|  |
| --- |
| Description: E:\My Documents\Desktop\Logo_FPT_University_doc.jpg |
| Report 06: Final Report |
|  |
|  |
|  |
| |  |  | | --- | --- | | **OOPMS Team** | | | **Group Members** | Ngô Đức Duy – 60163  Phạm Nguyễn Trường Giang – 60132  Tô Công Thanh Hải – 60140  Mạnh Hoàng Trương – 60003 | | **Supervisor** | Teacher Lê Ngọc Thạch | | **Project Code** | OOPMS | | **Document Code** | OOPMS\_FinalReport\_v1.0 | |

**Ho Chi Minh City, 10th Aug, 2012**

**Record of Changes**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Change Item** | **Description** | **By** | **Version** |
| 20/07/2012 | Create outline | Create the outline document | DuyND | 0.1 |
| 25/07/2012 | All | Update | DuyND | 0.2 |
| 31/07/2012 | All | Update | DuyND | 0.3 |
| 01/08/2012 | All | Update | DuyND | 0.4 |
| 08/08/2012 | All | Update | DuyND | 0.5 |
| 09/08/2012 | All | Update | DuyND | 0.6 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Table of Contents

[Table of Contents 170](#_Toc332351081)

[A. Introduction 180](#_Toc332351082)

[1. Overview 180](#_Toc332351083)

[2. Scope 180](#_Toc332351084)

[3. Project Title 180](#_Toc332351085)

[4. Project start date/finish date 180](#_Toc332351086)

[5. Team member (Role and Responsibility) 181](#_Toc332351087)

[6. Current System (Limit) 181](#_Toc332351088)

[7. Solution 181](#_Toc332351089)

[8. Product 182](#_Toc332351090)

[9. Main Acceptable Criteria 183](#_Toc332351091)

[10. License 183](#_Toc332351092)

[11. References 184](#_Toc332351093)

[B. Software Project Management Plan 184](#_Toc332351094)

[I. Problem Definition 184](#_Toc332351095)

[1. Name of this Capstone Project 184](#_Toc332351096)

[2. Problem Abstract 184](#_Toc332351097)

[3. Project Overview 184](#_Toc332351098)

[3.1 The Current System 184](#_Toc332351099)

[3.2 The Proposed System 184](#_Toc332351100)

[3.3 Boundaries of the System 185](#_Toc332351101)

[3.4 Development Environment 185](#_Toc332351102)

[II. Project Organization 185](#_Toc332351103)

[1. Software Process Model 185](#_Toc332351104)

[2. Roles and Responsibilities 186](#_Toc332351105)

[3. Tools and Techniques 187](#_Toc332351106)

[III. Project Management Plan 188](#_Toc332351107)

[1. Tasks 188](#_Toc332351108)

*[1.1](#_Toc332351110)**[Develop Task List](#_Toc332351110)* [188](#_Toc332351110)

*[1.2](#_Toc332351111)**[Create Software Project Plan](#_Toc332351111)* [188](#_Toc332351111)

*[1.3](#_Toc332351112)**[Create Software Requirements Specification](#_Toc332351112)* [189](#_Toc332351112)

*[1.4](#_Toc332351113)**[Design Database](#_Toc332351113)* [189](#_Toc332351113)

*[1.5](#_Toc332351114)**[Create Software Design Description](#_Toc332351114)* [189](#_Toc332351114)

*[1.6](#_Toc332351115)**[Coding](#_Toc332351115)* [190](#_Toc332351115)

*[1.7](#_Toc332351116)**[System Test](#_Toc332351116)* [190](#_Toc332351116)

*[1.8](#_Toc332351117)**[Deployment](#_Toc332351117)* [191](#_Toc332351117)

[2. Task Sheet: Assignments and Timetable 191](#_Toc332351118)

[IV. Coding Conventions 191](#_Toc332351119)

[C. Software Requirement Specifications 192](#_Toc332351120)

