

|  |  |
| --- | --- |
|  |  |

**Design Document**

**OOPMS**

**Friday, June 08, 2012**

**Version 0.1**

*Prepared by*

**Ngo Duc Duy**

Revision and Signoff Sheet

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Author** | **Version** | **Change reference** |
| 06/05/2012 | Ngo Duc Duy | 0.1 | Create new |
|  |  |  |  |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Version** | **Position** | **Date** |
| Manh Hoang Truong | 0.1 | Developer | 12-June-2012 |
| To Cong Thanh Hai | 0.1 | Developer | 12-June-2012 |
| Pham Nguyen Truong Giang | 0.1 | Developer | 12-June-2012 |
| Le Ngoc Thach | 0.1 | Project Manager | 12-June-2012 |

Table of Contents

[1 Introduction 30](#_Toc304151916)

[1.1 Purpose 30](#_Toc304151917)

[1.2 Scope 30](#_Toc304151918)

[1.3 Intended Audiences and Document Organization 30](#_Toc304151919)

[1.4 Acronyms and Abbreviations 30](#_Toc304151920)

[1.5 References 31](#_Toc304151921)

[2 Architecture design 31](#_Toc304151922)

[2.1 Overview 31](#_Toc304151923)

[2.2 Application Logical Architecture 31](#_Toc304151924)

[2.1 Component view 33](#_Toc304151925)

[3 Database design 33](#_Toc304151926)

[3.1 Entity Relationship Diagram 33](#_Toc304151927)

[3.2 Schema 34](#_Toc304151928)

[3.1 Detail Schema 34](#_Toc304151929)

[4 Application Security 34](#_Toc304151930)

[4.1 Custom Permission 34](#_Toc304151931)

[4.2 Security Group 34](#_Toc304151932)

[4.3 Security Group v.s. List 35](#_Toc304151933)

[5 Workflow and Business Processing 31](#_Toc304151934)

[5.1.1 PM Request 32](#_Toc304151935)

[6 Interface Design 35](#_Toc304151936)

[6.1 Layout 35](#_Toc304151937)

[6.2 Themes 35](#_Toc304151938)

[7 Details design 35](#_Toc304151939)

[8 Configuration 36](#_Toc304151940)

[8.1 Application Configuration 36](#_Toc304151941)

[8.2 System Configuration 36](#_Toc304151942)

[9 Packaging and Deployment 36](#_Toc304151943)

# Introduction

## Purpose

This document contains the detailed design for to be developed application on target platform. It defines, technically, how applications will operate. Developers will base on this document and corresponding SRS to conduct development plan, task assignment and implementation of the new application.

## Scope

This document is prepared for the application OOPMS in scope of the capstone project of FU K4B.

## Intended Audiences and Document Organization

This document is intended for:

* Development team: Developers and Testers
* Rollout Technical Team: Responsible for deploying applications to UAT and Production environments.
* Customer Representatives: Responsible to review & approve the document.

Below are main sections of the document:

* **Introduction** : This section describes the general introduction of this document
* **Architecture Design :** This section describes the high-level technical assessments and decisions for the application.
* **Technical Solutions :** This section describes mechanism used in the project.
* **Data Design**: This section describesin detail how data is structured and manipulated in this application.
* **Interface Design:** This section describesin detail how UI is designed in general ( layout , theme ).
* **Application Security**: This section describles security matrix in detail
* **Configuration:** This section describes all configuration needed for the application to function properly.
* **Packaging and Deployment:** This section describles how applications could be packaged and deployed.
* Note:Please refer section 1.4 for all acronyms and abbreviations you may encounter within this document.

## Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| # | Item | Description |
| 1 | ASP | ActiveX Server Page |
| 2 | JVM | Java Virtual Machine |
| 3 | HTTP | Hypertext-Transfer Protocol |
| 4 | MVC | Model – View – Control |
| 5 | DAO | Data Access Object, this object is responsible for attaching to a system, extracting some information, based on specific requirements, and creating a value object. |
| 6 | OOPMS | Open-One Project Management System |

## References

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Title | Version | File Name / Link | Description |
| 1 | SRS Document | 1.0 |  |  |
| 2 | User Requirement | 1.0 |  |  |

