|  |
| --- |
|  |

|  |
| --- |
| **Architecture Design** |
| Human Resource Management Project |
|  |
| This document describes the architecture design for Human Resource Management (HRM) project. In this document, the view, including static view, physical view, and dynamic view will be shown. In addition, the data model will be described in this. |
|  |
| **KIM TUONG** |
| **[Pick the date]** |
|  |

Table of Contents

**No table of contents entries found.**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Description** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Documentation roadmap

# Scope

This document covers following:

* Client’s vision, project goals, context, and constraints of the system being developed
* Architectural drivers (high level functional requirements, quality attributes, and constraints) and prioritized utility tree
* Architectural views
* Architectural decisions
* Tradeoff analysis

The specification is intended for the following audience:

* Stakeholders for the Human Resource Management(GSD) project
  + - * Team HRM-PIM
* Van Lang’s IT Faculty
  + - * Audience that might want to get an insight into or analyze the architecture and the design of the HRM project

# 1.2 Document organization:

This document includes three important parts:

**Part 1:** How to read this document

**Part 2:** System overview- The summarize of architectural drivers (High level requirement, quality attribute and constraint)

**Part 3:** View

*Physical* – How software and hardware interact with each other.

*Static* – The module of HRM system.

*Dynamic* – The interaction between HRM components.

The structure for presenting the view

**Session 1**: Primary presentation- The figure to present the view

**Session 2**: Element catalog – The table for describing the element which present in figure

**Session 3**: Element behavior- The flow of each component (This part is just for Dynamic perspective)

**Session 4:** Architecture background- Includes design decision and reasons for designing

1. System overview:

# Project Overview

**Project name:** Human Resource Management Program

**Clients:** Human Resource Department- Van Lang University

**Mentors:** Quang Nguyen (Van Lang University)

**Developer team:**

* Nhung Huynh- Team leader
* Tan Tran
* Tuong Nguyen
* Nguyen Dinh
* Dang Nguyen
* Loc Phan
* Quyet Nguyen
* Tung Nguyen

# Project Background

Human Resource Department in Van Lang University gets used to manage personal information in MS Excel. Now they need a new tool that supports only for managing human resource in Van Lang.

The vision of the project is to develop personal information management. In the future, the system will develop other function related to payroll, insurance information…

# **Project Goal**

HRM is particularly developed for human resource management in university colleges. The system consists of key modules:

* Personal information management
* Employee labor contract management
* Recruitment & training processing
* Payroll
* Administration panel – Utilities

1. System Context:

|  |  |
| --- | --- |
| **Actor** | **Description** |
| Educated Department |  |
| Administrator |  |
| Salary Group |  |
| Department |  |
| HR Group |  |
| Account Department |  |

*Figure 2: System Context of HRM project*

1. Architecture drivers:

*See more in Architecture Driver Document*

1. Architecture Style/Patterns & Tactics:

This section discusses the architectural styles and patterns that the HRM architecture possesses. It also provides rationale for selection of these styles.

# Styles/ Pattern

We identify that the style that will be used in HRM- PIM (Personal Information Management) project will be client and server style based on the architecture analyzing. Following is the rationale for choosing client and server as the style for the HRM- PIM project:

* Personal information data needs to be shared with all authorized users. Users can access these data from globally distributed locations and number of users may range from 10 to 50. In order to achieve these requirements and qualities, we decided to have a PIM server that would store and manage data.
* HRM project is web application, all user will access via Web browser with Silverlight runtime is available as a [plug-in](http://en.wikipedia.org/wiki/Plug-in_(computing)) for [web browsers](http://en.wikipedia.org/wiki/Web_browser). The PIM server will provide service based on the requests from client.

**Tiered Architecture:**

In more detail, we decided to use tiered architecture to separate the client, server and database to enhance the security and modifiability. The first tier consists of PIM client. The second tier consists of business services. The third tier provides data management services which using Entity Framework. We will delve into details in later sections.



**Figure 5.1.1 Client and server style for HRM-PIM application**

In this solution, there is a PIM server that responsible for providing the appropriate services that are invoke by the client. The PIM server will also store the data object mapped with Database server and The PIM client will invokes service for manage the personal information such as read and write data in to database, report, import, export data…The user will uses Web browser to access to the PIM Client.

# Tactics used

1. **Modifiability**

*Localization of changes*

We grouped together the components that we anticipated to be affected by the similar kind of changes. E.g. we grouped the view components together; view-model components together… We tried to generalize these modules based on their functions.

