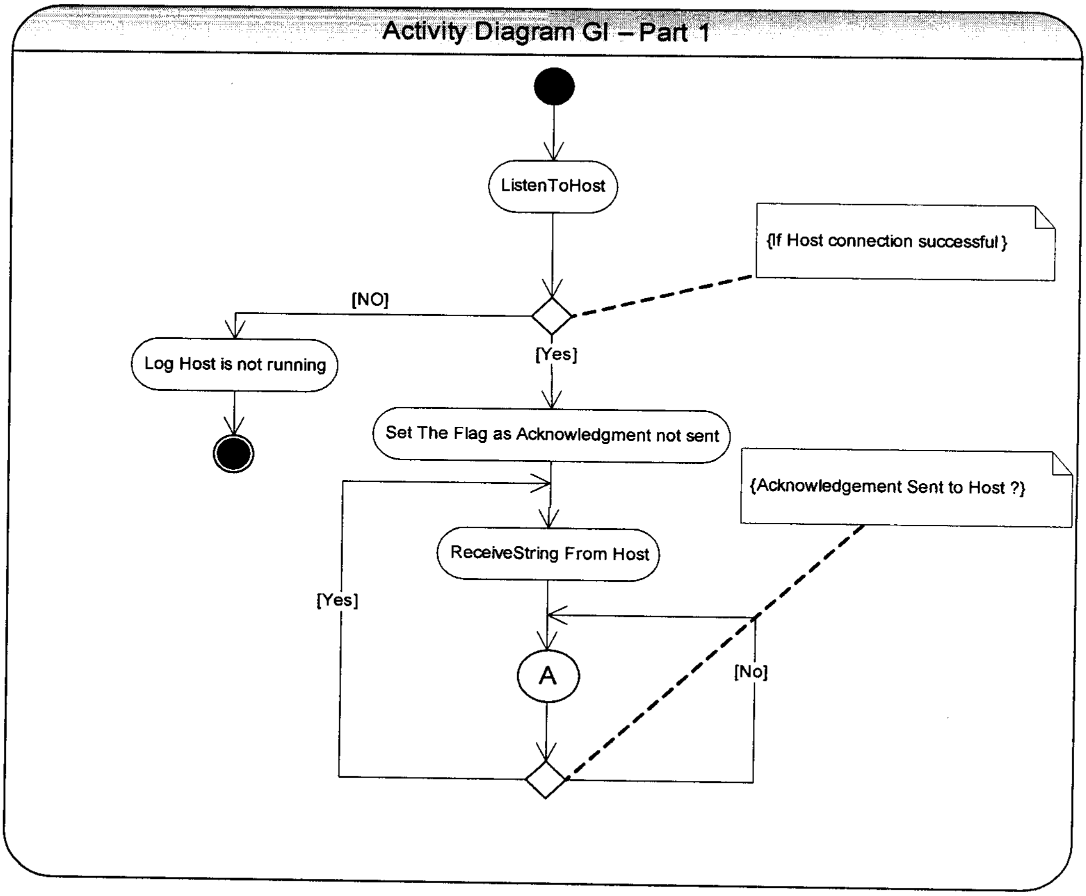

Activity Diagram GI – Part 1
ListenToHost
{If Host connection successful}
[NO]
Log Host is not running
[Yes]
Set The Flag as Acknowledgment not sent
{Acknowledgement Sent to Host ?}
ReceiveString From Host
[Yes]
A
[No]

Figure 4.2.1 Activity Diagram GI-Part 1

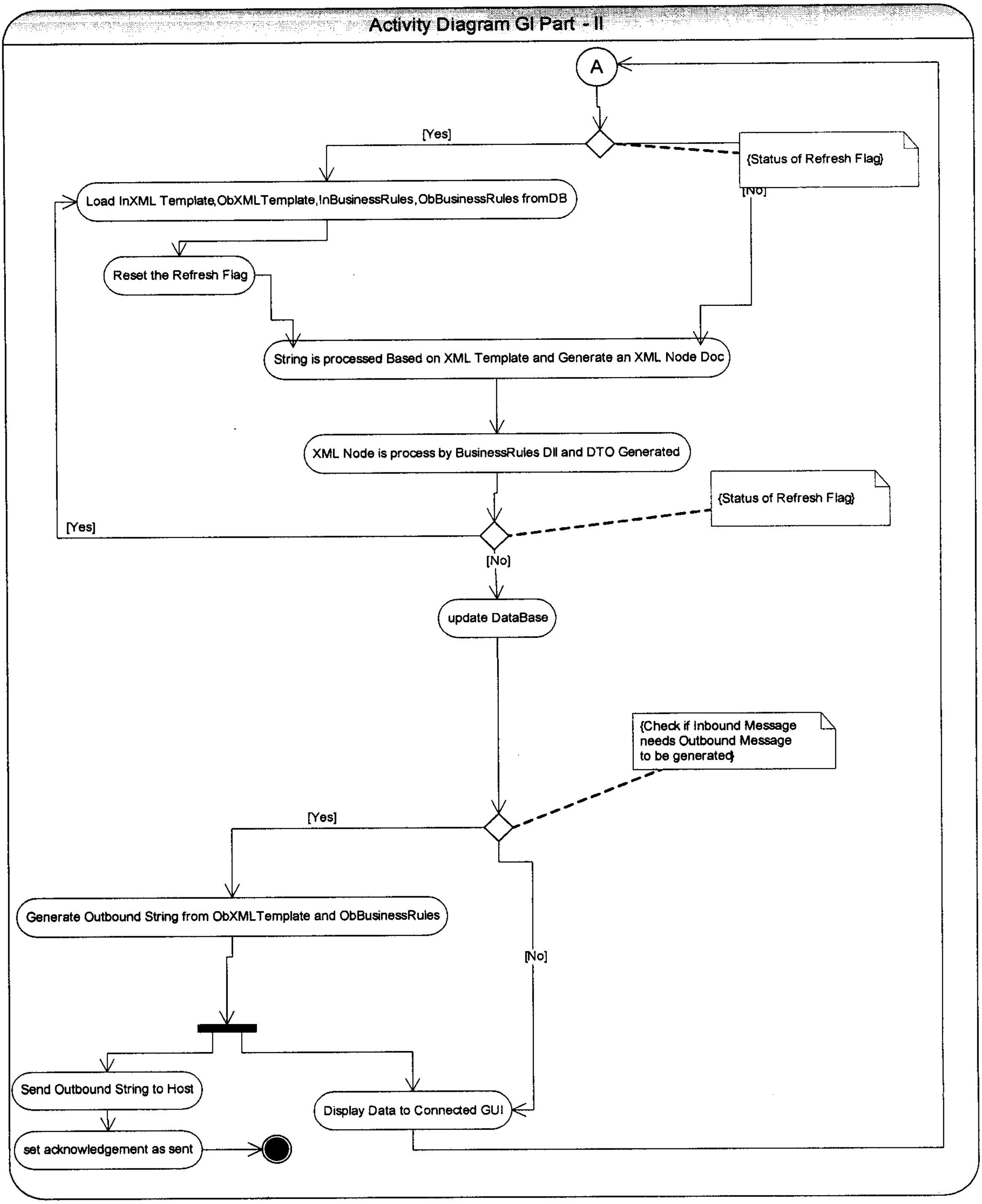

Activity Diagram GI Part - II
A
[Yes]
{Status of Refresh Flag}
[No]
Load InXML Template,ObXMLTemplate,InBusinessRules,ObBusinessRules fromDB
Reset the Refresh Flag
String is processed Based on XML Template and Generate an XML Node Doc
XML Node is process by BusinessRules Dll and DTO Generated
{Status of Refresh Flag}
[Yes]
[No]
update DataBase
{Check if Inbound Message needs Outbound Message to be generated}
[Yes]
Generate Outbound String from ObXMLTemplate and ObBusinessRules
[No]
Send Outbound String to Host
set acknowledgement as sent
Display Data to Connected GUI