Table 1.1: List of References

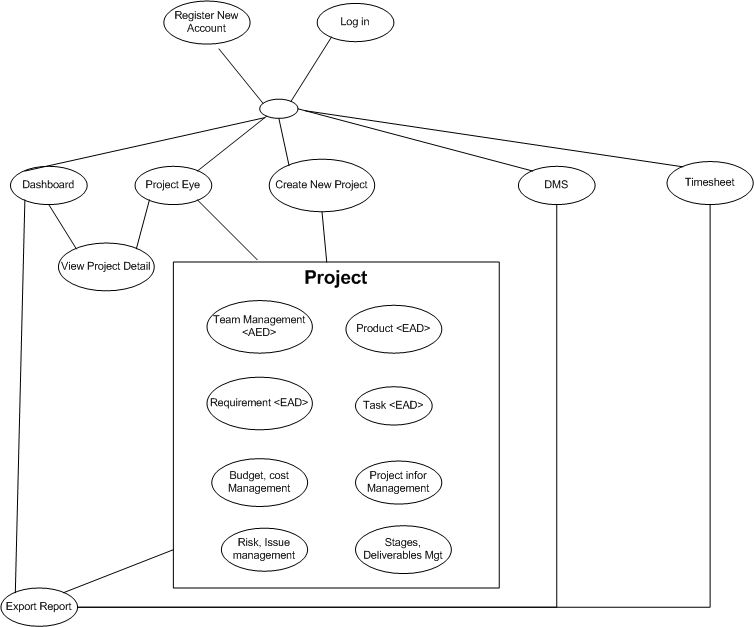
# Architecture design

## User case view

**Table of Use Case**

### Overview

Main flow of Use Case



## Architectural Representation

The following diagram shows the primary tiers in the proposed n-tier architecture. This diagram shows the main layers in this architecture and the vision of how they fit together.



Figure 1 – N-tier architecture of SD System

### Presentation Layer

This layer controls the display to the end user. For the presentation layer of OOPMS, the development framework is based on MVC Model architecture. The framework is responsible for:

Managing requests/responses from/to the clients.

Controlling display to the end user.

Assembling a model that can be presented in a view.

Performing UI validation.

Providing a controller to delegate calls to business logic and other upstream processes.

Handling exceptions from other layers.

### Business Layer

This layer manages the business processing rules and logic.

Handling application business logic and business validation.

Managing transactions.

Allowing interfaces for interaction with other layers.

Managing dependencies between business level objects.

Adding flexibility between the presentation and the persistence layer so they do not directly communicate with each other.

Exposing a context to the business layer from the presentation layer to obtain business services.

Managing implementations from the business logic to the persistence layer.

### Data Access Layer

This layer manages access to persistent storage. The primary reason to separate data access from the rest of the application is that it is easier to switch data sources and share Data Access Objects (DAOs) between applications.

This layer manages reading, writing, updating, and deleting stored data.

### Data Layer

In OOPMS, the storage is managed by a relational database. Oracle 10g Express is used for this layer to provide the management of stored data.

## 2.3 Packages/Components view



### UI Components

This package includes the implementation for the JSP, MVC architecture proposed to be used in the Presentation Layer to handle the display to the end user.

**Validation**: All validation of incoming requests parameters to the server should be validated using JavaScript Validation or JSP client side control .

### Business Object

This package includes the implementation of business objects. **Business Object** (BO) layer is used to perform the business operations. The Business Object layer will access the DAO to access database. Transactions should be managed within this business layer.

### Transfer Data Objects ( Entity )

Transfer Data Objects is java class, contains lightweight structures for related business information. These are sometimes referred to as data transfer objects. A value object (VO) is a lightweight, serializable object that structures groups of data items into a single logical construct. .In addition, VOs are useful in communication among all layers of the application.

### Data Access Object

This package includes the implementation of Data Access Object. Using Oracle CLient object here to make the application more flexible to access database. Oracle Client object includes basic functions to work with database: *select, insert, update, delete*.

### Exceptions

This package will include all general exceptions that will typically used by more than one package. The try-catch clauses should be kept to a minimum.

### Utils

This package includes all utilities will be wisely used in the modules.

### Logging

This package includes implemented logging classes.

# Technical Solutions

## Exception handling mechanism

The try-catch clauses should be kept to a minimum.

Whenever an exception raise , system will log exception detail to Windows event and redirect user to Error page to display standard error message!

## Logging mechanism

Logging is an important and pretty useful mechanism for every application. It can help developers to debug and improve their code or test it’s functionality. In OOPMS, logging component is developed based on log4j logging API.

