

# **CareVoyant Patient Portal**

A Manuscript

Submitted to

the Department of Computer Science

and the Faculty of the

University of Wisconsin-La Crosse

La Crosse, Wisconsin

by

**Vinoth Perkinian**

in Partial Fulfillment of the

Requirements for the Degree of

**Master of Software Engineering**

April, 2010

# CareVoyant Patient Portal

By Vinoth Perkinian

We recommend acceptance of this manuscript in partial fulfillment of this candidate's requirements for the degree of Master of Software Engineering in Computer Science. The candidate has completed the oral examination requirement of the capstone project for the degree.

---

Dr. Kasi Periyasamy  
Examination Committee Chairperson

---

Date

---

Dr. Dave Riley  
Examination Committee Member

---

Date

---

Dr. Mark Headington  
Examination Committee Member

---

Date

## **ABSTRACT**

PERKINIAN, VINOTH, S., “CareVoyant Patient Portal”, Master of Software Engineering, May 2010, (Dr. Kasi Periyasamy).

Information technology (IT) has the potential to improve the quality, safety, and efficiency of healthcare applications by providing new methods and techniques to healthcare providers and patients to readily access and use health information. Both private and public sectors have engaged in numerous efforts to promote the use of IT within healthcare institutions and across all areas where healthcare is provided. Some of these efforts include developing and promoting industry-wide standards, funding research to investigate the impact of IT on quality, and providing incentives that encourage investment in IT. CareVoyant is a software company that develops software solutions for healthcare applications. The purpose of this project is to develop a web-based application called CareVoyant Patient Portal that will work seamlessly with CareVoyant integrated application suite. The portal was developed to provide exposure to patient related healthcare information through web to patients and responsible parties. Through the patient portal, enterprises that use CareVoyant integrated application suite will be able to authorize their patients and responsible parties (guarantors) to gain access to their information such as demographic, financial, and clinical information. In addition, the user interface for this project has been developed in such a way that the application can be dynamically configured or modified to add or remove functionalities without modifying the code or redeploying the application. This feature introduced a new concept of advanced user interface design which also added a marketing value to the product.

## **ACKNOWLEDGEMENTS**

I would like to express my sincere thanks to my project advisor Dr. Kasi Periyasamy for his valuable guidance. I thank him for his time and expertise. I also would like to thank the project sponsor, CareVoyant Inc, Pasupathy Kandasamy (Pasu) and Senthil Muthukumar, who initiated this project and provided the support for this project. I need to express my special thanks to my co-workers and fellow IT personnel for their valuable input and advice. I would also like to thank to the Computer Science Department, Murphy Library and the University of Wisconsin-La Crosse for providing me with the computing environment for my project. In addition, I wish to thank my parents and my brother for their patience and encouragement during the tenure of this project. I truly appreciate the sacrifices they made to enable me to complete this degree. Finally, I want to thank all my friends and family for their support and encouragement during the duration of this project and the MSE program. I couldn't have done it without you.

# TABLE OF CONTENTS

|  | <u>Page</u> |
|--|-------------|
| ABSTRACT.....  | iii         |
| ACKNOWLEDGEMENTS.....  | iv          |
| TABLE OF CONTENTS.....   | v           |
| LIST OF TABLES.....  | vi          |
| LIST OF FIGURES.....   | vii         |
| GLOSSARY.....  | viii        |
| 1. Background Information.....                                   | 1           |
| 2. Selecting an Appropriate Software Life Cycle Model.....       | 4           |
| 3. Development of the CareVoyant Patient Portal System.....      | 13          |
| 3.1 Understanding the CareVoyant Main Application Suite.....     | 13          |
| 3.2 User Interface for CareVoyant Patient Portal.....            | 16          |
| 3.3 Technology for CareVoyant Patient Portal.....                | 17          |
| 4. The CareVoyant Patient Portal Application.....                | 18          |
| 4.1. High Level Architectural Design.....                        | 18          |
| 4.2. Detailed Architecture of CareVoyant Patient Portal.....     | 19          |
| 4.3. Database Design.....  | 24          |
| 4.4. User Interface Design.....                                  | 26          |
| 4.5. Security Design.....  | 29          |
| 4.6. Deploying CareVoyant Patient Portal.....                    | 29          |
| 4.7. Testing.....  | 30          |
| 5. Limitations.....  | 31          |
| 6. Continuing Work.....  | 32          |
| 7. Conclusion.....   | 33          |
| BIBLIOGRAPHY.....  | 34          |
| APPENDIX A: Selected CareVoyant Patient Portal Screen Shots..... | 35          |

## LIST OF TABLES

| <u>Table</u>  | <u>Page</u> |
|---|-------------|
| Selecting a Life Cycle Model Based on Characteristic of Requirements.....           | 6           |
| Selecting a Life Cycle Model Based on Characteristic of Project Team.....           | 7           |
| Selecting a Life Cycle Model Based on Characteristic of User Community .....        | 9           |
| Selecting a Life Cycle Model Based on Characteristic of Project Type and Risk ..... | 10          |
| Summary of Total Responses .....  | 12          |
| Core Functionalities .....  | 14          |
| Third Party Functionalities .....   | 15          |
| CareVoyant Databases .....  | 15          |
| Types of Control / Component .....  | 27          |

## LIST OF FIGURES

| <u>Figure</u>  | <u>Page</u> |
|--|-------------|
| Architecture of CareVoyant Main Application Suite .....  | 17          |
| High-Level Architecture of CareVoyant Patient Portal.....  | 18          |
| Detailed Architecture of CareVoyant Patient Portal.....  | 21          |
| Accessible Functionalities of CareVoyant Patient Portal by Administrative user .....                 | 22          |
| Accessible functionalities of CareVoyant Patient Portal by Patient and Guarantor users .....         | 23          |
| Database Diagram of CareVoyant Patient Portal for newly added Company Tables (Company Database)..... | 25          |
| Database Diagram of CareVoyant Patient Portal for newly added System Tables (System Database) .....  | 25          |
| Login Screen .....   | 35          |
| Guarantor Homepage .....   | 36          |
| My Patients Control for Guarantor .....  | 37          |
| Demographic Information Control .....  | 38          |
| Guarantor Information Control .....  | 39          |
| Account Statement Control.....   | 40          |
| Patient Insurance Plan Control.....  | 41          |
| Demographic Information change request Control.....  | 42          |
| Vitals Control.....  | 43          |
| Medication Control with built-in Drug Interaction and Drug Education .....                           | 44          |
| Account Statement Viewer .....   | 45          |
| User Approval Screen .....   | 46          |
| Custom Control Management Screen .....   | 47          |
| Built-in Custom Control Management Screen.....   | 48          |
| Homepage Control Management Screen for Patient / Guarantor .....                                     | 49          |

## **GLOSSARY**

### **ASP.Net**

The latest technology from Microsoft for creating web applications based on the .NET framework. It is a framework for building web applications built on top of the Microsoft Common Language Runtime (CLR).

### **C#**

An Object Oriented programming language and associated development environment from Microsoft for its COM programming model and designed for the Common Language Infrastructure. [10]

### **Data Integrity**

A process that ensures that the particular data is complete. It refers to the condition in which data is identically maintained during any operation, such as transfer, storage, and retrieval. It also includes the preservation of data for its intended use.

### **Durable Medical Equipment**

Durable Medical Equipment (DME) is a term of art used to describe any medical equipment used at home to aid in a better quality of living.

### **Dynamic link libraries (DLL)**

A Microsoft's implementation of the shared library concept in the Microsoft Windows and OS/2 operating systems. These libraries usually have the file extension DLL.



## **Executable**

An executable or executable file is a file whose contents are meant to be interpreted as a program by a computer.

## **Health Insurance Portability and Accountability Act**

The Administrative Simplification provisions of the Health Insurance Portability and Accountability Act of 1996 (HIPAA, Title II) required the Department of Health and Human Services (HHS) to establish national standards for electronic healthcare transactions and national identifiers for providers, health plans, and employers.[8]

## **Integrated Development Environment**

An Integrated Development Environment (IDE) also known as integrated design environment or integrated debugging environment is a software application that provides comprehensive facilities to computer programmers for software development.

## **JavaScript**

JavaScript is an object-oriented scripting language used to enable programmatic access to objects within both the client application and other applications. It is primarily used on the client-side, and is implemented as an integrated component of the web browser, allowing the development of enhanced user interfaces and dynamic websites.

## **Microsoft SQL Server 2005**

A relational database management system (RDBMS) produced by Microsoft. Its primary query language is Transact-SQL, an implementation of the ANSI/ISO standard Structured Query Language (SQL) used by both Microsoft and Sybase.

### **Telerik Radcontrols**

Telerik Radcontrols for ASP.NET AJAX includes more than 60 controls with proven reliability that helps building high-quality, professional line of business web applications, created by Telerik Corporation. [6]

### **UML Class Diagram**

In the Unified Modeling Language (UML), the class diagram describes the conceptual model of classes that illustrates the interfaces, their associations and the relationship between classes. [1]

### **Use Case**

Use Case describes the visible requirement of the system. They are used in requirement analysis phase and contribute to test plans and user guides. [1]

### **Visual Studio .NET**

A Microsoft GUI development environment for building various software applications.

## **1. Background Information**

Healthcare industry is one of the largest industries in 2008, contributing 14.5% of the nation's economy, providing 14.3 million jobs for wage and salary workers in the United States. Healthcare related occupation being ranked as ten of the 20 fastest growing occupation in the nation, which is predicted to generate 3.2 million new wage and salary jobs between 2008- 2018, more than any other industry, largely in response to rapid growth in the elderly population.

Therefore, in the rapidly changing healthcare industry, technological advances have made many significant impacts. Some of them are electronic medical records (EMR), software products for image processing and distribution, and maintenance of large volume of health insurance records. This technological advancement has resulted in providing comprehensive and perceived improvement on patient care and worker efficiency.

Cost effectiveness is also improved with the increased use of integrated software systems. Such systems combine two or more segments of the healthcare industry to increase efficiency through the streamlining of functions, primarily financial, clinical and managerial. These changes will continue to reshape not only the nature of healthcare workforce, but also the manner in which healthcare is provided.

Devices such as hand-held computers are used to record a patient's medical history. Information on vital signs and orders for tests are transferred electronically to databases. This process eliminates the need for paper and reduces recordkeeping errors. These new technologies will continue to enable earlier diagnoses of many diseases which often increase the ability to treat conditions that were previously not treatable.