[I. User Requirement Specification 192](#_Toc332351121)

[Overview OOPMS User requirement 192](#_Toc332351122)

[1. Dashboard 192](#_Toc332351123)

[2. Planner 192](#_Toc332351124)

[3. Report 192](#_Toc332351125)

[4. Project Eye 192](#_Toc332351126)

[5. Timesheet 194](#_Toc332351127)

[6. DMS 194](#_Toc332351128)

[7. Admin 194](#_Toc332351129)

[8. Requirements 195](#_Toc332351130)

[II. System Requirement Specification (Specific Requirements) 195](#_Toc332351131)

[2.1 Introduction 195](#_Toc332351132)

[2.1.1 Purpose 195](#_Toc332351133)

[2.1.2 Scope 196](#_Toc332351134)

[2.1.3Definitions, Acronyms, and Abbreviations 196](#_Toc332351135)

[2.1.4 References 196](#_Toc332351136)

[2.2 Overall Description 196](#_Toc332351137)

[2.2.1 Product perspective: 197](#_Toc332351138)

[2.2.2 Product functions: 197](#_Toc332351139)

[2.2.3 User characteristics: 197](#_Toc332351140)

[2.2.4 Constraints: 198](#_Toc332351141)

[2.2.5 Assumptions and dependencies: 198](#_Toc332351142)

[2.3 FUNCTIONAL Requirements 198](#_Toc332351143)

[2.3.1 Dashboard 198](#_Toc332351144)

[2.3.2 Planner 198](#_Toc332351145)

[2.3.3 Report 198](#_Toc332351146)

[2.3.4 Project Eye 199](#_Toc332351147)

[2.3.5 Timesheet 200](#_Toc332351148)

[2.3.6 DMS 200](#_Toc332351149)

[2.3.7 Admin 201](#_Toc332351150)

[2.3.8 Requirements 201](#_Toc332351151)

[2.4 Use case model 201](#_Toc332351152)

[2.4.1 General 203](#_Toc332351153)

[2.4.2 Admin 210](#_Toc332351154)

[2.4.3 Planner 219](#_Toc332351155)

[2.4.4 Project Eye 225](#_Toc332351156)

[2.4.5 Change Budget 230](#_Toc332351157)

[2.4.6 Add Expense 231](#_Toc332351158)

[2.4.7 Edit Expense 232](#_Toc332351159)

[2.4.8 Delete Expense 233](#_Toc332351160)

[2.4.9 Add Risk, Issue 234](#_Toc332351161)

[2.4.10 Edit Risk, Issue 235](#_Toc332351162)

[2.4.11 Delete Risk, Issue 236](#_Toc332351163)

[2.4.12 Add Change Request 237](#_Toc332351164)

[2.4.13 Edit Change Request 239](#_Toc332351165)

[2.4.14 Delete Change Request 240](#_Toc332351166)

[2.4.15 Add Product 241](#_Toc332351167)

[2.4.16 Edit Product 241](#_Toc332351168)

[2.4.17 Delete Product 242](#_Toc332351169)

[2.4.18 Add Stage 243](#_Toc332351170)

[2.4.19 Edit Stage 244](#_Toc332351171)

[2.4.20 Delete Stage 245](#_Toc332351172)

[2.4.21 Add Deliverable 247](#_Toc332351173)

[2.4.22 Edit Deliverable 248](#_Toc332351174)

[2.4.23 Delete Deliverable 249](#_Toc332351175)

[2.4.24 View Info 250](#_Toc332351176)

[2.4.25 Report 251](#_Toc332351177)

[2.4.26 Add Requirement 257](#_Toc332351178)

[2.4.27 Update Requirement 258](#_Toc332351179)

[2.4.28 Delete Requirement 259](#_Toc332351180)

[2.4.29 Sort Requirements 260](#_Toc332351181)

[2.4.30 Search defect 262](#_Toc332351182)

[2.4.31 Add defect 263](#_Toc332351183)