Figure 4.2.2 Activity Diagram GI-Part 2

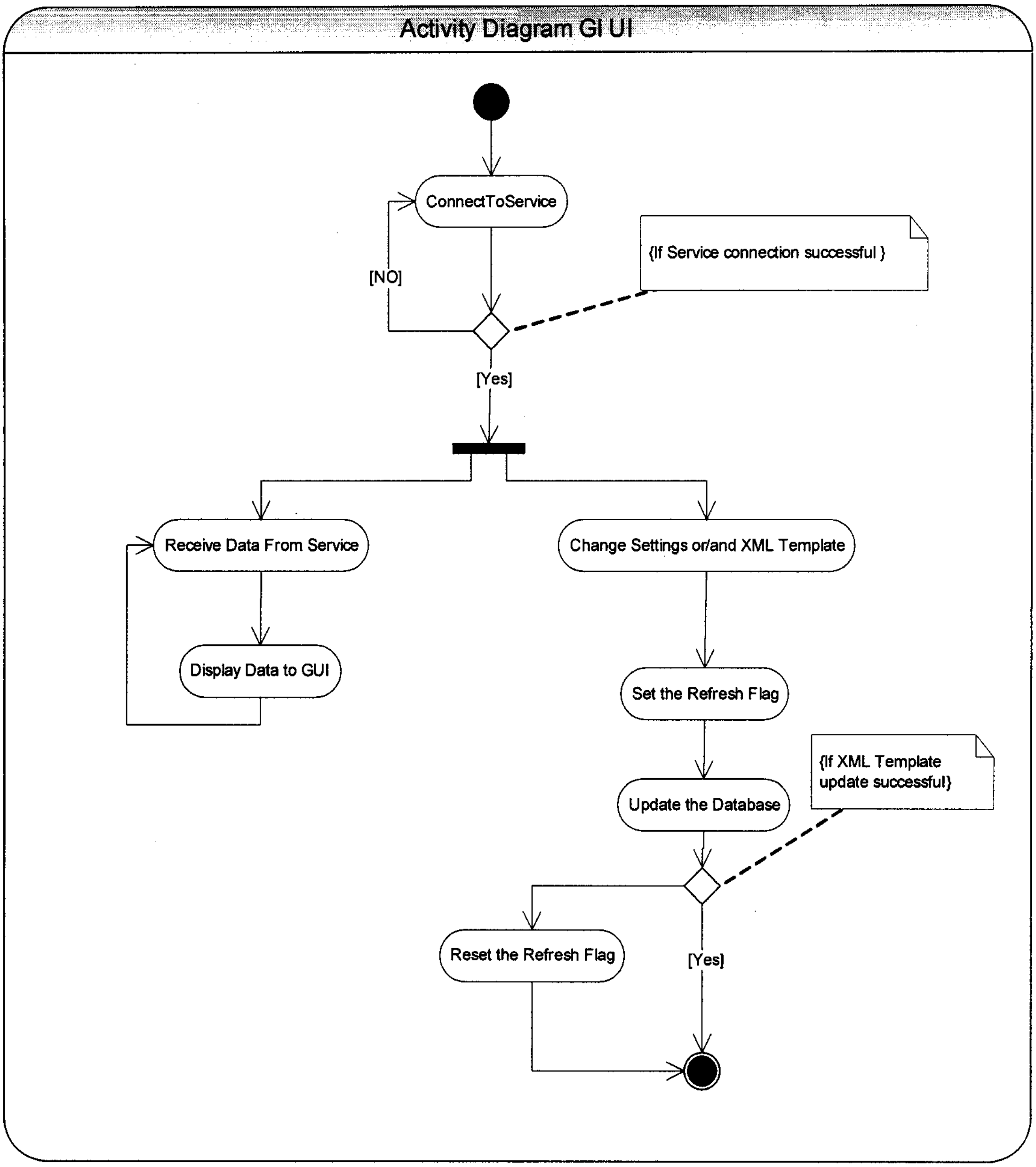
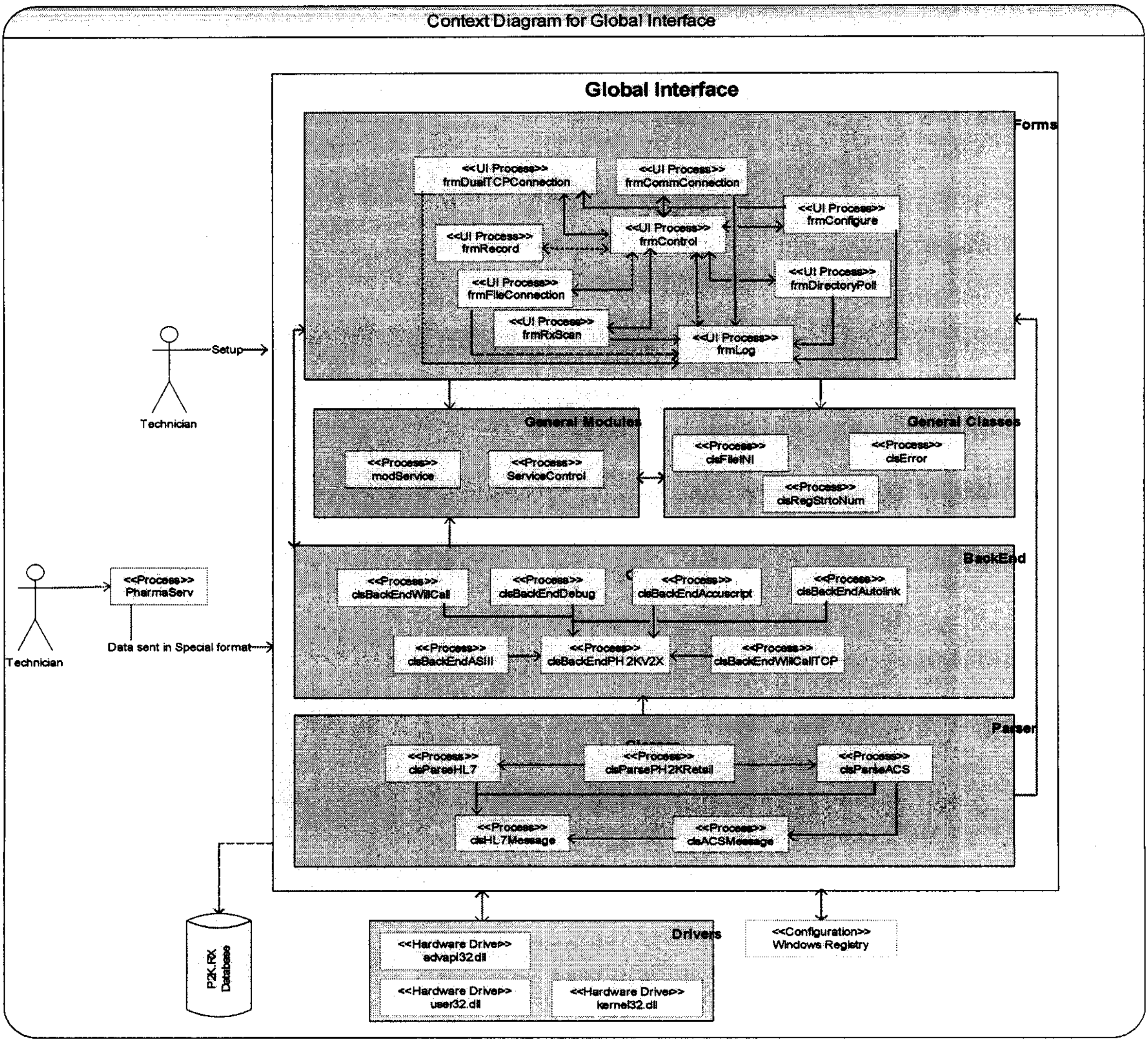


Figure 4.2.3 Activity Diagram GI-User Interface

4.3. DATAFLOW DIAGRAM

Data flow diagrams are graphical representation depicting information regarding the flow of control and the transformation of data from input to output. The DFD may be used to represent the system or software at any level of abstraction. In fact, DFD can be partitioned into levels. A Level 0 DFD called Context Level Diagram represents the entire software system as a single bubble with its interactions. The Context diagram shows the overall system with the users who will be interacting with it.

4.3.1 CONTEXT DIAGRAM (LEVEL- 0)



Context Diagram for Global Interface
Global Interface
Forms
<<UI Process>> frmDualTCPConnection
<<UI Process>> frmCommConnection
<<UI Process>> frmConfigure
<<UI Process>> frmRecord
<<UI Process>> frmControl
<<UI Process>> frmFileConnection
<<UI Process>> frmDirectoryPoll
<<UI Process>> frmRxScan
<<UI Process>> frmLog
Technician
Setup
General Modules
<<Process>> modService
<<Process>> ServiceControl
General Classes
<<Process>> clsFileINI
<<Process>> clsError
<<Process>> clsRegStrtoNum
BackEnd
Technician
<<Process>> PharmaServ
Data sent in Special format
<<Process>> clsBackEndWillCall
<<Process>> clsBackEndDebug
<<Process>> clsBackEndAccuscript
<<Process>> clsBackEndAutolink
<<Process>> clsBackEndASIII
<<Process>> clsBackEndPH 2KV2X
<<Process>> clsBackEndWillCallTCP
Parser
<<Process>> clsParseHL7
<<Process>> clsParsePH2KRetail
<<Process>> clsParseACS
<<Process>> clsHL7Message
<<Process>> clsACSMessage
P2K.RX Database
Drivers
<<Hardware Driver>> advapi32.dll
<<Hardware Driver>> user32.dll
<<Hardware Driver>> kernel32.dll
<<Configuration>> Windows Registry

Figure 4.3.1 Context Diagram (Level 0)

4.3.2 LEVEL-1 DATA FLOW DIAGRAM

The Level 1 DFD will explain the major modules in the whole system, i.e., how the data flow between each of these modules.