Industry growth will occur as a result of the shift from inpatient to less expensive outpatient and home healthcare because of improvements in diagnostic tests and surgical procedures, along with patients' desire to be treated at home. However, since the adoption of electronic health record is relatively low, the Federal government provides incentives through 2009 American Recovery and Reinvestment Act (ARRA) for those qualifying Care Delivery Organizations (CDO) who implement and use certified electronic medical records within the 2010-2011 time frame [11].

Due to this rapid advancement, growth and complexity of the healthcare industry, there is an evident need for software industry to offer continuous solutions through innovative, integrated and intuitive software suites.

CareVoyant, a healthcare Software company and the sponsor of this project, introduced its first physician billing solution in early 1990s. As a result of the demanding need for software solutions in healthcare industry, CareVoyant have continuously achieved in building software solutions that extensively consolidates clinical, financial, point of care and business intelligence functions of the healthcare domain, in a single software suite by using the latest Microsoft technologies.

CareVoyant's integrated application suite is categorized into (1) ambulatory care environment, comprising of physicians, rehab/therapy, Durable Medical Equipment (DME), and rural health markets, and (2) post-acute continuum, comprising of nursing homes, community care, home healthcare, and hospice markets. CareVoyant's integrated application suite is segmented into several modules such as scheduling, billing, claims / statements, clinical integration, functional modules, business integration, dashboards, portals and analyzers [9].

The CareVoyant Patient Portal is a new module that will be added to the list of existing portals that are available in the CareVoyant's integrated application suite. This new module will allow CareVoyant's industrial clients, also called "Enterprises" in this manuscript, to provide authorized access for patients and responsible parties called "Guarantors" in this manuscript, to gain access to their information's such as demographic, financial, clinical, and their other related information. This report describes the development process used to develop the Patient Portal application and integrate it with the CareVoyant integrated application suite.

The developer had previous experience in developing and maintaining a software system called *Patient Scheduling System* by using Microsoft's .NET platform. This system is used to perform advanced patient scheduling functions and to maintain patient records and appointment histories for the Student Health Center, at the University of Wisconsin – La Crosse. The developer's experience in the scheduling system strongly motivated the sponsor to offer CareVoyant Patient Portal as a capstone project to the developer.

Since the project sponsor is a Microsoft Gold Certified Partner and an early adopter of the latest Microsoft technologies, the project sponsor requested an extensive incorporation of Microsoft's .NET platform. Therefore the project was developed in Microsoft .NET platform with SQL Server 2005 as the back-end.

## **2. Selecting an Appropriate Software Life Cycle Model**

A software life cycle model depicts the significant phases or activities of a software project from conception until the product is retired [12]. A typical software life cycle model encompasses several phases such as requirements, design, implementation, integration, testing, and maintenance.

Among the software life cycle models used in practice, the "waterfall model" was probably the first published and the least flexible. It follows a sequential software development process in which development is seen as flowing steadily downwards (like a waterfall) through the phases of requirements analysis, design, implementation, testing, integration, and maintenance. The major disadvantage of the waterfall model is that it does not work for certain problem domains, notably for those where the requirements are not well understood in advance and are likely to change significantly over the course of development.

The more recent and popular software life cycle model is the agile software development model in which the software development processes are built on the foundation of iterative development. To that foundation the software development process adds a lighter, more people-centric viewpoint than traditional approaches. Agile processes use feedback, rather than planning, as their primary control mechanism. The feedback is driven by regular tests and releases of the evolving software. This is the most popular and evolving development model in the current IT industry.

An in-depth discussion of the software life cycle models are beyond the scope of this report, but additional information can be found in many texts on Software Engineering practices. [3] [5]

The selection of a software life cycle model is the process of evaluating the specific needs and challenges of a project and then choosing the most appropriate model for the development process [2]. Life cycle model selection helps ensure the development is the most efficient and effective. If a wrong life cycle model is used, it can result in missing tasks and inappropriate task ordering which undercuts project planning and efficiency.

To choose the most effective life cycle model for this project, the characteristics of this project were examined. These include the requirements of the project, the project team, user community, the project type and risks involved by following the approach verbatim as given in the book [4] and the process executed is explained below.

- 1) The project was examined based on the following project characteristic categories, as demonstrated in Tables 1 through 4
  - a. Requirements : Table 1
  - b. Project team : Table 2
  - c. User community : Table 3
  - d. Project type and risk : Table 4
- 2) The questions presented for each category is answered by circling a yes or no in the matrices provided.
- 3) The importance of the category, or question within the category, is ranked in terms of this project.
- 4) The appropriate model is selected based on totaling the number of circled responses for each column in the matrices.
- 5) The category ranking was used to resolve the conflicts between the models if the totals are close or the same.

### **Requirements**

The requirement category (Table 1) consisted of questions related to things that have been requested by the sponsor for this project. They are often termed as functions or features of the system that will be provided in the project.

| <i>Requirements</i>  | <i>Waterfall</i>         | <i>V-Shaped</i>          | <i>Prototype</i>          | <i>Spiral</i>             | <i>RAD</i>                | <i>Incremental</i>        |
|--|--------------------------|--------------------------|---------------------------|---------------------------|---------------------------|---------------------------|
| Are the requirements easily defined and/or well known?                 | Yes                      | Yes                      | <input type="radio"/> No  | <input type="radio"/> No  | Yes                       | <input type="radio"/> No  |
| Can the requirements be defined early in the cycle?                    | Yes                      | Yes                      | <input type="radio"/> No  | <input type="radio"/> No  | Yes                       | Yes                       |
| Will the requirements change often in the cycle?                       | <input type="radio"/> No | <input type="radio"/> No | Yes                       | Yes                       | <input type="radio"/> No  | <input type="radio"/> No  |
| Is there a need to demonstrate the requirements to achieve definition? | <input type="radio"/> No | <input type="radio"/> No | Yes                       | Yes                       | Yes                       | <input type="radio"/> No  |
| Is a proof concept required to demonstrate capability?                 | <input type="radio"/> No | <input type="radio"/> No | Yes                       | Yes                       | Yes                       | <input type="radio"/> No  |
| Do the requirements indicate a complex system?                         | No                       | No                       | <input type="radio"/> Yes | <input type="radio"/> Yes | No                        | <input type="radio"/> Yes |
| Is early functionality a requirement?                                  | No                       | No                       | <input type="radio"/> Yes | <input type="radio"/> Yes | <input type="radio"/> Yes | <input type="radio"/> Yes |
| <b>Total</b>   | <b>3</b>                 | <b>3</b>                 | <b>4</b>                  | <b>4</b>                  | <b>2</b>                  | <b>6</b>                  |

Table 1. Selecting a Life Cycle Model Based on Characteristic of Requirements



## Project Team

The questions in Table 2 are related to project management and resource allocation that have direct impact on this project.

| <i>Project Team</i>  | <i>Waterfall</i>          | <i>V-Shaped</i>           | <i>Prototype</i>          | <i>Spiral</i>             | <i>RAD</i>                | <i>Incremental</i>        |
|--|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|
| Are the majority of team members new to the problem domain for the project?    | <input type="radio"/> No  | <input type="radio"/> No  | <input type="radio"/> Yes | <input type="radio"/> Yes | <input type="radio"/> No  | <input type="radio"/> No  |
| Are the majority of team members new to the technology domain for the project? | <input type="radio"/> Yes | <input type="radio"/> Yes | <input type="radio"/> No  | <input type="radio"/> Yes | <input type="radio"/> No  | <input type="radio"/> Yes |
| Are the majority of team members new to the tools to be used on the project?   | <input type="radio"/> Yes | <input type="radio"/> Yes | <input type="radio"/> No  | <input type="radio"/> Yes | <input type="radio"/> No  | <input type="radio"/> No  |
| Are the team members subject to reassignment during the life cycle?            | <input type="radio"/> No  | <input type="radio"/> No  | <input type="radio"/> Yes | <input type="radio"/> Yes | <input type="radio"/> No  | <input type="radio"/> Yes |
| Is there training available for the project team, if required?                 | <input type="radio"/> No  | <input type="radio"/> Yes | <input type="radio"/> No  | <input type="radio"/> No  | <input type="radio"/> Yes | <input type="radio"/> Yes |

Table 2. Selecting a Life Cycle Model Based on Characteristic of Project Team

| <i>Project Team</i>   | <i>Waterfall</i> | <i>V-Shaped</i> | <i>Prototype</i> | <i>Spiral</i> | <i>RAD</i> | <i>Incremental</i> |
|---|------------------|-----------------|------------------|---------------|------------|--------------------|
| Is the team more comfortable with structure than flexibility?                                     | Yes              | Yes             | No               | No            | No         | Yes                |
| Will the project manager closely track the team's progress?                                       | Yes              | Yes             | No               | Yes           | No         | Yes                |
| Is ease of resource allocation important?   | Yes              | Yes             | No               | No            | Yes        | Yes                |
| Does the team accept peer reviews and inspections, management / customer reviews, and milestones? | Yes              | Yes             | Yes              | Yes           | No         | Yes                |
| <b>Total</b>  | <b>6</b>         | <b>7</b>        | <b>3</b>         | <b>4</b>      | <b>4</b>   | <b>9</b>           |

Table 2. (Continued)

## User Community

The early project phases such as requirement gathering, end user analysis, etc can provide a good understanding of the user community (Table 3) and its expected relationship with the project.

| <i>User Community</i>   | <i>Waterfall</i> | <i>V-Shaped</i> | <i>Prototype</i> | <i>Spiral</i> | <i>RAD</i> | <i>Incremental</i> |
|---|------------------|-----------------|------------------|---------------|------------|--------------------|
| Will the availability of the user representatives be restricted or limited during the life cycle? | Yes              | Yes             | No               | Yes           | No         | Yes                |
| Are the user representatives new to the system definition?  | No               | No              | Yes              | Yes           | No         | Yes                |
| Are the user representatives' experts in the problem domain?                                      | No               | No              | Yes              | No            | Yes        | Yes                |
| Do the users want to be involved in all phases of the life cycle?                                 | No               | No              | Yes              | No            | Yes        | No                 |
| Does the customer want to track project progress?   | No               | No              | Yes              | Yes           | No         | No                 |
| <b>Total</b>  | <b>3</b>         | <b>3</b>        | <b>2</b>         | <b>3</b>      | <b>2</b>   | <b>5</b>           |

Table 3. Selecting a Life Cycle Model Based on Characteristic of User Community

### Project Type and Risk

Finally, the type of project and the risks (Table 4) that have been identified up to this point in the planning phase have been examined. The selection of a model that accommodates risk management does not mean that the creation of an action plan to minimize the risk identified is not required. The model simply provides a framework within which this action plan can be discussed and executed.