# Database design

## Entity Relationship Diagram

## Schema

**Overview**

|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Description** |
| 1 | Request | Store all request |
| 2 | CommentAttachs | Store comments and attach files in request |

## Detail Schema

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Category table** | | | | |
|  | **Field name** | **Type** | **Max Length** | **Descrition** |
| 1 | categoryID | String |  | Mã thể loại |
| 2 | categoryName | String |  | Tên thể loại |

# Application Security

## User Permission

|  |  |
| --- | --- |
| **Name** | **Permission** |
| Requester | New, Read , Edit |
| Reader | Read |
| Full Control | New, Read, Edit,Delete |
| No Acess | N/A |

## Security Group

|  |  |
| --- | --- |
| Name | **Description** |
| [Setup.Admin] | Administrator is responsible to manage request data, manage application config, report  and perform importing functions, etc. |
| [Requester] | A Requester who manages request, submits request for approval. |
| [Application.Admin] | A group of users who has special permission to process all requests in any step of application workflow |

## . Main functions ‘s security matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Usercase** | **Actor** | **Admin** | **Log**  **tracker** | **Biên tập viên** | **Tổng biên tập** | **Người dùng** | **System** |
| Quản lý phim | Thêm mới phim | X |  | X | X |  |  |
| Cật nhập phim | X |  | X | X |  |  |
| Xóa phim | X |  | X | X |  |  |
| Duyệt phim | X |  |  | X |  |  |
| Quản lý thể loại fim | Danh sách thể loại | X |  |  |  |  |  |
| Thêm mới thể loại | X |  |  |  |  |  |
| Cập nhật thể loại | X |  |  |  |  |  |
| Xóa thể loại | X |  |  |  |  |  |
| Quản lý tài khoản | Danh sách tài khoản | X |  |  |  |  |  |
| Thêm mới tài khoản | X |  |  |  |  |  |
| Cập nhật tài khoản | X |  |  |  |  |  |
| Xóa tài khoản | X |  |  |  |  |  |
| Quản lý khách hàng | Thêm mới khách hàng |  |  |  |  |  | X |
| Cập nhật khách hàng |  |  |  |  |  | X |
| Xóa khách hàng |  |  |  |  |  | X |

# Details function design

## Use case 01: Logon

### Screen design & Data description

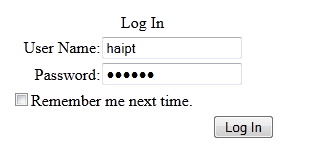
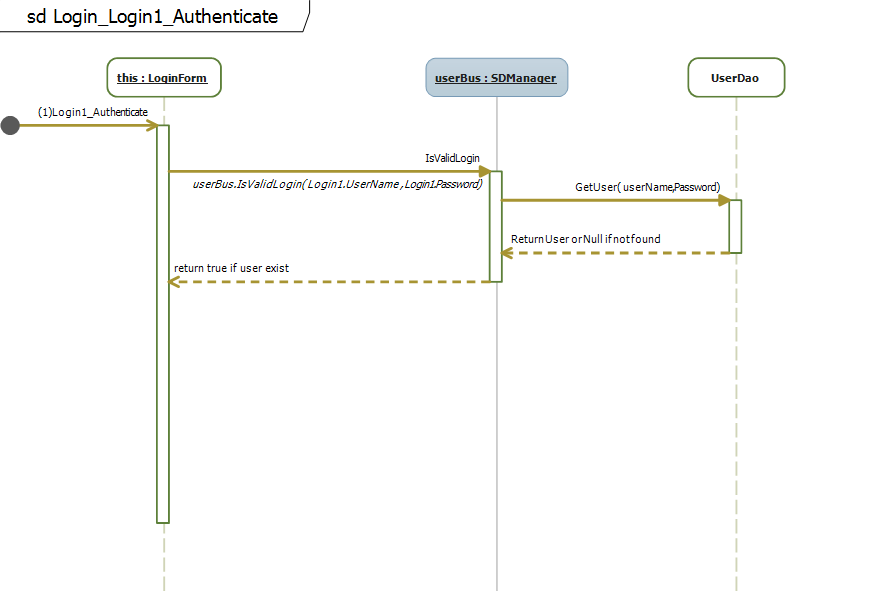


Figure 2 : Login Screen

### Class Diagram



### Sequense flow



## Use case X: Logout

### Screen design & Data description

### Class Diagram

### Sequense flow

# Interface Design

# Configuration

# Packaging and Deployment

# Appendix