*Separation of concern*

Based on the functional requirements, we categorized elements into different sets of components so that the developers can add new components or delete components without affecting the other components.

*Defer binding time.*

The application allows the end user or system administrator to make settings or provide input that affects behavior. E.g. The system administrator can change role permission or assign role in configuration file.

1. **Performance**

*Maintain multiple copies of either data or computations*

We have decided to use caching of data in server level by using Entity Framework. With Entity Framework, we can eliminate the direct data accessing to database which require amount of time for each transaction.

1. **Usability**

**Long running operations**

As most of the operations in the HRM- PIM are going to take time, we are going to design the UI components such that they do not halt when certain long running operations are in progress. We will design asynchronous invocations for such operations by using WCF- RIA services

**Maintain a model of the task**

The model of the task to be performed would be maintained to determine context, such that HRM-PIM system can have an idea of what the user is attempting and can provide assistance. For example, the users would not need to re-enter the same data again, once the data is obtained for the first time, the related information like project name, its components, resources etc, would be pre-populated. This would save time and resources at the users end to perform a task.

**Design time tactics**

Separating user from the rest of the application: we are using Model View View-Model (MVVM) pattern. E.g. UI views form the view, each views will have the corresponding view-models and the models will consists the action in UI controls

1. Architecture Overview:

# Component and connector (C& C) views of HRM- PIM system

# Primary Presentation



**Figure 6.1.1 High level C&C view of the HRM-PIM system**

The figure 5.1.1 depicts the client and server style. PIM client is a client of PIM server and Database Server is system database.

# Element catalog

Following table describes responsibilities of the different element

|  |  |
| --- | --- |
| **Component** | **Responsibility** |
| PIM Server | This component run contains all service components, business objects and data persistence mapped with PIM Database server. PIM server will provide the appropriate services whenever they are invoked by client. These services will be the business service of Personal Information Management and Authentication Service. These service components will be descripted later in Server decomposition.  PIM Server also contains data objects mapped with database in data server. These objects use Entity Framework |
| PIM Client | PIM client runs in Silverlight platform. It invokes the service from PIM server. It can read the data file and update the database on the PIM database server. PIM client also performs authentication of the user. |
| PIM Database Server | This component is SQL Database Server of HRM- PIM system. It contains data that will be used in all system. |
| Web browser | PIM client will be accessed by user via this component. Web browser must be available the Silverlight plug-in. |
| Configuration file | This file contains the Web configuration (host information) for running service. |
| Data files | Data files are the files that need to be imported in Database server. E.g. The import service can only import data in Excel files. |

Following table describes connector used

|  |  |
| --- | --- |
| **Connector** | **Purpose** |
| WebServiceCallReturnConn | This connector represents a web service call made by a caller to a callee. The information is transferred over http(s) connection. |
| FileReadWriteConnT | This connector allows a “user” role to read from or write to a disk file. |
| Server-DatabaseConnT | This connector represents a connection between the server and the database server. In fact, this is the connection between the data object and database by using the ADO.NET Entity Framework |

# C&C Views of PIM Client

# Primary Presentation



**Figure 6.1.2 High level C&C view of the PIM Client**

# Element catalog

Following table describes responsibilities of the different element

|  |  |
| --- | --- |
| **Component** | **Responsibility** |
| Personal Information Manager | This component is responsible for manage the personal information by calling the service. Personal Information includes:   * Data management (add, edit, delete, filter, view… the personal information) * Import management: read the data file and upload into database * Export management: write data into data file with specific format * Report management: create the report based on the inputted information from user |
| Catalog Manager | This component is responsible for manage all catalog that will be used in system. This component allows user edit the system catalog and write to PIM database server by calling the service in PIM Server |
| Login Manager | This component is responsible for validate the user name and password when user inputs them to login. This component will call the authentication service to validation. This component is responsible for login out of system. |
| Client Data | This component acts like a bridge between Personal Information Manager, Catalog Manager, Login Manager components and the PIM server. These components just call Client Data component and this component will calling the service on server instead of call the service by themselves. |

Following table describes connector used

|  |  |
| --- | --- |
| **Connector** | **Purpose** |
| WebServiceCallReturnConn | This connector represents a web service call made by a caller to a callee. The information is transferred over http(s) connection. |
| FileReadWriteConnT | This connector allows a “user” role to read from or write to a disk file. |
| Server-DatabaseConnT | This connector represents a connection between the server and the database server. In fact, this is the connection between the data object and database by using the ADO.NET Entity Framework |
| CallReturnConnT | Caller calls a certain method of callee and callee performs the requested operation and returns the result back to the caller. |

# C&C views of Personal Information Manager Component

# Primary Presentation



**Figure 6.1.3 C&C view of the Personal Information Manager component**

The fig 6.1.3 depicts the decomposition of component Personal Information Manager. The rationale for this decomposition is to further divide this component into four groups of component by separating the concerns based on high-level functional requirement. It consists some UI views that user interacts with by using web browser. The user can choose to view the personal information (detail and extend information). The authorized users can edit, delete, filter, export and import the personal information data into system. The UI events and properties will be identified and binding in View-Models and the Model will responsible for implement the user action.