[2.4.32 Update defect 265](#_Toc332351184)

[2.4.33 Timesheet Module 266](#_Toc332351185)

[2.5 NON-FUNCTIONAL Requirements 274](#_Toc332351186)

[2.5.1 Usability 274](#_Toc332351187)

[2.5.2 Reliability 274](#_Toc332351188)

[2.5.3 Performance 275](#_Toc332351189)

[2.5.4 Supportability 275](#_Toc332351190)

[2.5.6 On-line User Documentation and Help System Requirements 276](#_Toc332351191)

[2.5.7 Purchased Components 276](#_Toc332351192)

[2.5.8 Interfaces 276](#_Toc332351193)

[2.5.9 Licensing Requirements 276](#_Toc332351194)

[2.5.10 Legal, Copyright, and Other Notices 276](#_Toc332351195)

[2.5.11 Applicable Standards 276](#_Toc332351196)

[D. Software Design Description 276](#_Toc332351197)

[1. Introduction 277](#_Toc332351198)

[1.1 Purpose 277](#_Toc332351199)

[1.2 Scope 277](#_Toc332351200)

[1.3 Intended Audiences and Document Organization 277](#_Toc332351201)

[1.4 Acronyms and Abbreviations 278](#_Toc332351202)

[1.5 References 278](#_Toc332351203)

[2. Architecture design 278](#_Toc332351204)

[2.1 User case view 278](#_Toc332351205)

[2.2 Architectural Representation 280](#_Toc332351206)

[2.2.1 Presentation Layer 281](#_Toc332351207)

[2.2.2 Business Layer 282](#_Toc332351208)

[2.2.3 Data Access Layer 282](#_Toc332351209)

[2.2.4 Data Layer 282](#_Toc332351210)

[2.3 Packages/Components view 283](#_Toc332351211)

[2.3.1 UI Components 283](#_Toc332351212)

[2.3.2 Business Object 284](#_Toc332351213)

[2.3.4 Transfer Data Objects ( Entity ) 284](#_Toc332351214)

[2.3.5 Data Access Object 284](#_Toc332351215)

[2.3.6 Exceptions 284](#_Toc332351216)

[2.3.7 Utils 284](#_Toc332351217)

[2.3.8 Logging 284](#_Toc332351218)

[3. Technical Solutions 284](#_Toc332351219)

[3.1 Exception handling mechanism 284](#_Toc332351220)

[3.2 Logging mechanism 284](#_Toc332351221)

[4. Database design 285](#_Toc332351222)

[4.1 Entity Relationship Diagram 285](#_Toc332351223)

[4.2 Schema 285](#_Toc332351224)

[4.3 Detail Schema 286](#_Toc332351225)

[5. CRC Card Model 295](#_Toc332351226)

[6 Application Security 300](#_Toc332351227)

[6.1 User Permission 300](#_Toc332351228)

[6.2 Security Group 300](#_Toc332351229)

[7 Details function design 300](#_Toc332351230)

[1.1 Requirement \_UC01 - Add Requirement Use Case 301](#_Toc332351231)

[1.1.1 Class Diagram 301](#_Toc332351232)

[1.1.2 Sequence flow 301](#_Toc332351233)

[1.2 Requirement \_UC02 - Update Requirement Use Case 301](#_Toc332351234)

[1.2.1 Class Diagram 302](#_Toc332351235)

[1.2.2 Sequence flow 302](#_Toc332351236)

[1.3 Requirement \_UC03 - Delete Requirement Use Case 302](#_Toc332351237)

[1.3.1 Class Diagram 303](#_Toc332351238)

[1.3.2 Sequence flow 303](#_Toc332351239)

[1.4 Requirement \_UC04 - Sort Requirements Use Case 303](#_Toc332351240)

[1.4.1 Class Diagram 304](#_Toc332351241)

[1.4.2 Sequence flow 304](#_Toc332351242)