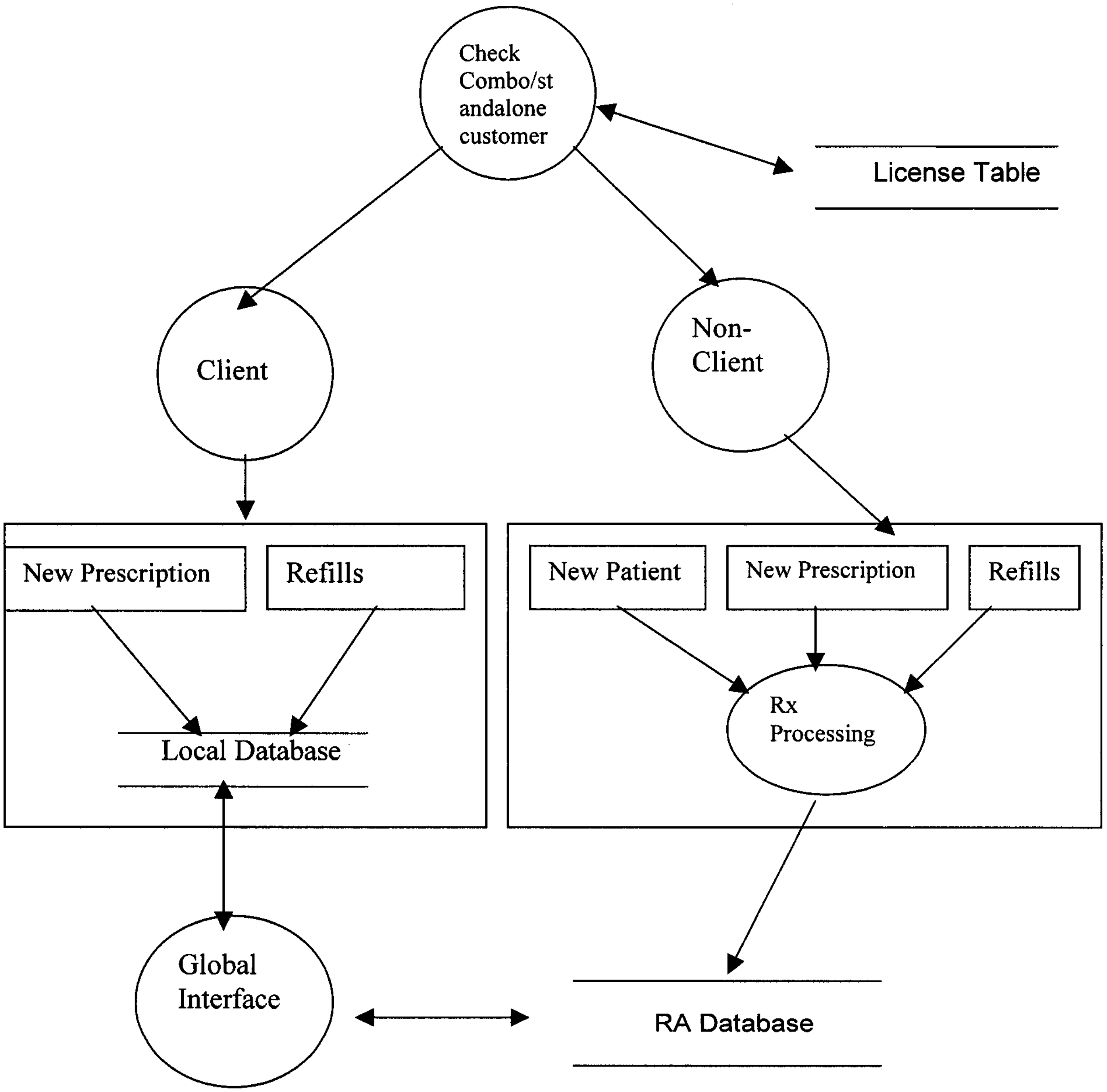


Figure 4.3.2 Level 1 Diagram

4.4 DATA BASE DESIGN

Database

A database is a collection of interrelated data stores with a minimum of redundancy to serve many applications. It minimizes the artificially embedded in using separate files. The primary objectives are fast response time to inquires, more information at low cost control of redundancy, clarity and easy to use, accuracy and fast recovery. The overall development in the database is to treat data as an organizational resource and as an integrated whole.

Data Integration

In a database, information from several files is co-ordinate, accessed and operated upon as though it is in a single file.

Date integrity

In a database all data are stored in a place and it allows each application to access it. This approach results in more consistent information.

Data Independence

This objective seeks to allow changes and organization of physical data with out re-programming the application. The approach used to design the system takes care of these objectives into considerations so as to organize the data.

The database design is used to prepare the tables used in the system.

4.4.1 Tables

SNo	Table Name	Description
1	PRESCRIPTION	Contains Prescription information
2	DUR	Contains DUR information
3	DISPENSESTATUS	Contains Dispense Status information
4	DELIVERYMETHOD	Required for Delivery Method information
5	STORES	Contains Stores information
6	PRESCRIBER	Contains Prescriber information
7	ORGPRESCRIBER	Contains Prescriber information related to the organization
8	PATIENT	Contains Patient information
9	ORGPATIENT	Contains Patient information related to the organization
10	DRUGS	Contains drugs information
11	ORGDRUGS	Contains drugs information related to the organization
12	BUSINESSRULES	Contains Business Rules required for parsing the incoming message strings from the host

Table 4.4.1 Global Interface Tables

4.4.2 Dependency

The Dependencies are as follows:

PRESCRIPTION

PrescriberId

The PrecriberId is updated in the PRESCRIBER table using the following rules:

The information received from the parser may be either PrecriberId or DoctorDEA. This value is compared with the PRESCRIBER: DoctorDEA column.

If this value is present in the DoctorDEA column, then the PrescriberId corresponding to DoctorDEA is retrieved from PRESCRIBER table and updated into the PRESCRIPTION table. The PRESCRIBER table is updated the new valued received. If this value is not present in the DoctorDEA field, then a new row is created in the PRESCRIBER table with this new value of PrecriberId/DoctorDEA received. The PrescriberId corresponding to this new row is retrieved from PRESCRIBER table and updated into the PRESCRIPTION table.

PatientId

The PatientId is updated in the PRESCRIBER table using the following rules:

The FamilyId, FirstName and LastName, received from the parser, are taken as a composite key. This composite value is compared with the PATIENT: FamilyId, FirstName, LastName columns. If this composite value is present in the PATIENT table, then the corresponding PatientId is retrieved and updated into the PRESCRIPTION table. The PATIENT table is updated with the new values received.

If this value is not present in the PATIENT table, then a new row is created with these values of FamilyId, FirstName and LastName. The PatientId corresponding to this new row is retrieved and updated into the PRESCRIPTION table.

Other dependencies if any will be taken care by the stored procedures.

4.4.3 Stored Procedures

The following are the stored procedures used by Global Interface:

SNo	SP Name	Description
1	PrescriptionDetails_Usp	This procedure will update the Prescription details
2	PatientDetails_Usp	This procedure will update the Patient details
3	PrescriberDetails_Usp	This procedure will update the Prescriber details
4	DURDetails_Usp	This procedure will update the DUR details
5	DrugsDetails_Usp	This procedure will update the Drug details
6	Sys_BusinessRule_Ssp	This procedure will read the Business Rules
7	Sys_BusinessRule_UsP	This procedure will update Business Rules
8	Getdetails_Ssp	This procedure will get the parser details
9	HCPrescriptionLabel_Ssp	This procedure will determine whether the Prescriber Label should be printed or not
10	Rxp_HcLabelPrint_Ssp	This procedure will supply the string to be printed along with the HC Label
11	Rxp_FlagUpdation_Usp	This procedure will set the flag after the printing is done

Table 4.4.3 Stored Procedure

4.4.4 Sample Table Structure

Database Name: Final_Database

Table Name : PATIENT

Column Name	**Data Type**	**Length**	**Allow Nulls**
PatientID	int	4	
FamilyID	Varchar	18	✓
FirstName	Varchar	50	✓
LastName	Varchar	50	✓
SecurityNumber	Varchar	30	✓
Address	Varchar	255	✓
City	Varchar	50	✓
State	Varchar	20	✓
Zipcode	Varchar	9	✓
PhoneNumber	Varchar	30	✓
Renumber	int	4	
CreatedBy	Varchar	30	
ModifiedBy	Varchar	30	
CreatedDateTime	Datetime	8	
ModifiedDateTime	Datetime	8	