| <i>Project Type and Risk</i>  | <i>Waterfall</i> | <i>V-Shaped</i> | <i>Prototype</i> | <i>Spiral</i> | <i>RAD</i> | <i>Incremental</i> |
|---|------------------|-----------------|------------------|---------------|------------|--------------------|
| Does the project identify a new product direction for the organization?         | No               | No              | Yes              | Yes           | No         | Yes                |
| Is the project a system integration project?                                    | No               | Yes             | Yes              | Yes           | Yes        | Yes                |
| Is the project an enhancement to an existing system?                            | No               | Yes             | No               | No            | Yes        | Yes                |
| Is the funding for the project expected to be stable throughout the life cycle? | Yes              | Yes             | Yes              | No            | Yes        | No                 |

Table 4. Selecting a Life Cycle Model Based on Characteristic of Project Type and Risk

| <i>Project Type and Risk</i>   | <i>Waterfall</i>                     | <i>V-Shaped</i>                      | <i>Prototype</i>                     | <i>Spiral</i>                        | <i>RAD</i>                           | <i>Incremental</i>                   |
|--|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|--------------------------------------|
| Is the product expected to have a long life in the organization?                         | <input checked="" type="radio"/> Yes | <input checked="" type="radio"/> Yes | No                                   | <input checked="" type="radio"/> Yes | No                                   | <input checked="" type="radio"/> Yes |
| Is high reliability a must?  | No                                   | <input checked="" type="radio"/> Yes | No                                   | <input checked="" type="radio"/> Yes | No                                   | <input checked="" type="radio"/> Yes |
| Is the system expected to be modified, perhaps in ways not anticipated, post deployment? | No                                   | No                                   | <input checked="" type="radio"/> Yes | <input checked="" type="radio"/> Yes | No                                   | <input checked="" type="radio"/> Yes |
| Is the schedule constrained?   | No                                   | No                                   | <input checked="" type="radio"/> Yes | <input checked="" type="radio"/> Yes | <input checked="" type="radio"/> Yes | <input checked="" type="radio"/> Yes |
| Are the modules interfaces clean?  | <input checked="" type="radio"/> Yes | <input checked="" type="radio"/> Yes | No                                   | No                                   | No                                   | <input checked="" type="radio"/> Yes |
| Are reusable components available?   | <input checked="" type="radio"/> No  | <input checked="" type="radio"/> No  | Yes                                  | Yes                                  | Yes                                  | <input checked="" type="radio"/> No  |
| Are resources (time, money, tools, and people) scarce?                                   | No                                   | No                                   | <input checked="" type="radio"/> Yes | <input checked="" type="radio"/> Yes | No                                   | No                                   |
| <b>Total</b>   | <b>3</b>                             | <b>6</b>                             | <b>5</b>                             | <b>8</b>                             | <b>3</b>                             | <b>10</b>                            |

Table 4. (Continued)

| <i>Characteristic</i> | <i>Waterfall</i> | <i>V-Shaped</i> | <i>Prototype</i> | <i>Spiral</i> | <i>RAD</i> | <i>Incremental</i> |
|-----------------------|------------------|-----------------|------------------|---------------|------------|--------------------|
| Requirements          | 3                | 3               | 4                | 4             | 2          | 6                  |
| Project Team          | 6                | 7               | 3                | 4             | 4          | 9                  |
| User Community        | 3                | 3               | 2                | 3             | 2          | 5                  |
| Project Type and Risk | 3                | 6               | 5                | 8             | 3          | 10                 |
| <b>Total</b>          | <b>15</b>        | <b>19</b>       | <b>14</b>        | <b>19</b>     | <b>11</b>  | <b>30</b>          |

Table 5. Summary of Total Responses

Based on the criteria discussed in Tables 1 through 4 and summary of total responses shown in Table 5, it is evident that the incremental prototype model would be the most effective model based on all the project characteristics that were evaluated. But since the requirements of this project were not well defined or not known in the early stages of the life cycle, it was decided that agile model would be the perfect and most effective model. This is because agile model adds requirement gathering on top of incremental prototyping approach for all the iterations through feedbacks such as team meetings, prototype demos, etc. Therefore CareVoyant Patient Portal was developed as an agile web development project.

### **3. Development of the CareVoyant Patient Portal System**

The development process of CareVoyant Patient Portal involved three major phases or stages. The first phase involved understanding the CareVoyant main application suite, particularly about its architecture and its underlying databases. In the second phase, the basic requirements were gathered and analyzed in order to come up with a focused objective of the project. Continued in this phase was an incremental prototyping approach, where several incremental prototypes were developed in an agile fashion to communicate with the sponsor and discover additional requirements as well as changes to the existing requirements. In the third phase, the product was enhanced to include several additional features, in order to enhance reusability and extensibility of the product.

#### **3.1 Understanding the CareVoyant Main Application Suite**

In order to add a portal to the CareVoyant's application suite, it was necessary to understand the main application suite itself from the perspective of end users, and from the perspective of its architecture along with its database. The project sponsor gave a complete demonstration of the CareVoyant main application suite and some of its integrated portals such as Physician Portal. In addition, the developer had a month-long training to understand the architecture of the database and the relationship between the databases and third party services. Later, the source code and the data sources of the CareVoyant main application suite were received from the sponsor and studied. After learning completely about the product and its underlying database thoroughly, the implementation details and the source code of certain functionalities that can be reused directly for this project were clearly identified. They were grouped into two major categories: (1) Core Functionalities (2) functionalities provided by third party. These functionalities are described in Table 6 and Table 7.

In addition, the underlying databases of CareVoyant main application suite that will also be used by this project are grouped into three major categories: (1) System Database (2) Company Database (3) Third Party Database; these are described in Table 8.

|   | <b>Module Name</b>           | <b>Functionalities</b>  |
|---|------------------------------|---|
| 1 | Demographic Module           | View or request changes of demographic information such as patient's address, contact numbers, etc.   |
| 2 | Guarantor Module             | View guarantor information such as guarantor's name, address, and contact numbers for a given patient   |
| 3 | Professional contacts Module | View professional contact information such as provider name, provider location, etc. for a given patient  |
| 4 | Plan Module                  | View or add insurance information such as insurance policy number, name, company details etc. for a given patient   |
| 5 | Appointment Module           | View an existing appointment for a given patient or request for a new appointment for a given patient   |
| 6 | Vitals Module                | View vital information such as weight, blood pressure, temperature, etc. for a given patient  |
| 7 | Medication Module            | View medication information such as name of the medication, frequency, method of intake, etc. for a given patient based on the status of the medication such as active, inactive or suspended |
| 8 | Account Statement Module     | View account statement information such as account balance, patient due, transactions, etc. for a given patient   |

Table 6. Core Functionalities



|   | <b>Module Name</b>      | <b>Functionalities</b>  |
|---|-------------------------|---|
| 1 | Drug Education Module   | View education information for a given drug such as chemical names, brand names, side effects, etc. |
| 2 | Drug Interaction Module | View drug interaction information between two or more drugs.  |

Table 7. Third Party Functionalities

|   | <b>Database Name</b> | <b>Purpose</b>  |
|---|----------------------|---|
| 1 | System Database      | This database contains information that is pertaining to a particular enterprise such as application settings, company information, etc.                                    |
| 2 | Company Database     | This database contains information that is pertaining to all companies that belong to a particular enterprise such as patient information, electronic medical records, etc. |
| 3 | Third party Database | This database contains information such as drug education, drug interaction as provided by third parties.   |

Table 8. CareVoyant Databases

During this process, the first step involved gathering the basic set of requirements and project objective from the sponsor. The developer had several meetings with the development and sales teams of the sponsor's company. It became clear after these meetings that it would add great value to both the sponsor and the developer, if an initial prototype for the subset of functionalities with some user interface screen shots were developed. Therefore this became the initial throwaway prototype and it helped in analyzing and refining the initial set of requirements and evaluating the feasibility of the product. This stage served as a great leap to start the incremental prototype for this project. Upon the sponsor's request, C# [10] was chosen as the programming language and the IDE used was Microsoft Visual Studio 2008 .NET.

The sponsor also specified Microsoft SQL 2005 as the database tool and provided the CareVoyant proprietary dynamic link libraries (DLL) to communicate with CareVoyant databases in order to maintain the same CareVoyant architecture for all its products.

### **3.2 User Interface for CareVoyant Patient Portal**

It was decided to design the best and advanced user interface concepts that are available in the market to support user friendliness and ease-of-use characteristics of the product. This eventually will lead a way to build a completely customizable user interface and help in making this product more flexible to future changes in requirement of the product and will also enhance reusability. Some of the key features that were aimed to be implemented in this phase included the following: (1) Personalization of the product for every user should be possible. (2) All the core functionalities in this project are expected to be pluggable components. (3) It was expected that all new functionalities can be built as a component using the product itself and can be plugged into the product without changing or redeploying the existing product. (4) Functionalities of the product should be customizable based on the demand or user roles. (5) End user can add or remove functionalities based on his/her demands

### 3.3 Technology for CareVoyant Patient Portal

The project sponsor insisted on using the Microsoft .NET Framework which forms the basis for the CareVoyant Main Application Suite's Service Oriented Architecture as shown in the figure 1. This n-tier design logically segments the .NET Framework into three functional levels: presentation layers, business layers and database layers.