# Element catalog

Following table describes responsibilities of the different element

|  |  |
| --- | --- |
| **Component** | **Responsibility** |
| Data Manager Views | This is view components and responsible for displaying screens in PIM client. It consist the user controls for data management of personal information including detail and extend information. These views primarily include the grid for showing the data and the edit screen for update or add new data. |
| Import Manager Views | This is view components and responsible for displaying screens in PIM client. It consist the user controls for import management. These views primarily include the screen with button for user to choose the data files need to import and the grid for showing the imported data that will be upload to database server. |
| Export Manager Views | This is view components and responsible for displaying screens in PIM client. It consist the user controls for export management. User will choose the data grid that needs to be export to Excel file. |
| Report Manager Views | This is view components and responsible for displaying screens in PIM client. It consist the user controls for report management. These views primarily include the field for user to input the information need to report and condition. The report result will be showed on new screen window. |
| Data Manager View-Models | This component is the root component. Each view will has corresponding view-model. Whenever a property on a View-Model object has a new value, it can raise the PropertyChanged event to notify the binding system of the new value. Upon receiving that notification, the binding system will bound properties on Data Manager View. This component responsible for implement the user action and properties from Views. The actions need to implement in data management are:   * Add data * Edit data * Delete data * Refresh data * Sort data * Print data * Save data   This component is also check the permission of user for showing the hide button based the user setting (Check to ensure that the users are allowed to implement the action) |
| Import Manager View-Models | This component is the root component. Each view will has corresponding view-model. This component responsible for implement the user action and properties from Import Manager Views. The primarily actions need to implement in import management are:   * Open file * Import data (Save data)   This component is also check the permission of user for showing the hide button based the user setting (Check to ensure that the users are allowed to implement the action) |
| Export Manager View-Models | This component is the root component. Each view will has corresponding view-model. This component responsible for implement the user action and properties from Export Manager Views. The primarily action need to implement in import management are:   * Export data (Save into file)   This component is also check the permission of user for showing the hide button based the user setting (Check to ensure that the users are allowed to implement the action) |
| Report Manager View-Models | This component is the root component. Each view will has corresponding view-model. This component responsible for implement the user action and properties from Report Manager Views. The primarily action need to implement in import management are:   * Report data   This component is also check the permission of user for showing the hide button based the user setting (Check to ensure that the users are allowed to implement the action) |
| Authentication Model | This model is responsible for loading, checking the permission of user and raising the event to all View-Models to notify and show the hide buttons on views. This component will be call at first when the view is loading. |
| Personal Information Model | The model component has business logic or data validation for implement the action in View-Model. It also responsible for raise the event complete when the action complete. This component is used by four View-Models (Data Manager, Import, Export, Report Manager). |

Following table describes connector used

|  |  |
| --- | --- |
| **Connector** | **Purpose** |
| FileReadWriteConnT | This connector allows a “user” role to read from or write to a disk file. |
| CallReturnConnT | Caller calls a certain method of callee and callee performs the requested operation and returns the result back to the caller. |

# C&C views of Personal Catalog Manager Component

# Primary Presentation



**Figure 6.1.4 C&C view of the Catalog Manager component**

The fig 6.1.4 depicts the decomposition of component Catalog Manager. It consists some UI views that user interacts with by using web browser. The user can choose to view the catalog data. The authorized users can edit, delete, filter catalog data into system. The UI events and properties will be identified and binding in View-Models and the Model will responsible for implement the user action.

