

Design and Enhancement of Distributed Architecture in Health Insurance System

R.Sriram¹, Dr J. Andrews²

^{1,2}Sathyabama University, Chennai, TamilNadu

Abstract

Background: The main objective of this paper is to review the literature about Distributed Architecture in Health Insurance System. Health Axis System involves major factors for applying Asp.Net Web Applications, Different Authentication, Authorization and Framework Techniques, covered for Different Coding Techniques. **Methods and Statistical Analysis:** This Paper is explained about distributed architecture online health Insurance. Health Axis Information about all sort of medical services providing by the hospital to corporate companies, claim settlements & adjustments services authorized by insurance companies. **Findings:** Experiment is analyzed to Create Secured Frame Implement Secured Health Insurance System. The Experimental Program Code Result is given to proposed method. **Conclusion:** In This paper give the point by point review of different methods and Techniques with coding structures to create Secured Architecture in Asp.Net Web Applications.

Keywords: Primary health Care, Claim Processing, adjudication, Quality healthcare, Co-Insurance, Premium

I. INTRODUCTION

Most of the Health Insurance System developed by different vendors and develop different plat forms. Interoperability is the ability of different subsystems to access and use the data reliably and quickly from various sources without the occurrences¹. Estimating the overall risk of health care expenses, among a targeted group, an insurer can develop a routine finance structure, such as a monthly premium or payroll tax; ensure that money is available to pay for the health care benefits specified in the insurance agreement¹. The benefit is administered by a central organization such as a government agency, private business, or not-for-profit entity. In the current healthcare scenario, patients have multiple health records in different healthcare information systems. Therefore interoperability in healthcare information systems ever more becomes a requirement rather than a feature.

II. TECHNOLOGY OVERVIEW

The most suitable approach to handle the interoperability issue among the crucial medicinal services segments in appropriated human services environment is blend of smart specialists and case based thinking. The unmistakable elements of each, gainful for the proposed framework, are portrayed underneath¹.

A. Client Layer

It is the Presentation Layer where in which the user interface for the client built.

B. Application Logic Layer

Application logic insurance engine is built using Asp.Net, MVC, and COM Components. A request may be fetching information, storing information, searching information².

The Client request to the Server Engine the WEB API Processing the request to Database Layer and back to UI Layer^{1,2}.

C. Data Layer

The Engine uses SQL Server 2012 or Cloud SQL Database for Persistence storage information. Insurance database representing data layer.

III. PROBLEMS IN THE EXISTING SYSTEM

1. Limited in Single System. (Need to execute Service Oriented Architecture)³.
2. Fewer clients Friendly.
3. Having part of manual work (Manual System does not imply that you are working with pen and paper, it additionally incorporate taking a shot at spread sheets and other basic programming).

4. Requires more number of representative's to work. (Not computerized)
5. Its time consuming process.
6. Less Storage for existing System.
7. Less performance.
8. Claim Processing will less Secure like HIFA and FIFA Processing³.
9. Every day Batch Process not implemented.
10. Track record its very tedious process.
11. Less secured for existing system in Database(SQL Injection Attack).

IV. GOAL OF THE PROPOSED SYSTEM

This task is extremely famous and generally utilized as a part of numerous enormous associations. This venture is meant to building up an online medical coverage against its worker. This is an online framework and is utilized keep up the vast majority of the exercises happening in enormous associations⁴. The Health Axis framework will keep up all data identified with the representative, support of cases relating (Batch procedure) to the insurance agencies, data about all kind of medicinal administrations giving by the doctor's facility to the corporate workers, claim settlements and changes against their administrations writing by the protection companies⁵. Furthermore this framework can likewise keep up the wellbeing related all data of the employees⁵

V. SOLUTION OF THESE PROBLEMS

The improvement of the new framework contains the accompanying exercises, which attempt to computerize the whole process keeping in perspective of the database coordin⁵.

