Physician Portal

Software Architecture Document

Mike Crowell

Version 3.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 6/11/2013 | 1.0 |  | Mike Crowell |
| 6/25/2013 | 2.0 |  | Mike Crowell |
| 7/12/2013 | 3.0 |  | Mike Crowell |

Software Architecture Document

**1.**                  **Introduction**

This Software Architecture Document describes the architecture of the Physician Portal application using different architectural views to illustrate different aspects of the system and to explain the significant architectural decisions. The Software Architecture Document includes a high-level description of the goals of the architecture, the use cases support by the system and architectural styles and components that have been selected to best achieve the use cases.

**1.1**               **Purpose**

The Software Architecture Document provides a comprehensive architectural overview of the Physician Portal Application, designed to fill a need within the health care industry for remote access to physician’s patient information.

**1.2**               **Scope**

The scope of this Software Architecture Document is to depict the architecture of the Physician Portal Application.

**1.4**               **References**

[KRU41]: The “4+1” view model of software architecture, Philippe Kruchten, November 1995, <http://www3.software.ibm.com/ibmdl/pub/software/rational/web/whitepapers/2003/Pbk4p1.pdf>

[RSA]: IBM Rational Software Architect

<http://www-306.ibm.com/software/awdtools/architect/swarchitect/index.html>

[RUP]: The IBM Rational Unified Process :

<http://www-306.ibm.com/software/awdtools/rup/index.html>

**1.5**               **Overview**

Section 2: Architectural Representation

Section 3: Physician Portal

Section 4: Use-Case View

Section 5: Logical View

Section 6: Process View

Section 7: Deployment View

Section 8: Implementation View

Section 9: Data View

Section 10: Size and Performance

Section 11: Quality

**2.**                  **Architectural Representation**

This document describes the architecture of Physician Portal Application using standard UML and ERD diagrams, including use case, component, package, and deployment diagrams. Readers of this document need to be familiar with the UML and ERD notation.

The below views will be specified in the document:

**Logical view**

* Audience: Designers.
* Area: Functional Requirements: describes the design's object model. Also describes the most important use-case realizations.
* Related Artifacts: Design model

**Process view**

* + Audience: Integrators.
  + Area: Non-functional requirements: describes the design's concurrency and synchronization aspects.
  + Related Artifacts: (no specific artifact).

**Implementation view**

* + Audience: Programmers.
  + Area: Software components: describes the layers and subsystems of the application.
  + Related Artifacts: Implementation model, components

**Deployment view**

* + Audience: Deployment managers.
  + Area: describes the mapping of the software onto the hardware and shows the system's distributed aspects.
  + Related Artifacts: Deployment model.

**Use Case view**

* + Audience: all the stakeholders of the system, including the end-users.
  + Area: describes the set of scenarios and/or use cases that represent some significant, central functionality of the system.
  + Related Artifacts : Use-Case Model, Use-Case documents

**Data view (optional)**

* Audience: Data specialists, Database administrators
* Area: Persistence: describes the architecturally significant persistent elements in the data model
* Related Artifacts: Data model.

**3.**                  **Architectural Goals and Constraints**

There are some crucial requirements and systems constraints that significantly impact the architecture:

* The Physician Portal Application will be deployed into Carroll IIS Server
* The Physician Portal Application development framework is .NET 4.0 and using Microsoft SQL Server 2008.
* Data persistence will be addressed using relational database.
* Query results must be returned within five seconds.
* For application access, there will be different permission criteria for admin access and user access.
* Authentication: Login using at least a user name and a password
* Authorization: according to their profile, online users must be granted or not to perform some specific actions depending whether they are admin, physician, or staff users.

**4.**                  **Use-Case View**

The following use cases are involved in the Physician Portal application.