Table 4.4.4.1 PATIENT

Database Name: Final_Database

Table Name : PRESCRIBER

Column Name	**Data Type**	**Length**	**Allow Nulls**
PrescriberID	int	4	
DocterDEA	Varchar	10	✓
PrescriberName	Varchar	30	✓
PrescriberAddress	Varchar	30	✓
PrescriberCity	Varchar	20	✓
PrescriberState	Varchar	2	✓
PrescriberZip	Varchar	9	✓
PrescriberPhone	Varchar	15	✓
RowVersion	int	4	
CreatedBy	Varchar	30	
ModifiedBy	Varchar	30	
CreatedDateTime	Datetime	8	
ModifiedDateTime	Datetime	8	

Table 4.4.4.2 PRESCRIBER

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 INTRODUCTION

System Implementation is the part of the software engineering life cycle, where, the design artifacts are converted to a working application. Coding is done in this stage using an apt framework and programming language, which would solve the specific problem the best way. Once the design is coded into a working application, it has to be verified, validated and tested in detail. The tested product if successful is deployed in the user environment.

5.2 SYSTEM VERIFICATION

System Verification answers the question "Am I building the product right?" It includes the review of interim work steps and interim deliverables during a project to ensure they are acceptable. Verification also determines if the system is consistent, adheres to standards, uses reliable techniques and prudent practices, and performs the selected functions in the correct manner. In data access, it verifies whether the right data is being accessed, in terms of the right place and in the right way.

For e.g., the drop downs gather data from the database, so each dropdowns should be verified whether they are bound to the correct database field. It is done during development of the key artifacts. Verification is a demonstration of consistency, completeness, and correctness of the software at each stage and between each stage of the development life cycle. In Result Analysis, verification is done during the development itself. Each database bindings are verified after binding to test whether the control is bound to the right data field.

5.3 SYSTEM VALIDATION

Validation answers the question “Am I building the right product?” This checks whether the developer is moving towards the right product, whether the development is moving towards the actual intended product that was agreed upon in the beginning. Validation also determines if the system complies with the requirements and performs functions for which it is intended and meets the organization’s goals and user needs. It is traditional and is performed at the end of the project. In data access, it checks whether we are accessing the right data, in terms of data required to satisfy the requirement.

Validation is performed after a work product is produced against established criteria ensuring that the product integrates correctly into the environment. It determines the correctness of the final software product by a development project with respect to the user needs and requirements.

Functional validation is done in the Automation of Pharmacy System to check whether each of the functions is done correctly as expected in every page. Each control in a Screen is designed to do some function. These functions are checked against the requirements stated for them. For e.g., clicking ‘Save’ button should take the corresponding action of saving the details into the database. Clicking the Edit icon should allow one to edit the contents that are being currently displayed. This level of validation can continue to all the controls in the system. This checking is usually done after the system is developed so that all activities that are affected can be checked.

Field level validation is done in Result Analysis to check whether each of the fields either accepts the data as expected and do the client side validation of data entered. For e.g. a field level validation on a text box would check against the type of data entered and follow rules such as length of entry etc.

The data type validation checks are conducted after the form is submitted. It takes place in the Action Form class of the struts framework. If the validation check fails then the processing stops and the control returns back to the original form that was submitted.

The validation is done in a step by step process. First the screen is loaded with the controls. When the user moves between controls on the screen, the validation events for the control that lost the focus are fired and appropriate error messages (if any) are displayed.

5.4 TESTING

Testing is a critical element of software quality and assurance and represents the ultimate review of specification design and coding. It is a vital activity that has to be enforced in the development of any system. This could be done in parallel during all the phases of system development. The feedback received from these tests can be used for further enhancement of the system under consideration. The testing phase conducts test using the Software Requirement Specification as a reference and with the goal to see whether the system satisfies the specified requirements.

Standard procedures have been followed in testing Result Analysis. Test cases are generated for each screen. These test cases will cover every possibility, which could result in both positive and negative results. These test plans are maintained for any further testing done on the system. The test plan stores information such as, the test script/input, expected output, actual output, comments and the name of the tester. This plan will be followed for all types of testing done in the system.

The main types of tests carried out on Automation of Prescription System are:

5.4.1 Unit Testing

Module or Unit Testing is the process of testing all the program units that make up a system. Unit testing focuses on an individual module thus allowing one to uncover all the errors made logically and while coding in the module.

In This Project each page is tested separately as a unit. Initially the flow of control and data through that page is checked. When considering a module as a unit, the flow of data and control through the whole module is tested. The result is stored in the test plan. In a page, each control is further tested in unit testing. The process is done in all the pages of the system. Once the errors are rectified, the testing procedure is repeated with same test cases to ensure this hasn't produced new errors. Hence this is a continuous process.

5.4.2 Integration Testing

Integration testing tests the process of integrating the various modules to form the completed system. Integration starts with a set of units each individually tested in isolation and ends when the entire application has been built. Integration testing verifies that the combined units function together correctly. It facilitates in finding problem that occur at interface or communication between the individual parts.

Automation of Pharmacy System followed top-down integration testing. Modules were linked to the main menu in a sequence as required in the real time operating mode of the system. The flow of data through the whole module in the upper level is taken and executed. A change of data made in one screen should have reflected in all other screens.

This process is continued from the page level to module level, finally to the system level. In the final stage, the whole system is taken together and tested for integration. A change in one place should be reflected through out the system. Regression testing is done after each change made into the software. This tests if the change has affected any part of Automation of Pharmacy System negatively after the change was made. The whole set of test cases need to be run again to do the regression testing.

5.4.3 System Testing

System testing is actually a serious of different tests, whose primary purpose is to fully exercise the computer-based system. This helps in verifying that all the system elements have been properly integrated and perform the allocated functions. It verifies the entire product after having integrated all software and hardware components, and validates it according to the original project requirement. The system testing takes into consideration the hardware, and the software. That is, Automation of Pharmacy System should be able to be run on the specified hardware for variety of cases. Automation of Pharmacy System is tested against recovery from errors.

5.4.3.1 Security Testing

Security testing is important in system testing. The system in no way shall be accessible to unauthorized users. Testing is done to ensure that a user with respective rights can only view the various forms and reports presented by Automation of Pharmacy System. If users try to perform something beyond his assigned rights corresponding messages should be displayed. The Automation of Pharmacy System in such cases redirects the user back to the previous page in such a case.

Another security issue involves the sensitive data in the system. The system is highly secure with authentication fixed at various levels of the hierarchy.

One more level of security is concerned with user rights. Each user is applied rights module wise. The menus can be configured to roles.

Users can also be configured to roles. Menu items are assigned to users dynamically based on the roles assigned to menu items as well as users. A match is done before displaying the menu to the user.

5.4.3.2 Stress Testing

Stress Testing executes a system in a manner that demands resources in abnormal quantity, frequency or volume. Automation of Pharmacy System was stress tested by having user simultaneous access to various modules in the system.

5.4.3.3 Regression Testing

Regression Testing is the testing that is performed after functional improvements or corrections have been made to the system to confirm that the changes have created no unintended side effects. Regression tests were run each time a module is replaced to ensure that a new build has not impaired the functioning of previous builds.

5.5 TEST PLAN

The Test plan decides on how the functionalities of the system will be proved and what test data are used. The Test data can rarely be comprehensive enough to test the whole system. Some aspects of the system can be tested only during live operation. The test was done in the following areas with appropriate test data.