[1.5 Admin \_UC01 - Admin Create new Project Use Case 305](#_Toc332351243)

[1.5.1 Class Diagram 305](#_Toc332351244)

[1.5.2 Sequence flow 306](#_Toc332351245)

[1.6 Admin \_UC02 - Admin Search Project Use Case 306](#_Toc332351246)

[1.6.1 Class Diagram 306](#_Toc332351247)

[1.6.2 Sequence flow 307](#_Toc332351248)

[1.7 Admin \_UC03 - Admin Edit Project Use Case 307](#_Toc332351249)

[1.7.1 Class Diagram 307](#_Toc332351250)

[1.7.2 Sequence flow 308](#_Toc332351251)

[1.8 Admin \_UC04 - Admin Mange Module Use Case 308](#_Toc332351252)

[1.8.1 Class Diagram 308](#_Toc332351253)

[1.8.2 Sequence flow 309](#_Toc332351254)

[1.9 Admin \_UC05 - Admin Team Management Use Case 309](#_Toc332351255)

[1.9.1 Class Diagram 309](#_Toc332351256)

[1.9.2 Sequence flow 310](#_Toc332351257)

[1.10 Admin \_UC06 - Admin Search User Use Case 311](#_Toc332351258)

[1.10.1 Class Diagram 311](#_Toc332351259)

[1.10.2 Sequence flow 311](#_Toc332351260)

[1.11 Admin \_UC07 - Admin Create User Use Case 312](#_Toc332351261)

[1.11.1 Class Diagram 312](#_Toc332351262)

[1.11.2 Sequence flow 312](#_Toc332351263)

[1.12 Admin \_UC08 - Admin Edit User Use Case 313](#_Toc332351264)

[1.12.1 Class Diagram 313](#_Toc332351265)

[1.12.2 Sequence flow 313](#_Toc332351266)

[1.13 Admin \_UC09 - Admin Change User’s Status Use Case 314](#_Toc332351267)

[1.13.1 Class Diagram 314](#_Toc332351268)

[1.13.2 Sequence flow 314](#_Toc332351269)

[1.14 ProjectEye \_UC01 - Create new Project Use Case 315](#_Toc332351270)

[1.14.1 Class Diagram 315](#_Toc332351271)

[1.14.2 Sequence flow 316](#_Toc332351272)

[1.15 ProjectEye \_UC02 - Edit Project Use Case 316](#_Toc332351273)

[1.15.1 Class Diagram 316](#_Toc332351274)

[1.15.2 Sequence flow 317](#_Toc332351275)

[1.16 ProjectEye \_UC03 - Manage Module Use Case 317](#_Toc332351276)

[1.16.1 Class Diagram 317](#_Toc332351277)

[1.16.2 Sequence flow 318](#_Toc332351278)

[1.17 ProjectEye \_UC04 - Team Management Use Case 318](#_Toc332351279)

[1.17.1 Class Diagram 318](#_Toc332351280)

[1.17.2 Sequence flow 319](#_Toc332351281)

[1.18 ProjectEye \_UC05 - Add Risk Use Case 319](#_Toc332351282)

[1.18.1 Class Diagram 319](#_Toc332351283)

[1.18.2 Sequence flow 320](#_Toc332351284)

[1.19 ProjectEye \_UC06 - Edit Risk Use Case 320](#_Toc332351285)

[1.19.1 Class Diagram 320](#_Toc332351286)

[1.19.2 Sequence flow 321](#_Toc332351287)

[1.20 ProjectEye \_UC07 - Delete Risk Use Case 321](#_Toc332351288)

[1.20.1 Class Diagram 321](#_Toc332351289)

[1.20.2 Sequence flow 322](#_Toc332351290)

[1.21 ProjectEye \_UC08 - Add Issue Use Case 322](#_Toc332351291)

[1.21.1 Class Diagram 322](#_Toc332351292)

[1.21.2 Sequence flow 323](#_Toc332351293)