* UC-1: User Login
* UC-2: View Physician’s Orders
* UC-3: View Patient Results
* UC-4: HL7 Import
* UC-5: View Interface Log
* UC-6: Add Site
* UC-7: Add Test Code
* UC-8: View Report

**UC-1: User Login**

**Use Case Overview**

|  |  |
| --- | --- |
| **Description** | This is the process of an Administrator, Physician or staff level user logging into their account. |
| **Actor(s)** | User, System |
| **Pre-Conditions** | User has already been added to the system. |
| **Post-Conditions** | Success end condition  User will be logged into their account and able to function in the system with their given privileges. |
| **Trigger** | User clicks the log in button |

**Main Flow**

|  |  |
| --- | --- |
| **Main** |  |
| 1 | User enters their user name. |
| 2 | User enters their password. |
| 3 | System validates user name and password. |
| 4 | System displays application home page with either administrator, physician, or staff level user functionality. |

**Alternate Flows**

|  |  |
| --- | --- |
| **Alt 1** | **User not in the system** |
| 1 | System does not find the entered user name. |
| 2 | User will be prompted to submit a request to be set up with a user account. |
| 3 | Use case ends |

**UC-2: View Physician Orders**

**Use Case Overview**

|  |  |
| --- | --- |
| **Description** | This case focuses on searching the database for orders based on physician id. |
| **Actor(s)** | User, System |
| **Pre-Conditions** | User has already been added to the system. |
| **Post-Conditions** | Success end condition  Orders for the logged in Physician will e displayed |
| **Trigger** | User logs in with Physician privileges. |

**Main Flow**

|  |  |
| --- | --- |
| **Main** |  |
| 1 | User logs into the system with physician credentials |
| 2 | System queries database using the id of the logged in physician to select from the order table. |
| 3 | A list of patients with orders by the logged in physician are displayed in the center panel of the web page. |
| 4 | If no match is found in the database the System will display a message indicating that there are no active orders for that physician. |
| 5 | User selects on a single patient row to display all orders for that patient. |
| 6 | System queries database using the id of the selected patient to select from the order table. |
| 7 | A list of orders for the selected patient is displayed below the patient row in the table. |

**UC-3: View Patient Results**

**Use Case Overview**

|  |  |
| --- | --- |
| **Description** | This case focuses on displaying results for a selected patient order |
| **Actor(s)** | User, System |
| **Pre-Conditions** | User is logged into their account with physician credentials.  A successful query has been performed with at least one patient with an order returned as the result. |
| **Post-Conditions** | Success end condition  User is able to view a result report for the selected patient order. |
| **Trigger** | User clicks on an order to view its corresponding result. |

**Main Flow**

|  |  |
| --- | --- |
| **Main** |  |
| 1 | From the physician’s patient list, user selects a patient row to view orders for that patient. |
| 2 | System queries database using the id of the selected patient to select from the order table. |
| 3 | A list of orders for the selected patient is displayed below the patient row in the table. |
| 4 | If the order has results available the order status will display as “Resulted”. |
| 5 | User selects the row of an order that has a result report available to view. |
| 6 | System queries database using the id of the selected order to select from the result table. |
| 7 | The result report is displayed below the selected order in the table. |

**UC-4: HL7 Import**

**Use Case Overview**

|  |  |
| --- | --- |
| **Description** | This case focuses on the system updating the database with data from an HL7 message received through the HL7 Import web service. |
| **Actor(s)** | System |
| **Pre-Conditions** | An HL7 message has been successfully received through the web service. |
| **Post-Conditions** | Success end condition  Database will be updated accordingly with either patient demographic, order, or result information. |
| **Trigger** | Outside entity calls getHL7Message method of HL7 Import web service. |

**Main Flow**

|  |  |
| --- | --- |
| **Main** |  |
| 1 | System parses first line of string message that was received to determine if it is a valid HL7 message. |
| 2 | If the message is not a valid HL7 message then system writes an entry to the interfacelog table to indicate that an invalid message was received. System takes no further action on the message. |
| 3 | If the string is a valid HL7 message, system will further parse the first line to determine if the message contains patient demographic, order, or result information. |
| 3 | If the message contains patient demographic information, the message will be passed to the ADT processor. |
| 4 | If the message contains order information, the message will be passed to the order processor. |
| 5 | If the system contains result information, the message will be passed to the result processor. |
| 6 | System will parse and process the message based on the respective ADT, order, and result rules and update the patient, order, and result tables accordingly |
| 7 | System updates the interfacelog table with the result of the message parsing and table updates. |

**UC-5: View Interface Log**

**Use Case Overview**

|  |  |
| --- | --- |
| **Description** | This case focuses on a user viewing the interface log. |
| **Actor(s)** | User, System |
| **Pre-Conditions** | User is logged into their account.  User is logged in with appropriate privileges to view the interface log. |
| **Post-Conditions** | Success end condition  The interface log is displayed. |
| **Trigger** | User selects the option to view the interface log |