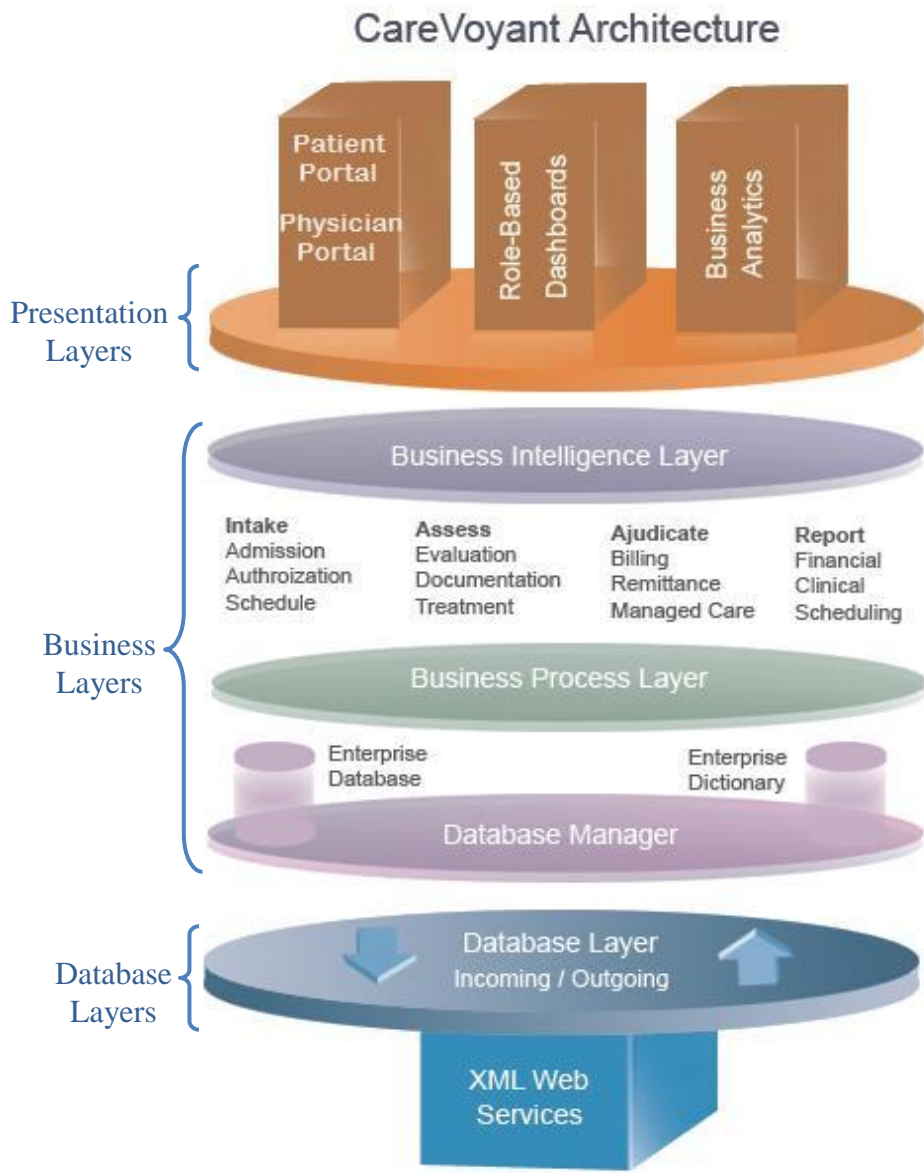


Figure 1. Architecture of CareVoyant Main Application Suite

## 4. The CareVoyant Patient Portal Application

This section explains the high-level architectural design of the CareVoyant Patient Portal application, detailed design architecture, database design, security design, and user interface along with some detailed usability issues that were factors in its construction.

### 4.1. High Level Architectural Design

The high-level architecture of CareVoyant Patient Portal is presented in the Figure 2. CareVoyant Patient Portal was developed using the Microsoft .NET framework and it is written in ASP.NET C#. An installation of the CareVoyant Patient portal executable resides on a server machine and the server will run several instances of the CareVoyant Patient portal for each company that belongs to the enterprise that hosts CareVoyant Main Application Suite. The user, from a client machine belonging to a particular company of the enterprise, will run an instance of the CareVoyant Patient Portal Application.

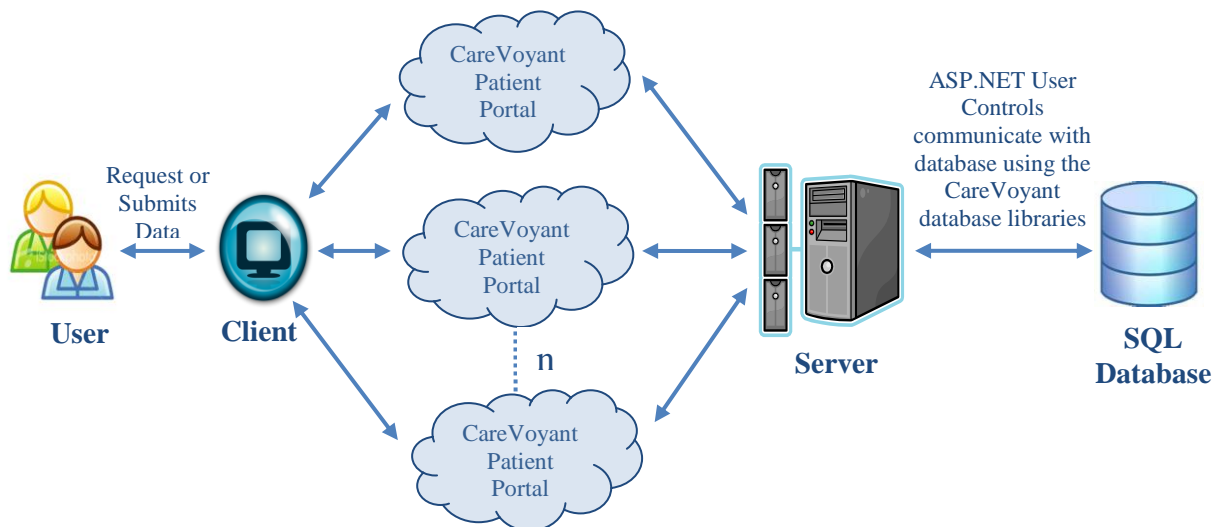


Figure 2. High-Level Architecture of CareVoyant Patient Portal

The CareVoyant Patient Portal application is a data driven application that retrieves information from and stores information to an SQL database based on the commands requested by the user.

The CareVoyant Patient Portal is designed in such a way that the system can be deployed as two-tiered and three tiered architecture and in certain cases, as an n-tiered architecture. In the two-tiered architecture, the client and server machines form the two tiers, and the application and database both reside in the server. In a three-tiered architecture, the third tier is introduced by separating the application from the database. Finally, in the n-tiered architecture, the client will be a separate machine, the application will be spread across several machines and also the database will be spread across several machines. This is the current architecture of the CareVoyant production environment. The CareVoyant Patient Portal was designed in this way to enable flexibility to adapt to different architectures.

## **4.2. Detailed Architecture of CareVoyant Patient Portal**

The CareVoyant Patient Portal architecture strictly follows the proprietary CareVoyant architecture, in which the application is structured as a group of classes. The CareVoyant Patient Portal application comprises of three layers, (1) presentation layer, (2) application layer and (3) database layer. The presentation layer consists of the classes that are responsible for establishing an interface between the other layers in the hierarchy and the end users. The application layer contains the application domain logic. It is also considered to be core engine of the application and acts as a bridge between the presentation layer and the database layer. The database layer contains the data and processes that manipulate the data requests and storage. The database layer may include several databases; in particular for CareVoyant application, the database consists of the company's database, the system database and third party information database.

Figure 3 illustrates the detailed architecture of the CareVoyant Patient Portal using a UML Class diagram. The CareVoyant Patient Portal class contains user login functionality and the main home page functionality which connect to the other functional areas of the application.

The application has three different types of users. They are (1) Enterprise Administrator (2) Patient and (3) Guarantor. Figure 4 and Figure 5 illustrate the accessible functionalities of the CareVoyant Patient Portal application using a use case diagram.