[1.22 ProjectEye \_UC09 - Edit Issue Use Case 323](#_Toc332351294)

[1.22.1 Class Diagram 323](#_Toc332351295)

[1.22.2 Sequence flow 324](#_Toc332351296)

[1.23 ProjectEye \_UC10 - Delete Issue Use Case 324](#_Toc332351297)

[1.23.1 Class Diagram 324](#_Toc332351298)

[1.23.2 Sequence flow 325](#_Toc332351299)

[1.24 ProjectEye \_UC11 - Add Change Request Use Case 325](#_Toc332351300)

[1.24.1 Class Diagram 325](#_Toc332351301)

[1.24.2 Sequence flow 326](#_Toc332351302)

[1.25 ProjectEye \_UC12 - Edit Change Request Use Case 326](#_Toc332351303)

[1.25.1 Class Diagram 326](#_Toc332351304)

[1.25.2 Sequence flow 327](#_Toc332351305)

[1.26 ProjectEye \_UC13 - Delete Change Request Use Case 327](#_Toc332351306)

[1.26.1 Class Diagram 327](#_Toc332351307)

[1.26.2 Sequence flow 328](#_Toc332351308)

[1.27 ProjectEye \_UC14 - Add Product Use Case 328](#_Toc332351309)

[1.27.1 Class Diagram 328](#_Toc332351310)

[1.27.2 Sequence flow 329](#_Toc332351311)

[1.28 ProjectEye \_UC15 - Edit Product Use Case 329](#_Toc332351312)

[1.28.1 Class Diagram 329](#_Toc332351313)

[1.28.2 Sequence flow 330](#_Toc332351314)

[1.29 ProjectEye \_UC16 - Delete Product Use Case 330](#_Toc332351315)

[1.29.1 Class Diagram 330](#_Toc332351316)

[1.29.2 Sequence flow 331](#_Toc332351317)

[1.30 ProjectEye \_UC17 - Add Stage Use Case 331](#_Toc332351318)

[1.30.1 Class Diagram 331](#_Toc332351319)

[1.30.2 Sequence flow 332](#_Toc332351320)

[1.31 ProjectEye \_UC18 - Edit Stage Use Case 332](#_Toc332351321)

[1.31.1 Class Diagram 332](#_Toc332351322)

[1.31.2 Sequence flow 333](#_Toc332351323)

[1.32 ProjectEye \_UC19 - Delete Stage Use Case 333](#_Toc332351324)

[1.32.1 Class Diagram 333](#_Toc332351325)

[1.32.2 Sequence flow 334](#_Toc332351326)

[1.33 ProjectEye \_UC20 - Add Deliverable Use Case 334](#_Toc332351327)

[1.33.1 Class Diagram 334](#_Toc332351328)

[1.33.2 Sequence flow 335](#_Toc332351329)

[1.34 ProjectEye \_UC21 - Edit Deliverable Use Case 335](#_Toc332351330)

[1.34.1 Class Diagram 335](#_Toc332351331)

[1.34.2 Sequence flow 336](#_Toc332351332)

[1.35 ProjectEye \_UC22 - Delete Deliverable Use Case 336](#_Toc332351333)

[1.35.1 Class Diagram 336](#_Toc332351334)

[1.35.2 Sequence flow 337](#_Toc332351335)

[1.36 ProjectEye \_UC23 - View Info Use Case 337](#_Toc332351336)

[1.36.1 Class Diagram 337](#_Toc332351337)

[1.36.2 Sequence flow 338](#_Toc332351338)

[1.37 Planner\_UC01- Filter Task Use Case 338](#_Toc332351339)

[1.37.1 Class Diagram 338](#_Toc332351340)

[1.37.2 Sequence Flow 339](#_Toc332351341)

[1.38 Planner\_UC02 - Add Task Use Case 340](#_Toc332351342)

[1.38.1 Class Diagram 340](#_Toc332351343)