**Main Flow**

|  |  |
| --- | --- |
| **Main** |  |
| 1 | User selects the interface log tab from the menu. |
| 2 | System queries database for interface log information. |
| 3 | Interface log is displayed in the center panel of the web page. |

**UC-6: Add Site**

**Use Case Overview**

|  |  |
| --- | --- |
| **Description** | This case focuses on a user adding a new site |
| **Actor(s)** | User, System |
| **Pre-Conditions** | User is logged into their account.  User is logged in with an administrator account. |
| **Post-Conditions** | Success end condition  A new site is added. |
| **Trigger** | User selects Add New Site option. |

**Main Flow**

|  |  |
| --- | --- |
| **Main** |  |
| 1 | User types new site ID into the Site ID field.. |
| 2 | User types in a site name into the Site Name field. |
| 3 | Optional: User fills in the address, city, state, and zip field. |
| 4 | User clicks submit button. |
| 5 | System verifies that site id is unique. |
| 6 | If the site ID is unique, system adds new site to the database. |
| 7 | System displays a message indicating that the new site has been successfully added. |

**Alternate Flows**

|  |  |
| --- | --- |
| **Alt 1** | **Site ID already exists in database** |
| 1 | System prompts user that the entered ID already exists and a new unique site ID must be entered. |
| 2 | Use case resumes at step 1. |

**UC-7: Add Test Code**

**Use Case Overview**

|  |  |
| --- | --- |
| **Description** | This case focuses on a user adding a new test code |
| **Actor(s)** | User, System |
| **Pre-Conditions** | User is logged into their account.  User is logged in with an administrator account. |
| **Post-Conditions** | Success end condition  A new test code is added. |
| **Trigger** | User selects Add New Test Code option. |

**Main Flow**

|  |  |
| --- | --- |
| **Main** |  |
| 1 | User types new test code into the Test Code field. |
| 2 | User types in a description into the Description field. |
| 4 | User clicks submit button. |
| 5 | System verifies that the test code is unique. |
| 6 | If the test code is unique, system adds new test code to the database. |
| 7 | System displays a message indicating that the new test code has been successfully added. |

**Alternate Flows**

|  |  |
| --- | --- |
| **Alt 1** | **Test Code already exists in database** |
| 1 | System prompts user that the entered Test Code already exists and a new unique test code must be entered. |
| 2 | Use case resumes at step 1. |

**UC-7: View Report**

**Use Case Overview**

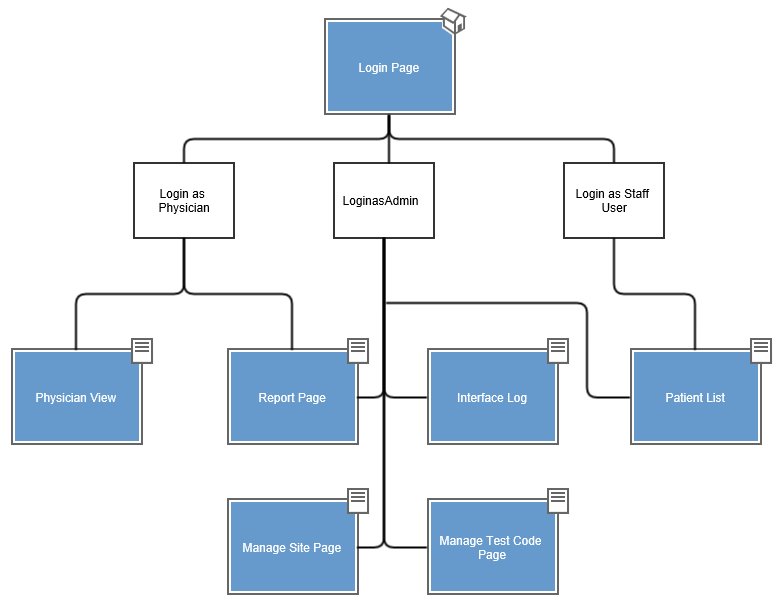
|  |  |
| --- | --- |
| **Description** | This case focuses on a user viewing a report. |
| **Actor(s)** | User, System |
| **Pre-Conditions** | User is logged into their account.  User is logged in with an administrator or physician account. |
| **Post-Conditions** | Success end condition  The report is displayed. |
| **Trigger** | User selects the option to view a report. |