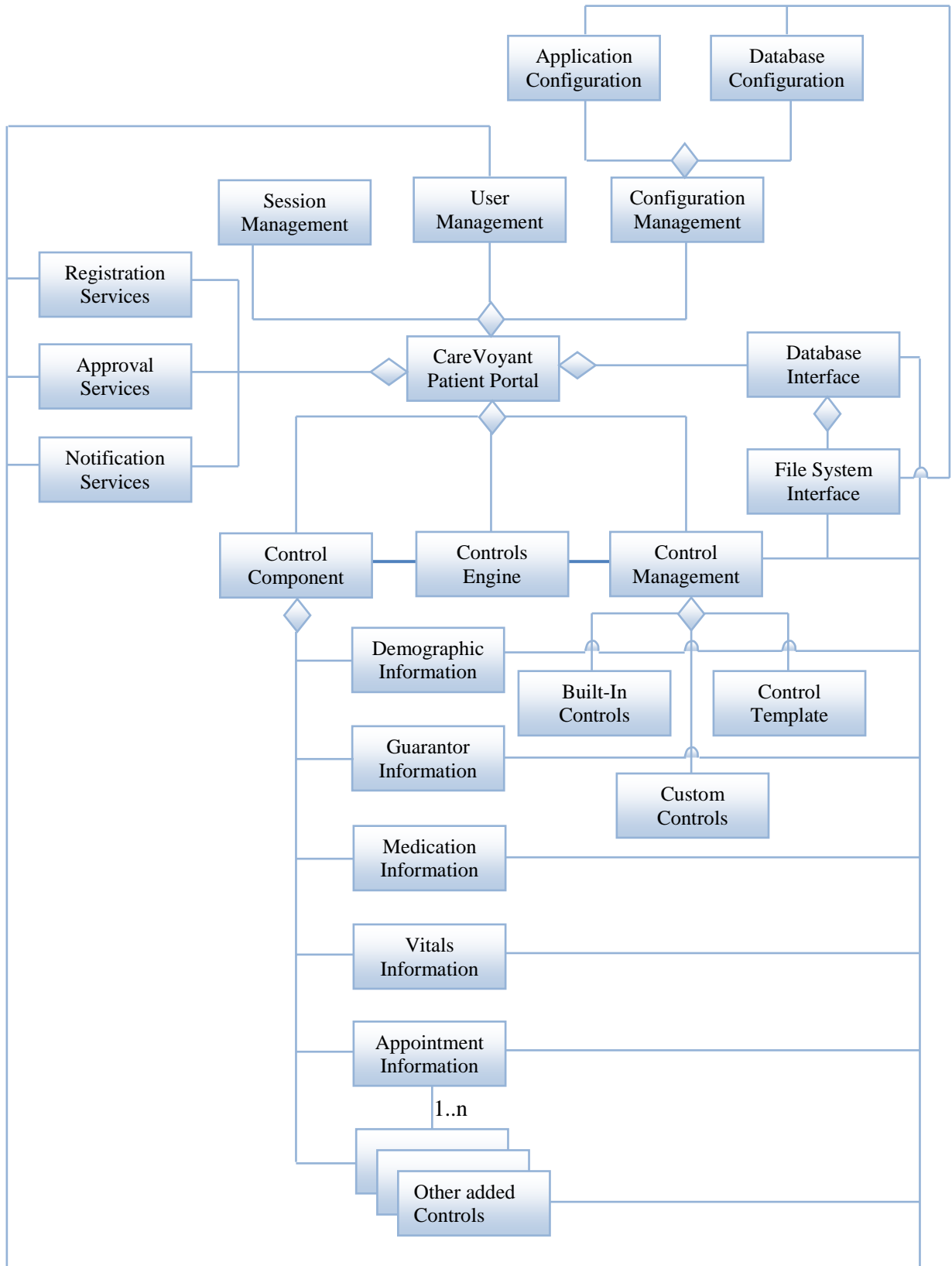


Figure 3. Detailed Architecture of CareVoyant Patient Portal

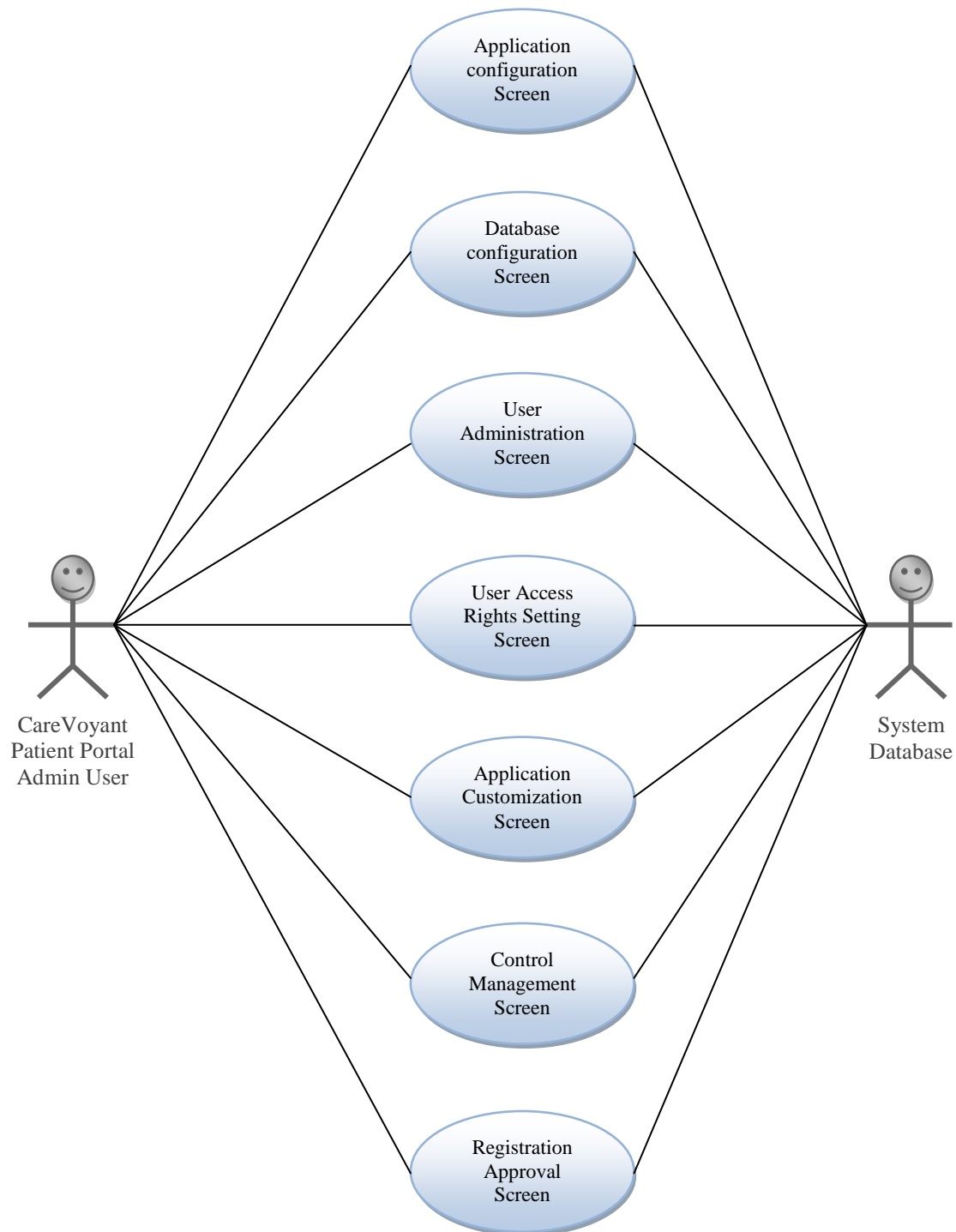


Figure 4. Accessible Functionalities of CareVoyant Patient Portal by Administrative user



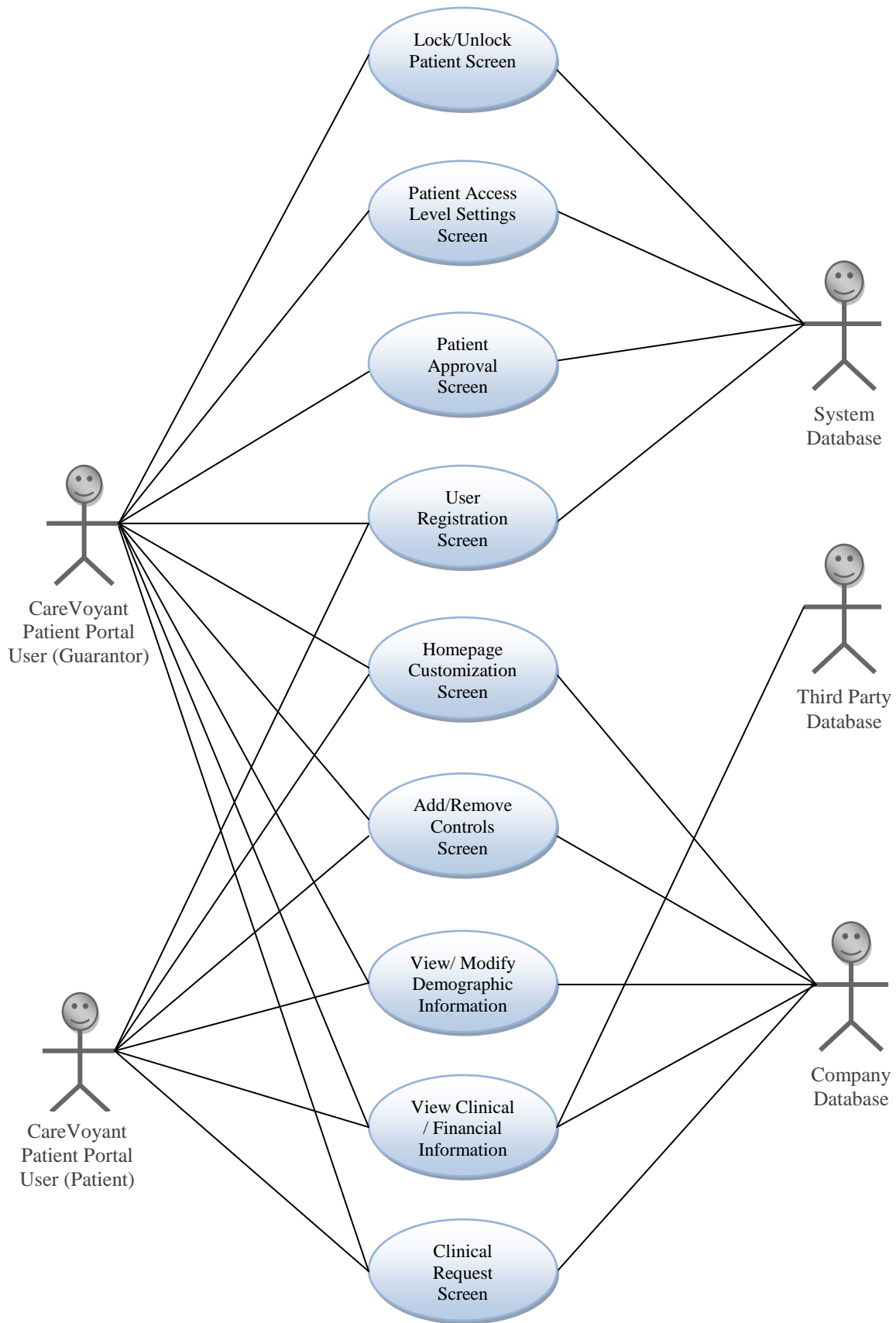


Figure 5. Accessible functionalities of CareVoyant Patient Portal by Patient and Guarantor users

### **4.3. Database Design**

The Patient Portal uses three different databases namely the company database, the system database and the third party database, all together comprising of more than 3700 tables. Therefore, it was a very complicated task to understand the existing database, to figure out the tables that the CareVoyant Patient Portal will use and finally, to come up with the database design for adding new tables to the existing three databases without affecting the tables and data in the existing databases

The database design strictly followed the standard CareVoyant database architecture and the naming conventions. This was done to make it less complicated for the CareVoyant developers to understand the newly added tables and will be able to modify it without any difficulty. Figure 6 and Figure 7 illustrate the database diagrams for the newly added tables to company and system databases respectively for the CareVoyant Patient Portal system.

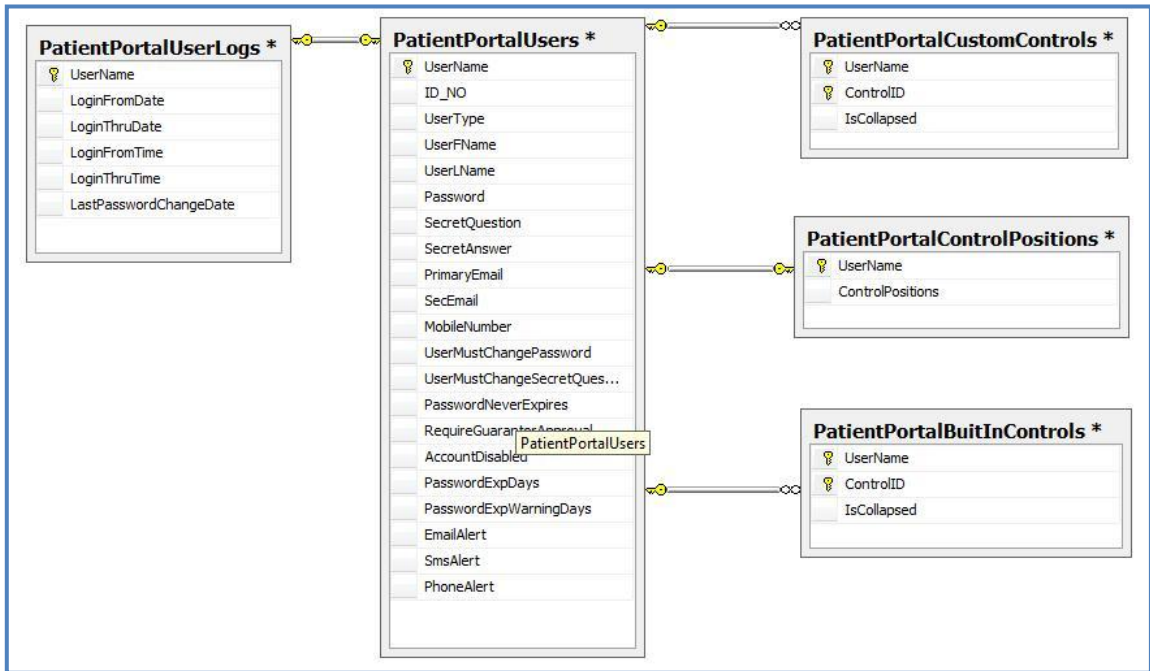


Figure 6. Database Diagram of CareVoyant Patient Portal for newly added Company Tables (Company Database)