# Element catalog

Following table describes responsibilities of the different element

|  |  |
| --- | --- |
| **Component** | **Responsibility** |
| Catalog Data Manager Views | This is view components and responsible for displaying screens in PIM client. It consist the user controls for catalog management. These views primarily include the grid for showing the catalog data and the edit screen for update or add new catalog data |
| Catalog Data Manager View-Models | This component is the root component. Each view will has corresponding view-model. Whenever a property on a View-Model object has a new value, it can raise the PropertyChanged event to notify the binding system of the new value. Upon receiving that notification, the binding system will bound properties on Catalog Manager View. This component responsible for implement the user action and properties from Views. The actions need to implement in catalog management are:   * Add data * Edit data * Delete data * Refresh data * Sort data * Print data * Save data |
| Catalog Model | The model component has business logic for implement the action in View-Model. It also responsible for raise the event complete when the action complete. The common business logic that this component responsible for:   * Delete catalog data * Save catalog data * Get catalog data by key * Get catalog data |

Following table describes connector used

|  |  |
| --- | --- |
| **Connector** | **Purpose** |
| CallReturnConnT | Caller calls a certain method of callee and callee performs the requested operation and returns the result back to the caller. |

# C&C views of Personal PIM Server

# Primary Presentation



**Figure 6.1.5 C&C view of the PIM Server**

The fig 6.1.5 depicts the internals of PIM server. It includes the WCF-RIA services that will be invoked by PIM client. These services will uses the business objects and the business object will call to data objects which uses Entity Framework for design an data model mapped with database server.

# Element catalog

Following table describes responsibilities of the different element

|  |  |
| --- | --- |
| **Component** | **Responsibility** |
| PIM Business Service | This is WCF- RIA service component and hosted on the server. This component provides main services that used in PIM system. It primarily includes Data Manager Service, Export service and Import service. |
| Authentication service | This is WCF- RIA service component and hosted on the server. This component provides services that are invoked by Login Manager in PIM Client. The service is focus mainly on Login and Logout service. |
| Report service | This is WCF- RIA service component and hosted on the server. This component provides services that are invoked by Report Manager. |
| PIM Business Objects | This component contains the attribute of PIM entity with their behavior. E.g. The business object “Employee” with attribute Employee Name, Birthday, Division, Job Title…and the behavior is get and set. |
| PIM Data Objects | This component run in Entity Framework that is the data model mapped which the HRM-PIM Database. The business cannot access directly to database, and have to access through Data Object for reading and writing data. Entity Framework also support the lazy loading. |

Following table describes connector used

|  |  |
| --- | --- |
| **Connector** | **Purpose** |
| WebServiceCallReturnConn | This connector represents a web service call made by a caller to a callee. The information is transferred over http(s) connection. |
| FileReadWriteConnT | This connector allows a “user” role to read from or write to a disk file. |
| Server-DatabaseConnT | This connector represents a connection between the server and the database server. In fact, this is the connection between the data object and database by using the ADO.NET Entity Framework |
| CallReturnConnT | Caller calls a certain method of callee and callee performs the requested operation and returns the result back to the caller. |

# Element behavior

This section depicts the behavior diagram of PIM Data Manager and Import manager to visualize how action of components and how the client and server interact with each other. All the description of component in the following diagram is specified in previous section.

 **Figure 6.1.6 Element behavior of the PIM system- Data management**

 **Figure 6.1.6 Element behavior of the PIM system- Import management**

# Architecture background

# Architecture decision

**Architecture decision #1- AD1**

We have decided to have the server to share the data among the users (N-tier architecture). The number of user can be up to 50 transactions at a time so that have the server will ensure that the user can access to system in everywhere. Especially, with the Manage Profile function, the user can access to system at home to update their profile. We also decide to have a database server separates with the PIM server.

*Trade-off analysis*

**Security:** This decision promotes the attribute security of system because the N-tier separate user with the server and database server so that it will reduce the attack from unauthorized users. The user will not know the location of server and database server.

**Performance:** This decision may inhabit the performance scenario (QAS.02). The PIM client must connect to PIM server and invoke services so that the response time will be increased. It depends on the server computation and the network bandwidth.

**Architecture decision #2- AD2**

We decide to use MVVM pattern in the client tier. There are three component groups View, View-Model, Model. Each group will have different responsibility.

*Trade-off analysis*

**Modifiability:** MVVMsupport the tactic “Localize the changes” so that it promotes the modifiability. All the components that have the same effect by a change will be grouped together so that developer can modify the system without affect the other components. E.g. The modifying the business logic in Model components will not cause effect in the View component and vice versa

**Performance:** The decision also promotes performance but it not clearly. The View components do not process anything. All processing is happened in View-Model and Model so that response time will be improved.

# Architecture rationale

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Quality** | **Quality ID** | | **Concern** | **Response measure** |
| Performance |  | |  |  |
| Rationale: | | | |
| Modifiability |  |  | |  |
|  | Rationale: | | | |