**Main Flow**

|  |  |
| --- | --- |
| **Main** |  |
| 1 | User selects one of the available reports.  The following reports are available:  Physician Activity.  Test Code Usage. |
| 2 | System queries database for report information. |
| 3 | System builds report. |
| 4 | Report is displayed in the center panel of the web page. |
| 5 | Optional: User may choose to export the report by clicking the export button. System then transfers file to user’s local machine using HTTP. |

**5.**                  **Logical View**

**5.1**               **Overview**

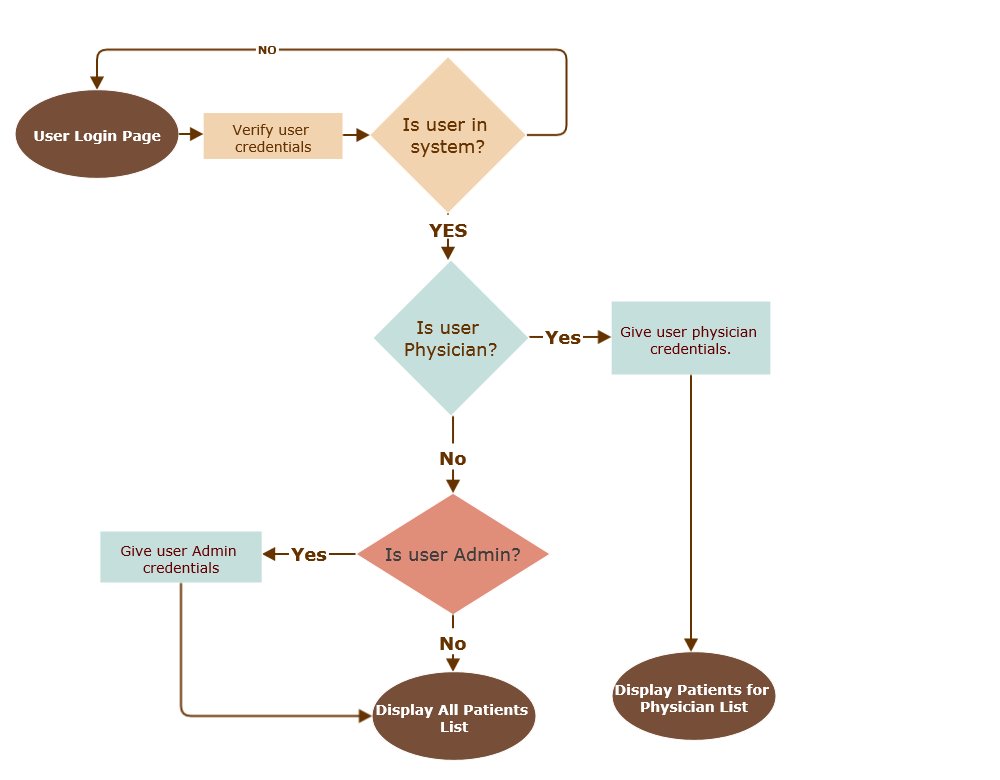
****

**5.2**               **Architecturally Significant Design Considerations**

|  |  |
| --- | --- |
| DataAccess.cs | This class provides a single repository for all of the business logic. It provides the LINQ queries that access the data layer. The DataAccess class is referenced in the view classes, providing a layer of separation between view and data. |
| HL7ImportWebService.asmx | Web service that exposes a method which allows the application to receive messages from external systems. |
| IParser.cs | Interface that provides the contract for parsing of the messages being imported into the application. This layer of abstraction provides the application to import messages of different formats. |
| HL7Parserv2.cs | This class implements the IParser and provides the specific logic needed for the Physician Portal application to read HL7 messages and allow it to integrate in the health care world. |
| AccountMembershipProvider.cs and AccountRoleProvider.cs | These two classes are overridden to provide custom code that will allow the .NET security features to make use of the external Physician Portal SQL database so the security rules will be driven by the design in that database. |