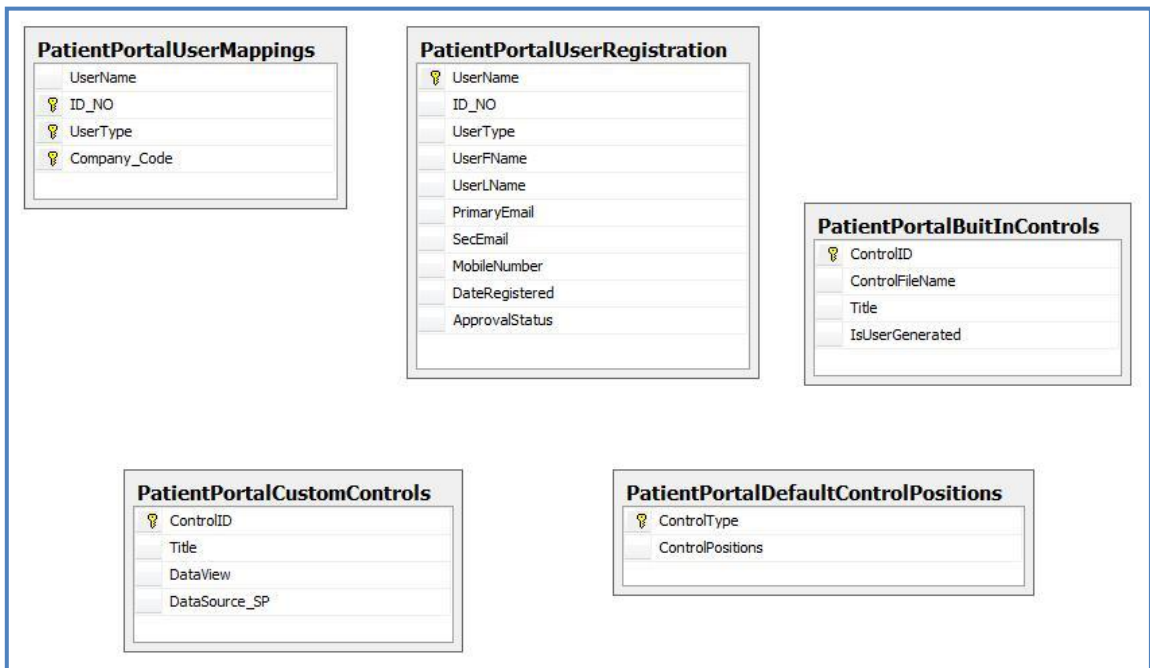


Figure 7. Database Diagram of CareVoyant Patient Portal for newly added System Tables (System Database)

#### **4.4. User Interface Design**

The sponsor wanted the application to be completely configurable and personalized by end users according to their needs. In addition, the sponsor also requested that the user interface should be as simple as possible and should accommodate all the required functionalities that the end users might expect. In order to satisfy these two requirements specified by the sponsor, a third party user interface tool called Telerik Radcontrols for ASP.NET is used to create customizable and friendly user interface for this application.

The CareVoyant Patient Portal contains an advanced and more user friendly web-based user interface, by eliminating some of the disadvantages of a conventional web-based user interface such as page post-backs and high response time. The performance of the product was greatly improved by having a user interface that will avoid page post back for most of the operations.

The look and feel of the product was enhanced by using third party tools and skins. The CareVoyant Patient Portal takes advantage of the Microsoft Visual Studio .NET controls for ASP.NET, third party controls for ASP.Net and the concept of custom user controls available in ASP.NET. The code is written in C# along with JavaScript to enhance the look and feel of the user interface. The user interface code is clearly separated from the application logic code by making use of the code behind methodology that is offered as a choice in Microsoft Visual Studio .NET. This allows better encapsulation, easier maintenance and better readability and flexibility.

As discussed in an earlier chapter, each of the core functionalities in this application is built as a pluggable component using a custom user control available in ASP.NET. The product took advantage of the model of Microsoft Visual Studio .NET Custom user controls along with the third party tool called Telerik Radcontrols [6] for ASP.NET C# and also the feature that is available in ASP.NET called “web parts”. These tools allowed the product to be customized, such as the functionalities can be dragged and dropped or removed from the users screen and saved for future visits to the application. Each such control has a defined specification that helps in reuse of

this functionality by any other CareVoyant application as a plug-in control. This concept of customized user control, led a way to a new concept of advanced user interface design. The user interface of the application can be dynamically configured or modified to add or remove functionalities without modifying the code or redeploying the application.

Apart from the read only built-In controls developed during the design time, there are three other types of pluggable user interface control/component that an administrator of this application can create during the runtime. They are: (1) Simple Custom Control (2) Custom Built-In Control and (3) Imported Built-In Custom Controls. The differences between all the types of pluggable controls are illustrated in the Table 9.

|   | <b>Control / Component Type</b> | <b>Features</b>   |
|---|---------------------------------|---|
| 1 | Simple Custom Controls          | <ul style="list-style-type: none"> <li>• These types of controls are included in the application dynamically during runtime. Each such control is re-built every time the application is used based on the specification saved by the administrators during the creation of the control.</li> <li>• This type of control can be removed from the application at any time by the administrators. It is the simplest of all types of controls, because only simple functionality such as display of data can be processed by this control.</li> </ul> |

Table 9. Types of Control / Component

|   | <b>Control / Component Type</b>   | <b>Features</b>   |
|---|-----------------------------------|---|
| 2 | Custom Built-In Controls          | <ul style="list-style-type: none"> <li>• This type of control is built into the application by using the control creation functionality that is available in the application and is rendered to the application during runtime. It can be saved along with the other built-in controls at the application support files location. This control can be removed from the application at any time by the administrators.</li> </ul>  |
| 3 | Imported Built-In Custom Controls | <ul style="list-style-type: none"> <li>• These types of control are built separately by the CareVoyant developers at the CareVoyant development location or by the other developers at the enterprise locations, by following the defined specification and based on the requirement of the enterprise. These controls are subsequently imported into the application by using the import controls functionality. The added controls are rendered to the application during runtime in a similar way to the customized built-In controls. They can also be saved along with the other built-in controls at the application support files location. This control can be removed from the application at any time by the administrators.</li> </ul> |

Table 9. Types of Control / Component (Continued)

## **4.5. Security Design**

The CareVoyant Patient Portal accesses and maintains highly sensitive and confidential information that must be protected by HIPPA [8] laws and regulations. Therefore security of the system and the data that is handled by this application is an obligatory aspect of this application. With the sponsor's request, the Patient Portal system uses a dynamic link library (DLL) developed by the sponsor company for capturing all security measures that are required for this application. This DLL takes care of the encryption of data as required by the HIPPA security regulations for the data that is handled by this application. In addition, all the data that is stored in the database is encrypted, by using the built in encryption feature that is available in SQL Server 2005 Enterprise Edition [13].

End users access the system is based on their roles. The users are given access to this application only after an administrator approves the identity of the users who have requested access to this application through the user registration functionality. In addition to this, to avoid fraud, several notification methods such as E-Mail, SMS, and Phone Notify [7] option are included to alert the users when there is a change made to the respective user data or healthcare information.

## **4.6. Deploying CareVoyant Patient Portal**

The CareVoyant Patient Portal application was deployed by installing the application on a Microsoft IIS Server that has the Microsoft .NET Framework 3.5 or higher installed. The installation process involved a setup of several configuration files as part of the application. The configuration files contain information about the server, file system, mail configuration, license information for third party tools and also the configuration settings for database access. The installation of this application was followed by manually configuring each of the configuration files with appropriate configuration settings information before the application was used for the first time.

In addition, the three existing databases of the CareVoyant main application suite were updated to include the newly created database objects for this application. This was done by the implementation team at CareVoyant by using the installation manual that is submitted to the sponsor on request.

#### **4.7. Testing**

There are three different types of testing applied to the CareVoyant Patient Portal and they are, 1) Unit testing, 2) Integration Testing and 3) System Testing. The developer performed unit testing and integration testing on the CareVoyant Patient Portal application, whereas the system testing is performed by the sponsor company itself due to the involvement of HIPAA regulations. The Unit testing was performed by the developer at the completion of each of the functionalities / component and the developer tested these modules using a real or valid client data, which helped in increasing the accuracy of the testing performed and also validate the testing that was performed. The integration testing was performed at 2 levels, one at the completion of each of the iterations and tested the integration of functionality / components with the CareVoyant Patient Portal and also at a higher level, when the entire CareVoyant Patient Portal application is completed and tested the Integration with the CareVoyant Main Application Suite. The System testing is currently performed by a dedicated team at the sponsor company, which involves testing the application for standards that are required by the HIPAA regulations as defined by the Center of Medicaid and Medicare Services (CMS), which is a part of United States Health Department.



## **5. Limitations**

The one limitation of the CareVoyant Patient Portal application is that it uses the data that is directly stored in the three CareVoyant databases. Therefore, any changes in the architecture or schema of any of the databases will create a major issue in the accuracy of data that is processed by this application. Currently, every time one or more of the databases is changed by the development team for an update or service, this application has to be completely tested before it is used again.

## **6. Continuing Work**

The CareVoyant Patient Portal was created due to the request from the customers of the sponsor company and it is also a part of the long term goal of the sponsor company. The developer packaged the code, database objects, requirement document, design document, user manual document and the installation manual document for the sponsor. The sponsor will complete the additional security enhancement, required HIPAA enhancements and will perform further in-house testing.

Currently, the custom controls files that are created or imported are not encrypted in this application. These controls can only be read using Microsoft visual Studio .NET, since this will create confidentiality and piracy issues when deployed. The sponsor will be working on encrypting these custom control files and create a new CareVoyant file type for these custom controls. This will make the controls to be read only by the CareVoyant proprietary applications.

Further modification to the user interface to add company information such as logo, copyright information and user interface standards to achieve similar look and feel of sponsor company software will also be performed. Sponsor will then present this application to their customers in their upcoming user conference and get an approval and feedback to further enhance or tune up the application before it is added to the CareVoyant application suite package in its next version.

## **7. Conclusion**

The CareVoyant Patient Portal application is a web-application that is packaged with the main CareVoyant application suite. The CareVoyant Patient Portal has been built with advanced user interface technologies, where the user interface can be customized according to the preferences desired by the enterprise.