1. The application with various controls (Telerik, Ajax, Anjular JS for validations, version control Tortoise workbench).
2. Host the application Cloud Computing access through VMWare
3. The system makes the overall project management much easier and flexible.
4. Various custom components used to implement entire application's Like (AdRotator, JSGrid, WorkBenchTools, CustomCalender control Creation, Autoevent Environment ect..).

5. There is no risk data management at any level while the project development is under process.
6. It Provides high level Protocols using secured manner.
7. Implement highly secured architecture for MVC Security controls.
8. To implements WebApi once complete the project to implement different client locations to Service only.
9. Increase DataBase storage space to implement SQL Server or BIGData , Cloud SQL.
10. The System To implement Cloud Environment to access everywhere for performance will increase.
11. To Implement (BATCH Process for the Entire System) . it's used for automatically track the track efficient manner.

VI. STUDY OF THE SYSTEM

In the adaptability of the utilizations the interface has been created a design idea at the top of the priority list, related through a program interface. The GUI'S at the top level have been classified as⁵.

1. Administrative User Interface.
2. Employee User Interface.
3. Claim Generation User Interface.
4. Batch Process User Interface.
5. The regulatory client interface focuses on the reliable data that is for all intents and purposes, part of the hierarchical exercises and which needs appropriate confirmation for the information gathering^{4,5}. The interfaces help the organizations with all the value-based states like Data insertion, Data erasure and Data redesigning alongside the broad information seek capabilities⁵.
6. The operational or non specific client interface helps the clients upon the framework in exchanges through the current information and required services⁶. The operational client interface likewise offers the normal clients in dealing with their own data some assistance with helping the conventional clients in dealing with their own data in an altered way according to the helped adaptabilities^{6, 7}.

VII. ARCHITECTURAL AND DESIGN FEATURES

Sr. No	Architecture/Design Feature	Description
1	Presentation	
1.1	HTML/ASPX Pages	Use .cshtml with Razor Syntax
1.2	Cascade style sheets (CSS), Themes, Skins	Style Definitions
1.3	Images and Logos	File system/get from the website folders;
1.4	Clientside validations	jQuery, MVC and Data Annotations
1.5	jQuery controls/Plugins	Identify jQuery Plugins
1.6	Third party custom controls	Telerik MVC/jQuery based grid controls
1.7	Navigation	Use CSS based Top Menu, Left side Navigation
1.8	Alerts and Messages	Use jQuery Modal for alerts. jQuery Modal gives flexibility over the interface and actions
1.9	Popup windows	Use jQuery Modal to provide popup functionality; Need customization options like minimize, drag , maximize where required
1.10	AJAX	Validation, load data on demand
1.11	jQuery Server calls	JSON/RESTful services, Use jQuery .ajax function with
1.12	Page Refresh Full/Partial	Use AJAX for Partial Refresh
1.13	Browser compatibility	Target IE 7/8/9, Firefox, Chrome, Safari, Modernizr. Use CSS Reset to avoid cross-browser differences
1.14	User Experience and Usability	User Friendly
1.15	Data Pagination	Load on scroll
1.16	Mobile UI	Mobility UI

Sr. No	Architecture/Design Feature	Description
1.17	Tablet UI	Tablet UI Design
1.18	Sitemap	Global Site Map
2	Business Logic	
2.1	Business Rules and Validations	Configure Business Rules by Customer and stored in Meta Data Services database
2.2	Business Implementation Classes	Business Layer
2.3	Data Transfer Objects	Make them as close as possible to Database, Interface on DTOs, Data Mappers
2.4	WCF Services	Acts as interface between different environment
2.5	Interfaces (Components)	Communicate for each Layered Structure
2.6	Data annotations	DataRegion Specific View Models/Separate Views By State
2.7	Business Workflows	Identify Business Workflows
2.8	Service Contracts	Define service operations that service support/implement
2.9	Data Contracts	Data contract for communication between services and clients
3	Data	
3.1	Data access classes	DataBase Layer
3.2	Stored Procedures	For Performance wise
3.3	User defined functions	UI Functions
3.4	Views	Multiple user access a Table
3.5	Triggers	Data Manipulation
3.6	Database design	Follow Normalization Process ,etc
3.7	Exception Handling	Error Handling
3.8	Concurrency handling and Transaction Management	Transaction Management