**6.**                  **Process View**

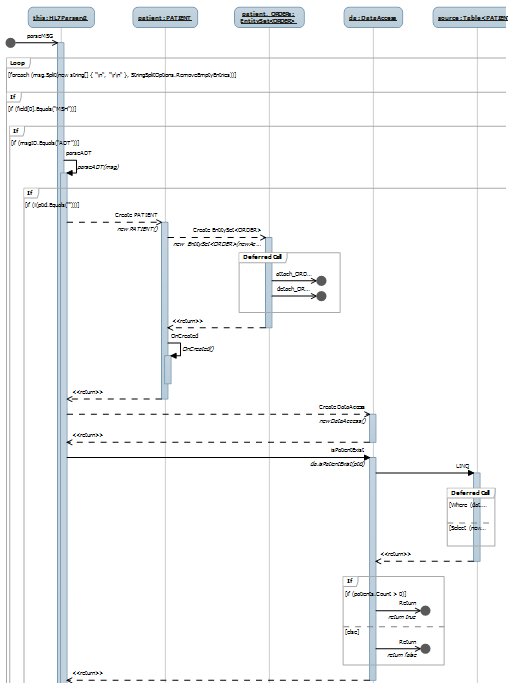
This section consists of three process diagrams. The first is a flow chart representation of the user login process. The second is a sequence diagram representation of the user login process. The third is a sequence diagram representation of the HL7 parsing process.

****

Login Flow Chart



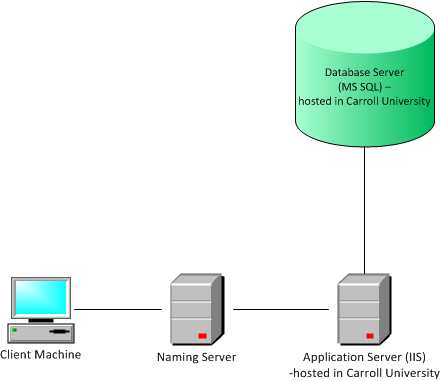
Login Sequence Diagram



HL7 Parser Sequence Diagram

**7.**                  **Deployment View**

Both the Physician Portal application website and the Physician Portal database will be hosted in Carroll University servers. Clients will have access to website using any browser in their personal computer.



**8.**                  **Implementation View**

**8.1**               **Overview**

Although MVC isn’t used in the Physician Portal application, it was designed to provide the same separation between the view, business logic, and data access layers. This design will allow for seamless changes in one layer that will not impact the code in the other layers.

**8.2**               **Layers**

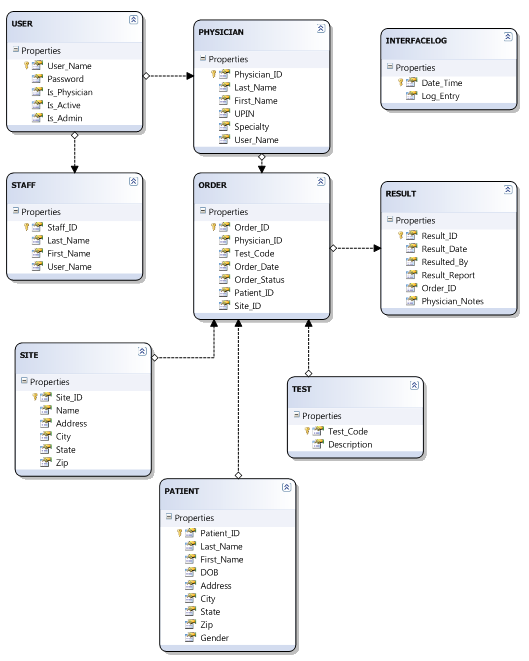
**View Layer:**

This layer provides interfaces with users through different .aspx pages. Through the interface, users input are processed by back-end C# codes behind. To communicate with the database, this layer use different model classes written in C#.

**Logic Layer:**This layer connects view layer with data source layer. More specifically, it processes, user inputs, get access to data source, and manipulate the data based on user input, and output results to the View layer. Linq was used to create a collection classes, each class represents a table in the Microsoft SQL database. The querying brain is DataAccess.cs. Back-end codes in View Layer use this class to query information from the Linq Tables.

**Data Access Layer:**This layer is represented by Linqtoalldata.dbml. The physical connection to the SQL server database is managed by LINQ.

**9.**                  **Data View**



**10.**             **Size and Performance**

During the first deployment, it will be available to admin users for testing and uploading data. After that, it will be accessible to a wider audience for further integration testing. Page response times will be less than 10 seconds and ideally should be returned within a second. The system will be backed up on in timely manner.

**11.**             **Quality**

Only registered physician, admin, and staff users can access the system. The application will be available 24/7. Users will be notified for scheduled downtime for backup and maintenance. Support will be contacted if admin requires additional information about the application. . Any changes done within the architecture will have less/none effect on other layers of the architecture thus maintaining high portability of the system.