The CareVoyant Patient Portal has been designed as a communication interface between the enterprise / company and the patients / guarantors. It captures vital healthcare information of the patients / guarantors stored in the enterprise / company database. It provides all healthcare related information to patients/ guarantors through a secure web environment. The application will provide home page personalization and features customization for every patient / guarantor who uses this application to further enhance the usability of the application by the patients / guarantors.

This application has been built with reusability in mind so that its features are developed as reusable components which can be used in any other product of similar application domain.

## BIBLIOGRAPHY

1. C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development* (Third Edition), Prentice Hall, 2004.
2. S. McConnell, *Rapid Development*, Microsoft Press, 1996.
3. C. Fox, *Introduction to Software Engineering Design: Processes, Principles and Patterns with UML2* (First Edition), Addison Wesley, May 2006.
4. R.T. Futrell, D.F. Shafer, L. Shafer, *Quality software project management- Software Quality Institute series*, Prentice Hall PTR, 2002.
5. H.V. Vliet, *Software Engineering: Principles and Practice* (Second Edition), John Wiley and Sons, 2002.
6. <http://www.telerik.com/products/aspnet-ajax.aspx>, ASP.NET AJAX Controls, Telerik Corporation, August 2009.
7. <http://cdyne.com/products/>, Data Quality APIs, CDYNE Corporation, 2010.
8. <http://www.cms.hhs.gov/hipaageninfo/>, HIPAA – Overview, Centers for Medicare & Medicaid Services, February 2010.
9. [http://www.carevoyant.com/solutions\\_overview.aspx](http://www.carevoyant.com/solutions_overview.aspx), Integrated Healthcare Solutions Overview, CareVoyant Inc, 2010.
10. <http://msdn.microsoft.com/en-us/vcsharp/>, Visual C# Development Center, Microsoft Corporation, January 2010.
11. <http://www.recovery.gov/>, Physician Reimbursement program, American Recovery and Reinvestment Act (ARRA), 2009.
12. [http://www.softpanorama.org/SE/software\\_life\\_cycle\\_models.shtml](http://www.softpanorama.org/SE/software_life_cycle_models.shtml), Software Life Cycle Models, Open Source Software Educational Society, August 2009.
13. <http://www.microsoft.com/sqlserver/2005/en/us/compare-features.aspx>, SQL Server 2005 Features Comparison- Security, Microsoft Corporation, 2009.

## APPENDIX A: Selected CareVoyant Patient Portal Screen Shots



Figure 8. Login Screen

CAREVOYANT  
 Empowering Providers at the Point of Contact  
 Microsoft GOLD CERTIFIED Partner

Home | Request Changes | Add Request | Pending Request | Settings | LogOut

My Website

| Patient No | Last Name | First Name | Address        | City      | Zipcode |
|------------|-----------|------------|----------------|-----------|---------|
| 2042       | ANDREWS   | SAMANTHA   | 378 RIVER ROAD | SLU GROVE | 8088    |

Ready

Demographic Information

Name: SAMANTHA ANDREWS

Address: 378 RIVER ROAD SLU GROVE IL 8088

Date of Birth: 02/12/1923    Age: 87    Gender: F

Phone: 8475555555    Emergency Phone: 8478755552

Contact: JOHN ANDREWS    Primary Physician: SAMANTHA BAUER

Employer:    Address:    Work Phone: -

Last Statement Date: 05/19/2008    Balance: \$2719.0000

Guarantor Information

First Name: SAMANTHA    Middle Initial:    Last Name: ANDREWS

Address: 378 RIVER ROAD SLU GROVE IL 8088

Age: 87    Phone: 8475555555    Emergency Phone: 8478755552

Cell:    Email: johnandrews@aol.com

Contact: JOHN ANDREWS

Employer:    Address:    Work Phone: -

Professional Contacts

No Professional Contacts data available for the Selected Patient.

Ready

Account Statements

Recent Statements

Tuesday, August 05, 2008    STU17 INCL 2    [View](#)    [Download](#)

Insurance Plan

Insurance Name: PRIVATE.PAY

Policy Number:    Group Number:

Relationship to Insured: PATIENT IS THE INSURED

Guarantor Noted (need insured name): 20752

CO Type: 0.0000

Effective: 01/01/1900    Expiration: 01/01/1900

Appointment Information

Please Select the Patient from Above

Ready

Clinical Information - Medication

| Medications   | Status | LastName/FirstName | Physician from Provider Codes | Interaction |
|---|--------|--------------------|-------------------------------|-------------|
| No Medication data available for the Selected Patient |        |                    |                               |             |

Ready

Clinical Information - Vitals

| Date  | Time | Temperature | Pulse | Blood Pressure (Systolic / Diastolic) | Weight | Bit |
|---|------|-------------|-------|---------------------------------------|--------|-----|
| No Vitals data available for the Selected Patient |      |             |       |                                       |        |     |

Ready

Copyright © 2009 Carevoyant, Inc. All rights reserved.    Disclaimer | Privacy Policy

Figure 9. Guarantor Homepage

| My Patients |           |            |                 |           |         |
|-------------|-----------|------------|-----------------|-----------|---------|
| Patient No  | Last Name | First Name | Address         | City      | Zipcode |
| 20843       | ANDREWS   | SAMANTHA   | 5788 RIVER ROAD | ELK GROVE | 60065   |
| Ready       |           |            |                 |           |         |

Figure 10. My Patients Control for Guarantor

| Demographic Information                            |  |
|--|--|
| <b>Name:</b> SAMANTHA ANDREWS                      |  |
| <b>Address:</b> 5788 RIVER ROAD ELK GROVE IL 60065 |  |
| <b>Date of Birth:</b> 02/12/1923                   | <b>Age:</b> 87 <b>Gender:</b> 2          |
| <b>Phone:</b> 8478989889                           | <b>Emergency Phone:</b> 8476878980       |
| <b>Contact:</b> JOHN ANDREWS                       | <b>Primary Physician:</b> SAMANTHA BAKER |
| <b>Employer:</b>                                   |  |
| <b>Address:</b>                                    |  |
| <b>Work Phone:</b> -                               |  |
| <b>Last Statement Date:</b><br>08/05/2008          | <b>Balance:</b> \$3319.00000             |

Figure 11. Demographic Information Control



Guarantor Information

|  |                                   |                                       |
|--|-----------------------------------|---------------------------------------|
| <b>First Name:</b> SAMANTHA                        | <b>Middle Initial:</b>            | <b>Last Name:</b> ANDREWS             |
| <b>Address:</b> 5788 RIVER ROAD ELK GROVE IL 60065 |                                   |                                       |
| <b>Age:</b> 87                                     | <b>Phone:</b> 8478989889          | <b>Emergency Phone:</b><br>8476878980 |
| <b>Cell:</b>                                       | <b>Email:</b> johnandrews@aol.com |                                       |
| <b>Contact:</b> JOHN ANDREWS                       |                                   |                                       |
| <b>Employer:</b>                                   |                                   |                                       |
| <b>Address:</b>                                    |                                   |                                       |
| <b>Work Phone:</b> -                               |                                   |                                       |

Figure 12. Guarantor Information Control

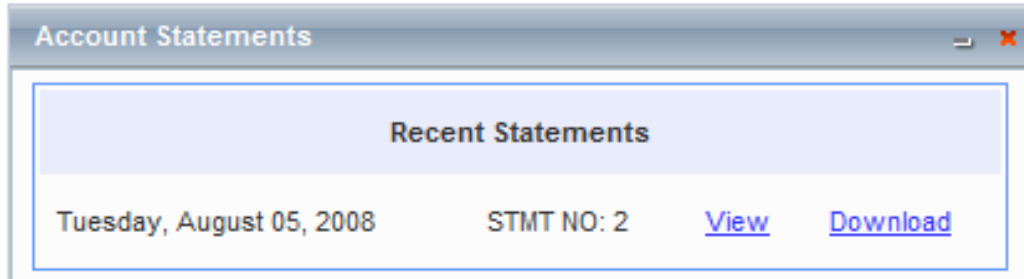


Figure 13. Account Statement Control

**Patient Plan**

**Insurance Name: PRIVATE PAY**

**Policy Number:**                      **Group Number:**

**Relationship to Insured: PATIENT IS THE INSURED**

**Guarantor\_No(we need insured name): 20782**


**CO Pay: 0.00000**

|                              |                               |
|------------------------------|-------------------------------|
| <b>Effective: 01/01/1900</b> | <b>Expiration: 01/01/1900</b> |
|------------------------------|-------------------------------|

Figure 14. Patient Insurance Plan Control

| Demographic Information           |                              |                                  |
|-----------------------------------|------------------------------|----------------------------------|
| <b>Patient No:</b> 20843          | <b>Last Name:</b> ANDREWS    | <b>First Name:</b> SAMANTHA      |
| <b>Middle Initial:</b>            | <b>DOB:</b> 02/12/1923       | <b>Age:</b> 87                   |
| <b>Sex:</b> Female                | <b>SSN:</b> 213344687        | <b>Address1:</b> 5788 RIVER ROAD |
| <b>Address2:</b>                  | <b>City:</b> ELK GROVE       | <b>State:</b> ILLINOIS           |
| <b>County:</b> MIDLAND            | <b>Zip:</b> 60065            | <b>Phone:</b> 8478989889         |
| <b>Cell Phone:</b> 8479897790     | <b>Emg.Phone:</b> 8476878980 | <b>Emg.Contact:</b> JOHN ANDREWS |
| <b>Email:</b> johnandrews@aol.com |                              | <a href="#">Edit</a>             |

Edit Information of Patient with Patient\_NO : 20843

|                    |   |                  |  |
|--------------------|---|------------------|--|
| Last Name:         | <input type="text" value="ANDREWS"/>      | First Name:      | <input type="text" value="SAMANTHA"/>  |
| Middle Name:       | <input type="text"/>                      | Date of Birth:   | <input type="text" value="2/12/1923"/>  |
| Age:               | <input type="text" value="87"/>           | Sex:             | <input type="text" value="Female"/>  |
| SSN:               | <input type="text" value="213344687"/>    | Address 1:       | <input type="text" value="5788 RIVER ROAD"/>   |
| Address 2:         | <input type="text"/>                      | City:            | <input type="text" value="ELK GROVE"/>   |
| State:             | <input type="text" value="ILLINOIS"/>     | County:          | <input type="text" value="MIDLAND"/>   |
| Zip code:          | <input type="text" value="60065"/>        | Phone:           | <input type="text" value="8478989889"/>  |
| Cell Phone:        | <input type="text" value="8479897790"/>   | Emergency Phone: | <input type="text" value="8476878980"/>  |
| Emergency Contact: | <input type="text" value="JOHN ANDREWS"/> | Email:           | <input type="text" value="johnandrews@aol.com"/>   |