Program Input

1. Form Design
2. Data Conversion
3. Data Transmission
4. Data Control
5. Validation
6. Error Routine and Correction Messages

Program Output

1. User acceptance of output
2. Report Design
3. Data Transmission
4. Distribution of output
5. Error handling procedure

5.6 SAMPLE TEST CASES

SNo	Module	Test Step Description	Expected Result	Actual Result
1	Global Interface	Click on the 'Global Interface UI' icon in the desktop.	System should display the logon screen with the UserID, Password fields and Login, Exit buttons and request for authentication.	System displayed the logon screen with the UserID, Password fields and Login, Exit buttons and request for authentication.
2	Global Interface	Validate the field length and type of the fields in the screen.	User Id field should be a textbox with size 30 and mandatory field. Password field should be a masked textbox of size 30 and mandatory field.	User Id field has a textbox with size 30 and mandatory field. Password field has a masked textbox of size 30 and mandatory field.
3	Global Interface	Validate for the font, spacing, borders & user friendliness.	All screens should be according to the standards.	All screens are according to the standards.
4	Global Interface	Validate the tab order.	The tab order should be Password->Login->Cancel ->User ID.	The tab order is Password->Login->Cancel->User ID.
5	Global Interface	Validate the Hot keys functionalities by clicking on the required hot keys	Login : Alt + O Cancel: Esc	Login : Alt + O Cancel: Esc

SNo	Module	Test Step Description	Expected Result	Actual Result
6	Global Interface	Enter a valid user Id and password and click on 'Login' button.	System should launch 'Global Interface UI'.'Connection status' and 'View Log' windows should be displayed by default.	'Global Interface UI'. VP2: 'Connection status' and 'View Log' windows displayed by default.
7	Global Interface	Enter an invalid user Id and password.	System should display a message 'Invalid User ID/ Password' with the 'Ok' button.	System displayed a message 'Invalid User ID/ Password' with the 'Ok' button.
8	Global Interface	Click on 'OK' button.	System should clear the data and return the focus to user id field.	System should clear the data and return the focus to user id field.
9	Global Interface	Click on 'Ctrl+I'.	System should display a modal popup screen 'Select Parser' screen with the 'Parser Drop Down' which contains all the available Parsers.	System has displayed a modal popup screen 'Select Parser' screen with the 'Parser Drop Down' which contains all the available Parsers.
10	Global Interface	Change the format, length of some fields and confirm the change by clicking on 'Apply' button.	System should save the change and should refresh the screen.	System has saved the change and refreshed the screen.

Table 5.6 Sample Test Cases

CHAPTER 6

CONCLUTION AND FUTURE ENHANSEMENT

6.1 CONCLUTION

The “Automation of Pharmacy System” can be developed and operated under optimal expenses with currently available Hardware and Software technologies. The resources needed to run the system is highly reliable. The implementation & operation of the system can be carried out with minimal expense & expertise. This System has the potential of serving as a powerful tool in the speedier and more precise functioning. The specific benefits of the system maybe enumerated as follows...

- ✓ Clears Database record on long run of application
- ✓ Ensures security using 128 bit encryption technique.
- ✓ Fast, Accurate, Efficient and Time saving.
- ✓ Data Management is done efficiently & effectively.
- ✓ Minimum risk of data loss.
- ✓ Higher data security and validity.
- ✓ Simple and intuitive GUI for performing all functions.

The system meets the specific objectives. The Database management tools comprising Sort, Filter & Search functions allow for easier storage and management of large volumes of data. These functions also make the system capable of processing large amounts of data quickly and efficiently. On long run of application their may be large volumes of data which may be no longer used frequently.

6.2 FUTURE ENHANCEMENT

The following features listed below if introduced could benefit the users of the system.

- **e-Prescribing**

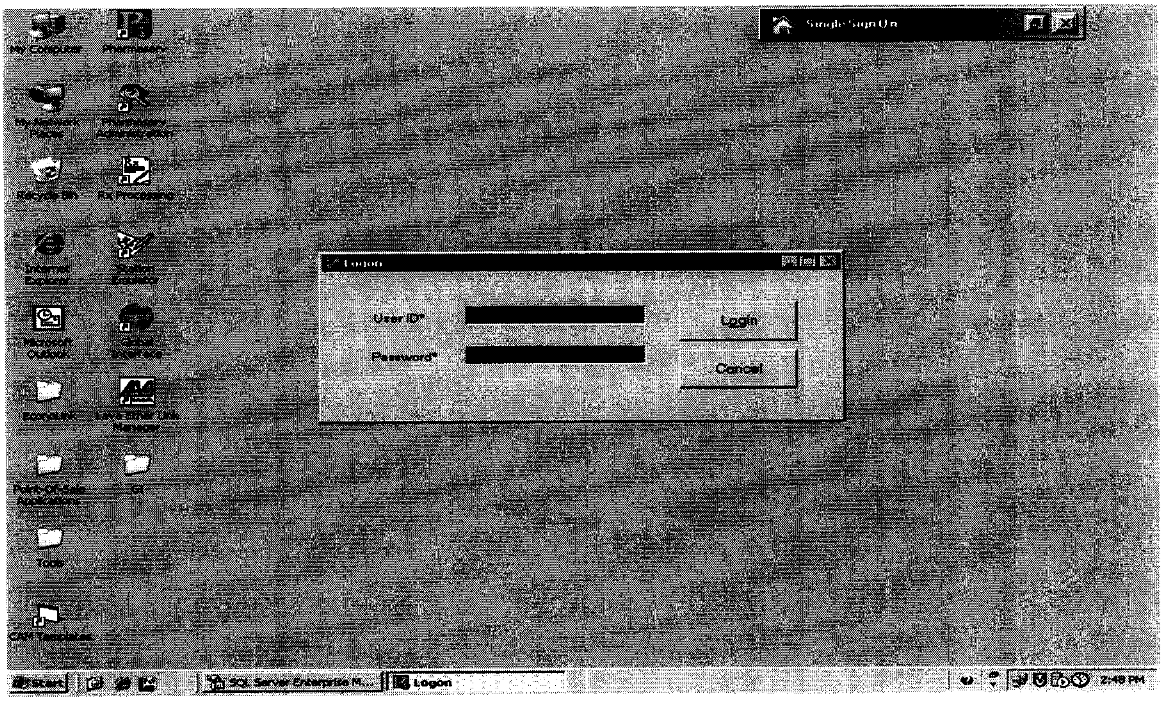
  The Electronic Prescribing functionality is to allow a user to transmit and receive electronic prescription messages. This electronic prescription process will have the benefits of increasing productivity of pharmacy staff since they will not be waiting on hold, decreasing the number of prescription errors due to transcription errors, better knowledge of where a refill is in the renewal process through tracking of process and ability to optimize workflow management for new prescription fulfillment.

- **Audit Data**

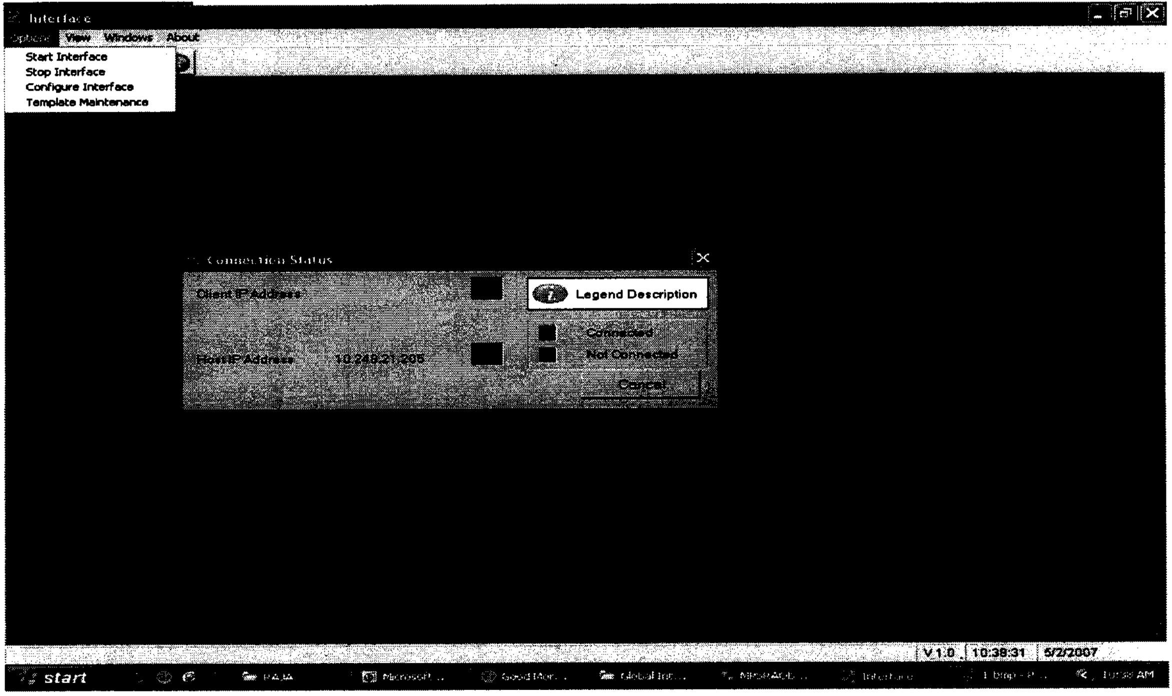
  The system shall record and make available certain events. Each event shall include the information such as User Name, Date/Time of event, Event Type and Event Description