Sr. No	Architecture/Design Feature	Description
3.9	Connection Pooling	Multiple Instance Checking
3.10	Data Migration	Migrate master tables data
3.11	ADO.NET	DataBase Activities
4	Core Framework	
4.1	Session Management	Use Out of Process state server for session management
4.2	Exception Handling	Enterprise Library for exception handling is used Policy will determine when to log, propagate, change the original exception message
4.3	Logging	Enterprise Library Logging application block is used for storing the exception stack trace.
4.4	Caching	Use Enterprise Library for Caching; SQL Cache Dependency. Use caching to store the data for dropdown fields, static pages Cache Invalidation
4.5	Authentication	Forms Authentication will be used and need to develop membership service
4.6	Authorization	Role based; Dynamic roles required for testing
4.7	Auto logout	After a period of inactivity, user will need to be logged out automatically with a prompt before 5 minutes of auto logout
4.8	Namespace and Directory structure	Add Difference Name Space and Components
4.9	Encryption	Query string parameters need to be encrypted
4.10	URL Redirection	Use MVC Redirect Options
4.11	Handling Query	Always encrypt query

Sr. No	Architecture/Design Feature	Description
	string parameters	string parameter values
4.12	Auditing	Detailed information is logged
4.13	Load Balancing	Performance Testing and Review
4.14	Security	SSL, Authentication, Resources like Scripts and Images
5	Framework/Patterns	
5.1	MVC	Presentation design pattern, extensible to other devices
5.2	SOA	Business logic exposed as services
5.3	Design Patterns (Factory, Adapter e.t.c)	Waterfall, Agile Methodology
5.4	Dependency Injection/loC	SQL Dependency
6	Deployment Strategy	
6.1	Deployment Environments (Alpha, Beta, Staging, Prod)	Build and Release
6.2	Virtual Directories / Sub sites	IIS
6.3	Deployment Architecture Cloud	Deployment
7	Other	
7.1	Performance Requirements	DataBase, Coding Standard
7.2	Testing automation	Nunit, Manual
7.3	Reusable Components	Telerik, AdRotator, Infragistics
7.4	External System Dependencies	
7.5	Microsoft Enterprise Library	Ado.Net Framework
7.6	Coding Standards and Best Practices	Uniqueness for Coding Standards
7.7	Code Review Tools	Nunit for Methods Checking
7.8	Detection of Unused Translations	Translations for new detection

Sr. No	Architecture/Design Feature	Description
8	Configuration Management	
8.1	Branching and Merging	SVN

Sr. No	Architecture/Design Feature	Description
8.2	Work Items and Progress Tracking	Tracing ,UTrack Tool
8.3	Automated Builds	Tortoise WorkBench