[Update](#) [Cancel](#)

Ready

Figure 15. Demographic Information change request Control

| Clinical Information - Vitals |          |             |       |   |          |                |
|-------------------------------|----------|-------------|-------|---|----------|----------------|
| Date                          | Time     | Temperature | Pulse | <-----Blood Pressure-----><br>(Standing   Sitting   Supine) | Weight   | BMI            |
| 12/31/2007                    | 2:34 PM  | 97.7 F      | 64    | 0/0 120/68 0/0  | 0.00 lbs | 0.00 lbs/in*in |
| 12/31/2007                    | 2:12 PM  | 96.5 F      | 60    | 0/0 122/62 0/0  | 0.00 lbs | 0.00 lbs/in*in |
| 12/28/2007                    | 1:10 PM  | 96.7 F      | 64    | 0/0 116/68 0/0  | 0.00 lbs | 0.00 lbs/in*in |
| 12/27/2007                    | 10:19 AM | 98.5 F      | 68    | 0/0 120/64 0/0  | 0.00 lbs | 0.00 lbs/in*in |
| 12/27/2007                    | 10:14 AM | 97.1 F      | 64    | 0/0 124/62 0/0  | 0.00 lbs | 0.00 lbs/in*in |
| 12/26/2007                    | 9:58 AM  | 97.3 F      | 64    | 0/0 132/66 0/0  | 0.00 lbs | 0.00 lbs/in*in |
| 12/26/2007                    | 9:53 AM  | 96.1 F      | 64    | 0/0 124/68 0/0  | 0.00 lbs | 0.00 lbs/in*in |
| 12/25/2007                    | 12:53 PM | 96.7 F      | 72    | 0/0 106/64 0/0  | 0.00 lbs | 0.00 lbs/in*in |
| 12/24/2007                    | 11:03 AM | 97.2 F      | 64    | 0/0 122/68 0/0  | 0.00 lbs | 0.00 lbs/in*in |
| 12/24/2007                    | 10:57 AM | 97.2 F      | 64    | 0/0 138/72 0/0  | 0.00 lbs | 0.00 lbs/in*in |

Change page: < > Displaying page 1 of 6, items 1 to 10 of 56.

Ready

Figure 16. Vitals Control

| Clinical Information - Medication |   |        |                    |                                     |             |
|-----------------------------------|---|--------|--------------------|-------------------------------------|-------------|
|                                   | Medications   | Status | LastName,FirstName | Physician phone from Provider Codes | Interaction |
| ▶                                 | NOVOLIN N 100 UNIT/ML SUSP, SUB-Q INJ SUBCUTANEOUS EVERY MORNING 12 UNITS | Active | L001430 F001430    | 8479259148                          | No          |
| ▼                                 | WELLBUTRIN SR 150 MG TAB ORAL 2 TIMES A DAY                               | Active | L001430 F001430    | 8479259148                          | <u>Yes</u>  |

**Medication Name: WELLBUTRIN SR 150 MG TAB**

**Details:**

**IMPORTANT NOTE:** This is a summary and does not contain all possible information about this product. For complete information about this product or your specific health needs, ask your health care professional. Always seek the advice of your health care professional if you have any questions about this product or your medical condition. This information is not intended as individual medical advice and does not substitute for the knowledge and judgment of your health care professional. This information does not contain any assurances that this product is safe, effective, or appropriate for you. BUPROPION SUSTAINED-RELEASE (ANTIDEPRESSANT) - ORAL (byou-PRO-pee-on)

**COMMON BRAND NAME(S):** Wellbutrin SR

**WARNING:** Antidepressant medications are used to treat a variety of conditions, including depression and other mental/mood disorders. These medications can help prevent suicidal thoughts/attempts and provide other important benefits. However, studies have shown that a small number of people (especially children/teenagers) who take antidepressants for any condition may experience worsening depression, other mental/mood symptoms, or suicidal thoughts/attempts. Therefore, it is very important to talk with the doctor about the risks and benefits of antidepressant medication (especially for children/teenagers), even if treatment is not for a mental/mood condition. Tell the doctor immediately if you notice worsening depression/other psychiatric conditions, unusual behavior changes (including possible suicidal thoughts/attempts), or other mental/mood changes (including new/worsening anxiety, panic attacks, trouble sleeping, irritability, hostile/angry feelings, impulsive actions, severe restlessness, very rapid speech). Be especially watchful for these symptoms when a new antidepressant is started or when the dose is changed.

**USES:** This medication is used to treat depression. It can improve your mood and feelings of well-being. It works by helping to restore the balance of natural chemicals (neurotransmitters) in your brain. Bupropion is also used to help people quit smoking by decreasing cravings and nicotine withdrawal effects.

Figure 17. Medication Control with built-in Drug Interaction and Drug Education

Account Statement-Tuesday, August 05, 2008 - Windows Internet Explorer  
 http://localhost/Patient%20Portal/PatientStatement.aspx?StatementNo=2

**CAREVOYANT**  
 2801 S. HULEN #500  
 FORT WORTH, TX 76109

| STATEMENT DATE | PAY THIS AMOUNT | ACCOUNT NO. |
|----------------|-----------------|-------------|
| 8/5/2008       | \$1500.00       | 20782       |

For all billing questions, call: 817-731-6121  
 Patient Name: ANDREWS, SAMANTHA

ADDRESS: [REDACTED]

ANDREWS, SAMANTHA  
 5788 RIVER ROAD  
 ELK GROVE IL 60065

MAKE CHECKS PAYABLE / REMIT TO:  
 CAREVOYANT  
 1821 WALDEN OFFICE SQUARE  
 SCHAUMBURG IL 60173

**STATEMENT**

| TRANSACTION DATE | DOCTOR | DESCRIPTION                | PROC CODE | CHARGES/ DEBITS/ | PMTS/ CREDITS |
|------------------|--------|----------------------------|-----------|------------------|---------------|
|                  |        | PREVIOUS BALANCE           |           | 0.00             |               |
| 8/5/2008         | 000003 | FORTEO MEDICATION 1 PENLET | ZX1002    | 1500.00          | 0.00          |

| Current   | 30 Days | 60 Days | 90 Days | 120 Days | 150 Days | Total Balance | Ins. Pending       |
|-----------|---------|---------|---------|----------|----------|---------------|--------------------|
| \$1500.00 | \$0.00  | \$0.00  | \$0.00  | \$0.00   | \$0.00   | \$1500.00     | \$0.00             |
|           |         |         |         |          |          |               | <b>PATIENT DUE</b> |
|           |         |         |         |          |          |               | \$1500.00          |

For all billing questions, call: 817-731-6121  
 Patient Name: ANDREWS, SAMANTHA

Done Internet | Protected Mode: On 70%

Figure 18. Account Statement Viewer

Patient Portal User Approval

| New Registrations                |       |          |           |           |                |
|----------------------------------|-------|----------|-----------|-----------|----------------|
| UserName                         | ID_NO | UserType | UserFName | UserLName | DateRegistered |
| No New Registration is available |       |          |           |           |                |

Select Company code :  Guarantor Approval Required ? :

[Available Matches](#)

Please select the User Registration from above to perform Verification

| Available Companies  |              |     |
|----------------------|--------------|-----|
| Company Name         | Company Code |     |
| > CAREVOYANT COMPANY | DEMO         | NUR |

Figure 19. User Approval Screen



**Custom Control Management**

ControlID:

Title:

DataView:

DataSource\_SP:

Figure 20. Custom Control Management Screen

**Built-In Custom Control Management**

**CareVoyant Built-In Controls**

| ControllID               | ControlFileName               | Title                             | IsUserGenerated          |
|--------------------------|-------------------------------|-----------------------------------|--------------------------|
| <a href="#">Edit</a> CV1 | AdmissionInfo.ascx            | Appointment Information           | <input type="checkbox"/> |
| <a href="#">Edit</a> CV2 | ClinicalInfo_Medications.ascx | Clinical Information - Medication | <input type="checkbox"/> |
| <a href="#">Edit</a> CV3 | ClinicalInfo_Vitals.ascx      | Clinical Information - Vitals     | <input type="checkbox"/> |
| <a href="#">Edit</a> CV4 | DemographicInfo.ascx          | Demographic Information           | <input type="checkbox"/> |
| <a href="#">Edit</a> CV5 | GuarantorInfo.ascx            | Guarantor Information             | <input type="checkbox"/> |
| <a href="#">Edit</a> CV6 | PatientPlan.ascx              | Patient Plan                      | <input type="checkbox"/> |
| <a href="#">Edit</a> CV7 | ProfessionalContactInfo.ascx  | Professional Contacts             | <input type="checkbox"/> |
| <a href="#">Edit</a> CV8 | AccountStatementInfo.ascx     | Account Statements                | <input type="checkbox"/> |

Select Task :  Create New BuiltIn Control  
 Upload New BuiltIn Control

ControllID:

Title:

DataView:

DataSource\_SP:

ASCX File :

ASCX.CS File:

Figure 21. Built-in Custom Control Management Screen

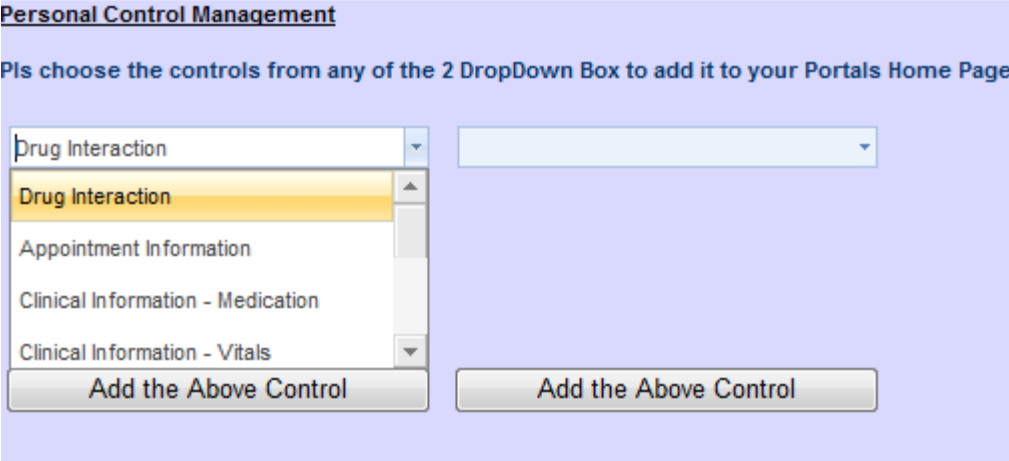


Figure 22. Homepage Control Management Screen for Patient / Guarantor