APPENDIX

Login Screen



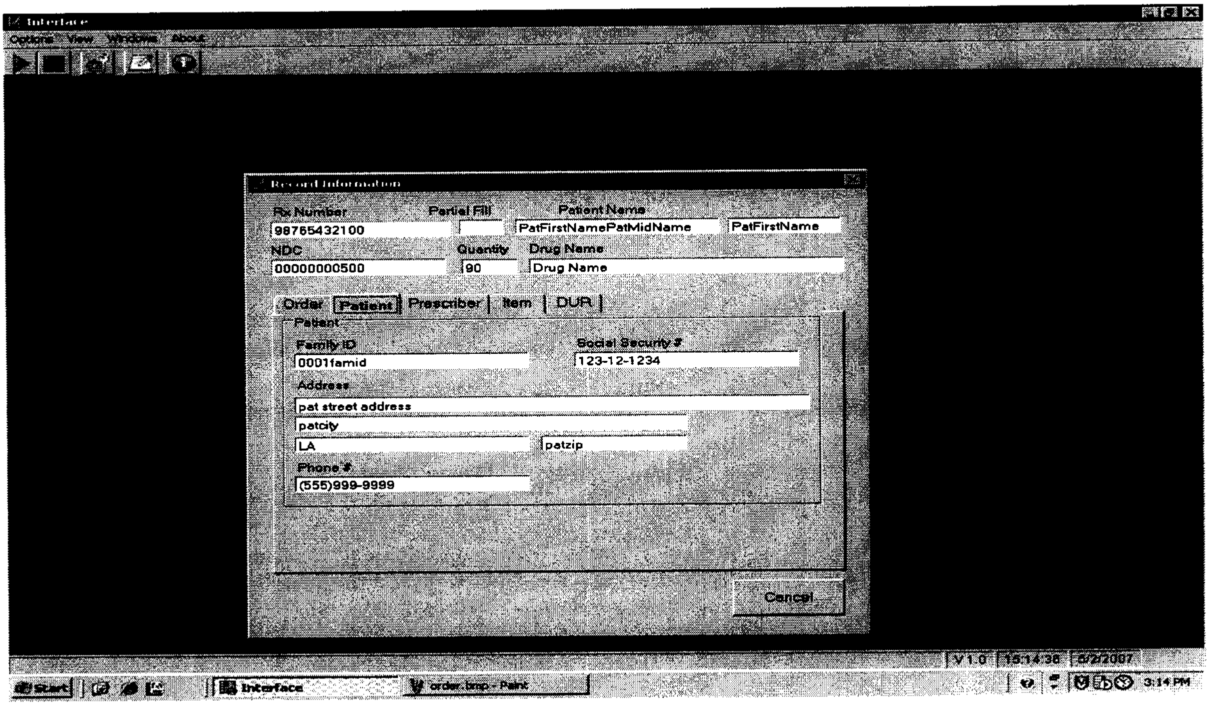
Main Screen



Host IP Address 10.248.21.205

V 1.0 | 10:38:31 | 5/2/2007

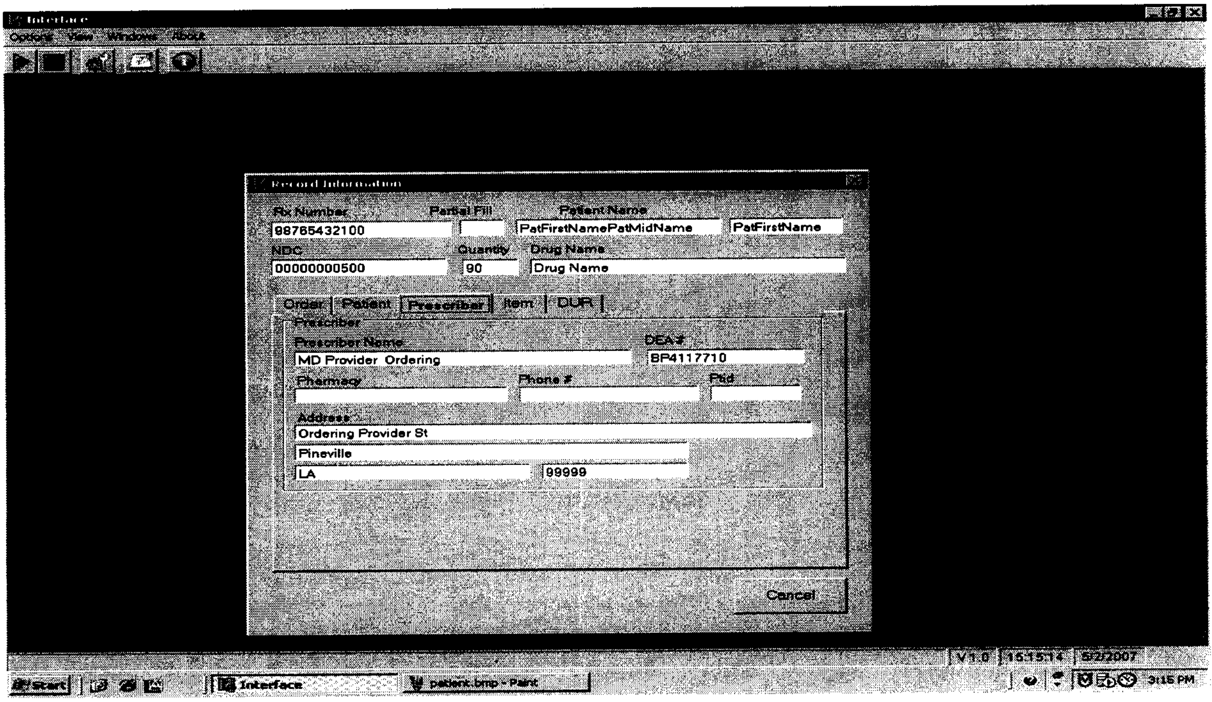
Record Information Screen- Patient Tab



Rx Number: 98765432100
Partial Fill:
Patient Name: PatFirstNamePatMidName PatFirstName
NDC: 00000000500
Quantity: 90
Drug Name: Drug Name

Patient
Family ID: 0001famid
Social Security #: 123-12-1234
Address:
pat street address
patcity
LA patzip
Phone #: (555)999-9999

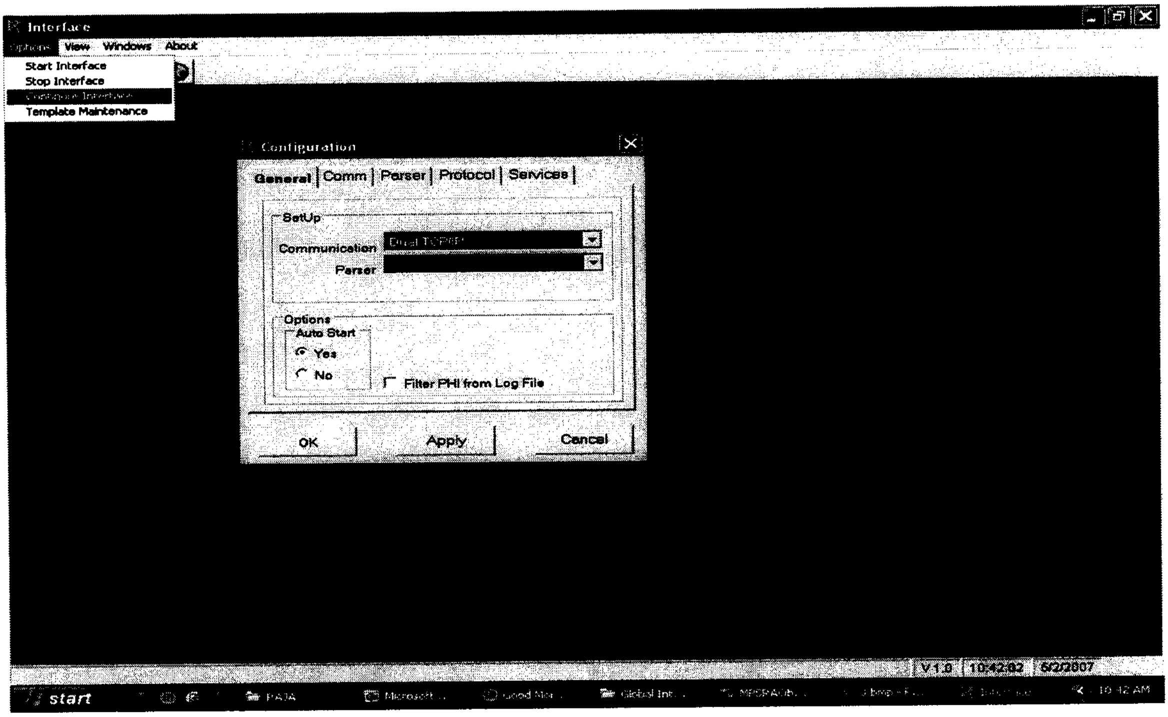
Record Information Screen -Prescriber Tab