A. DataFlow Diagrammatical Claim Process Representation

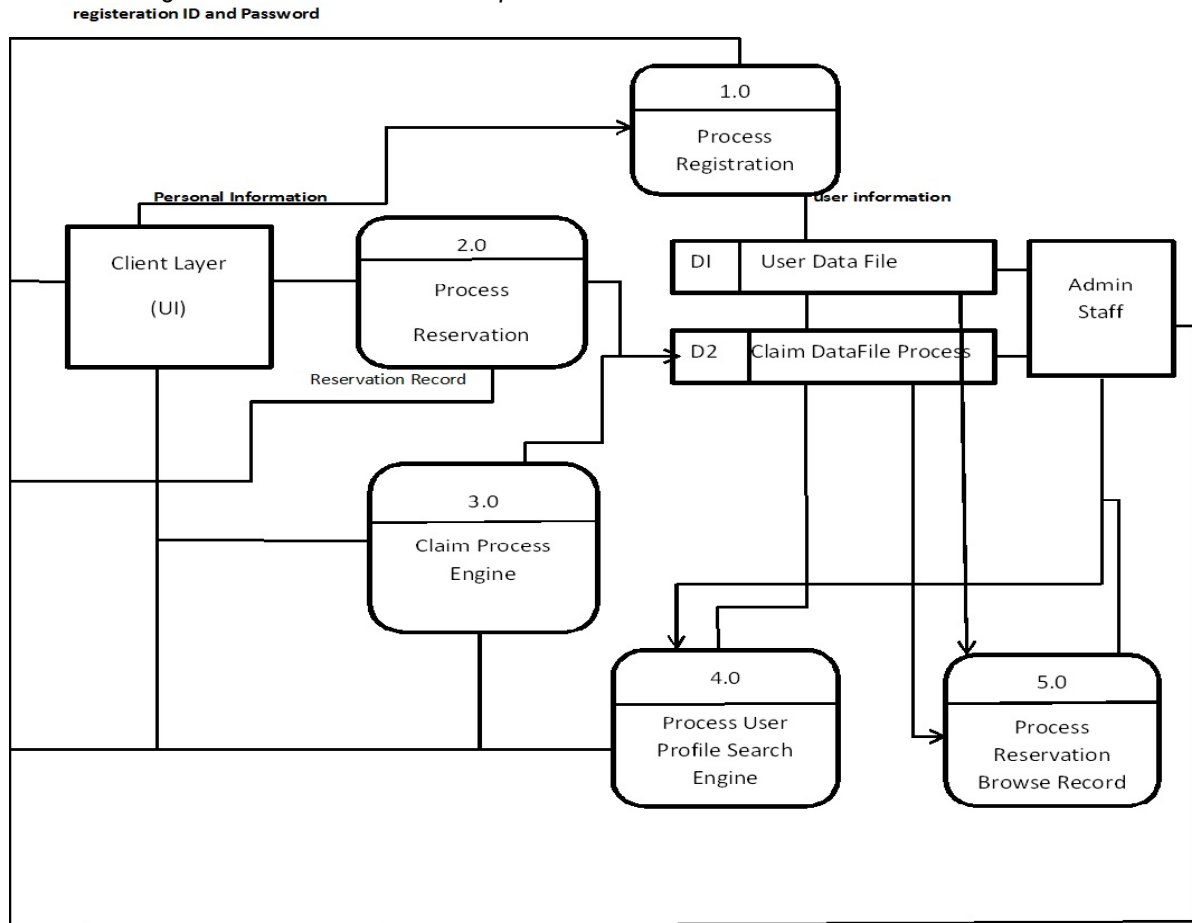


Fig.1 Dataflow Diagram Claim Process

VIII. CONCLUSION

This paper surveys diverse aspects of Insurance-based healthcare technologies and presents various health care network architectures and platforms that support access to the Insurance backbone and facilitate medical data.

REFERENCES

- [1]. Bhat, Jain. Factor Affecting the Demand for Health Insurance in a Micro Insurance Scheme. 2006; Indian institute of management; Ahmedabad: W.P. No. 2006-07-02.
- [2]. Kundu S. Health Insurance – An Alternative Healthcare Financing Mechanism in Rural India. IJHD. 2009; 3(2).
- [3]. Telecom Italia Lab, Java Agent Development Framework, <http://jade.tilab.com/>.
- [4]. World Health Organization. Study of RSBY Health Insurance in India. AHPSP, Alliance for Health Policy and Systems Research; 2011; 2011/126289-0.
- [5]. Abdullah L, Mohd R. IJCSI International Journal of Computer Science Issues. 2012; 9(1): No 2, 1694-0814 ISSN (Online).
- [6]. Barick BL. Occupational health programs: the ounce of prevention paying off. Occup Health Saf. 1985; 38-42.
- [7]. Khajehpour, Salari & E. Raheleh, (2013) "Advancements and Trends in Medical Case-Based"