Rx Number: 98765432100
Partial Fill:
Patient Name: PatFirstNamePatMidName PatFirstName
NDC: 00000000500
Quantity: 90
Drug Name: Drug Name

Prescriber
Prescriber Name: MD Provider Ordering
DEA #: BP4117710
Pharmacy:
Phone #:
Pid:
Address:
Ordering Provider St
Pineville
LA 99999

Configuration Screen



SetUp

Communication: Dual TCP/IP

Parser:

Options

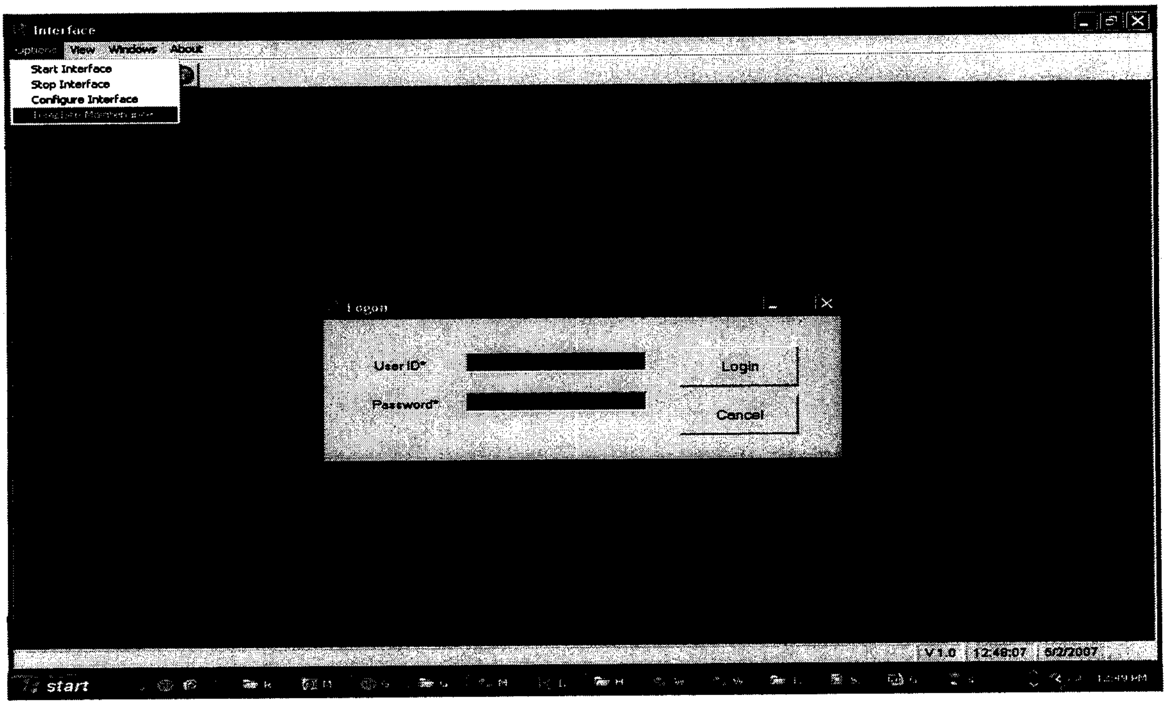
Auto Start

● Yes

○ No

□ Filter PHI from Log File

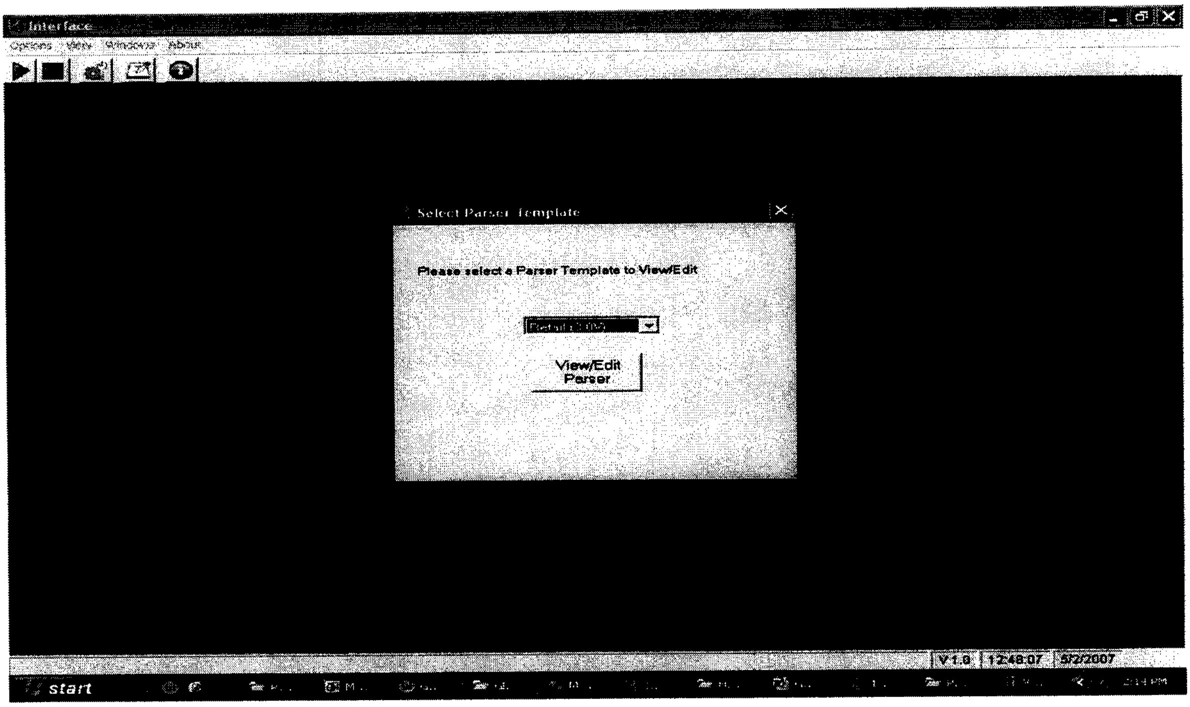
Template Maintenance Logon Screen



User ID*

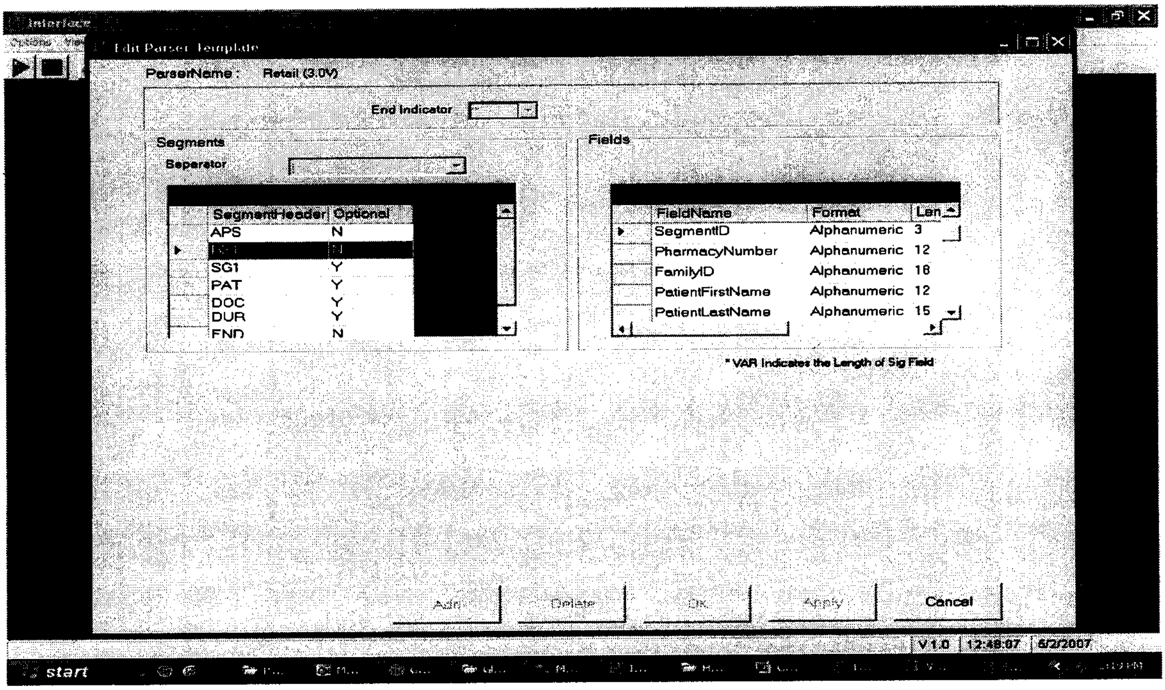
Password*

Select Parser Template Screen



Please select a Parser Template to View/Edit

Edit Parser Template Screen-1



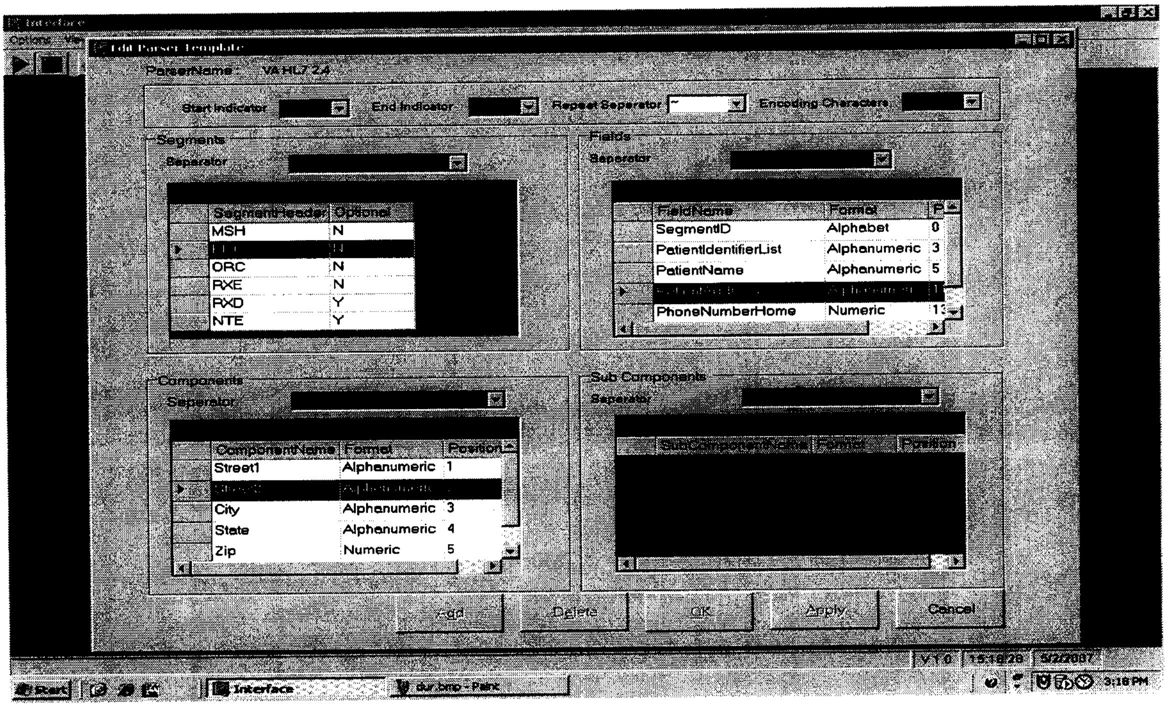
ParserName : Retail (3.0V)

SegmentHeader	Optional
APS	N
[illegible]	N
SG1	Y
PAT	Y
DOC	Y
DUR	Y
FND	N

FieldName	Format	Len
SegmentID	Alphanumeric	3
PharmacyNumber	Alphanumeric	12
FamilyID	Alphanumeric	18
PatientFirstName	Alphanumeric	12
PatientLastName	Alphanumeric	15

* VAR Indicates the Length of Sig Field

Edit Parser Template Screen-2



ParserName : VA HL7 2.4

Start Indicator

End Indicator

Repeat Seperator ~

Encoding Characters

Segments

Seperator

SegmentHeader	Optional
MSH	N
PID	N
ORC	N
RXE	N
RXD	Y
NTE	Y

Fields

Seperator

FieldName	Format	P
SegmentID	Alphabet	0
PatientIdentifierList	Alphanumeric	3
PatientName	Alphanumeric	5
[illegible]	[illegible]	1
PhoneNumberHome	Numeric	13

Components

Seperator

ComponentName	Format	Position
Street1	Alphanumeric	1
Street2	Alphanumeric	2
City	Alphanumeric	3
State	Alphanumeric	4
Zip	Numeric	5

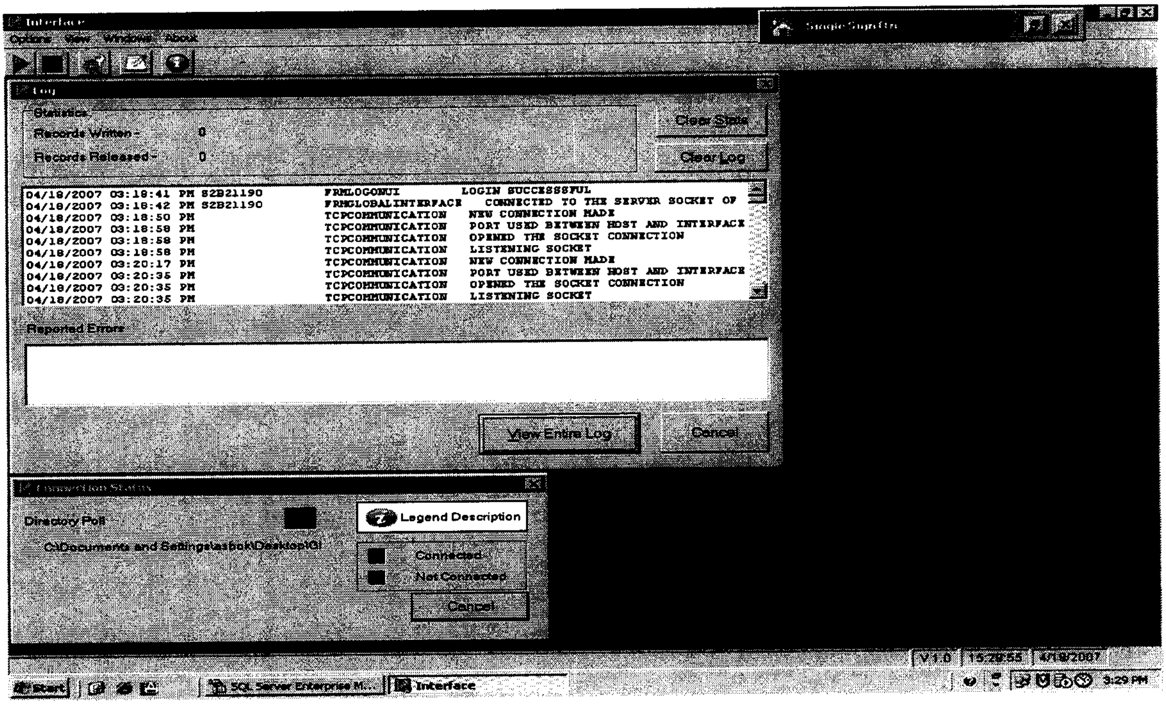
Sub Components

Seperator

SubComponentName	Format	Position

V1.0 | 15:18:28 | 5/2/2007

Log Screen



Statistics

Records Written - 0

Records Released - 0

```
04/18/2007 03:18:41 PM S2B21190        FRMLOGONUI         LOGIN SUCCESSSFUL
04/18/2007 03:18:42 PM S2B21190        FRMGLOBALINTERFACE   CONNECTED TO THE SERVER SOCKET OF
04/18/2007 03:18:50 PM                 TCPCOMMUNICATION   NEW CONNECTION MADE
04/18/2007 03:18:58 PM                 TCPCOMMUNICATION   PORT USED BETWEEN HOST AND INTERFACE
04/18/2007 03:18:58 PM                 TCPCOMMUNICATION   OPENED THE SOCKET CONNECTION
04/18/2007 03:18:58 PM                 TCPCOMMUNICATION   LISTENING SOCKET
04/18/2007 03:20:17 PM                 TCPCOMMUNICATION   NEW CONNECTION MADE
04/18/2007 03:20:35 PM                 TCPCOMMUNICATION   PORT USED BETWEEN HOST AND INTERFACE
04/18/2007 03:20:35 PM                 TCPCOMMUNICATION   OPENED THE SOCKET CONNECTION
04/18/2007 03:20:35 PM                 TCPCOMMUNICATION   LISTENING SOCKET
```

Reported Errors

Directory Poll ■

C:\Documents and Settings\asbok\Desktop\GI

■ Connected

■ Not Connected

V1.0 | 15:29:55 | 4/19/2007

BIBLIOGRAPHY

Reference Books:

- Visual Basic .NET Programming Bible

  -Bill Evjen.

- Professional VB.NET 2003 ,4th Edition

  - Wrox Series Publications

- Software Engineering

  -Ian Sommerville

Reference Websites:

- www.microsoft.com/msdn
- www.coduguru.com