[1.38.2 Sequence Flow 341](#_Toc332351344)

[1.39 Planner\_UC03 - Update Task Use Case 342](#_Toc332351345)

[1.39.1 Class Diagram 342](#_Toc332351346)

[1.39.2 Sequence Flow 343](#_Toc332351347)

[1.40 Planner\_UC04 - Delete Task Use Case 343](#_Toc332351348)

[1.40.1 Class Diagram 343](#_Toc332351349)

[1.40.2 Sequence Flow 344](#_Toc332351350)

[1.41 Planner\_UC05 - Import Task Use Case 344](#_Toc332351351)

[1.41.1 Class Diagram 344](#_Toc332351352)

[1.41.2 Sequence Flow 345](#_Toc332351353)

[1.42 Dashboard \_UC01 - Filter Project Use Case 345](#_Toc332351354)

[1.42.1 Class Diagram 345](#_Toc332351355)

[1.42.2 Sequence Flow 346](#_Toc332351356)

[1.43 Dashboard \_UC02 - Export Dashboard Use Case 346](#_Toc332351357)

[1.43.1 Class Diagram 346](#_Toc332351358)

[1.43.2 Sequence Flow 347](#_Toc332351359)

[8. Interface Design 347](#_Toc332351360)

[8.1 Project Eye Portlet 348](#_Toc332351361)

[8.2 Team Management 348](#_Toc332351362)

[8.3 Planner Portlet 349](#_Toc332351363)

[8.4 Requirement Portlet 349](#_Toc332351364)

[8.5 Timesheet Portlet 350](#_Toc332351365)

[9. Configuration 350](#_Toc332351366)

[10. Packaging and Deployment 350](#_Toc332351367)

[E. Software Test Documentation 351](#_Toc332351368)

[1. System Overview 351](#_Toc332351369)

[2. Test Approach 351](#_Toc332351370)

[3. Features to be tested 352](#_Toc332351371)

[3.1 Dashboard 352](#_Toc332351372)

[3.2 Planner 352](#_Toc332351373)

[3.3 Report 352](#_Toc332351374)

[3.4 Project Eye 352](#_Toc332351375)

[3.5 Timesheet 353](#_Toc332351376)

[3.6 DMS 353](#_Toc332351377)

[3.7 Requirement 353](#_Toc332351378)

[3.8 Admin 353](#_Toc332351379)

[3.9 Android 353](#_Toc332351380)

[4. Features not to be tested 354](#_Toc332351381)

[4.1 Text formatting 354](#_Toc332351384)

[4.2 Image formatting 354](#_Toc332351385)

[4.3 Performance 354](#_Toc332351386)

[4.4 Network Connection & Security 354](#_Toc332351387)

[4.5 Performance 354](#_Toc332351388)

[5. Test Case 354](#_Toc332351389)

[5.1 Project Eye 355](#_Toc332351390)

[5.1.1 Check view project list 355](#_Toc332351391)

[5.1.2 Create project 355](#_Toc332351392)

[5.1.3 Team management 355](#_Toc332351393)

[5.1.4 Create Product 355](#_Toc332351394)

[5.1.5 Delete Product 355](#_Toc332351395)

[5.1.6 Update Product 356](#_Toc332351396)

[5.1.7 Create Risk 356](#_Toc332351397)

[5.1.8 Delete Risk 356](#_Toc332351398)

[5.1.9 Update Risk 356](#_Toc332351399)

[5.1.10 Create Issue 356](#_Toc332351400)

[5.1.11 Delete Issue 356](#_Toc332351401)

[5.1.12 Update Issue 357](#_Toc332351402)

[5.1.13 Create Change 357](#_Toc332351403)

[5.1.14 Delete Change 357](#_Toc332351404)

[5.1.15 Update Change 357](#_Toc332351405)

[6. Checklists 358](#_Toc332351406)

[1.1 Checklist of Validation 358](#_Toc332351408)

[1.2 Submission Checklist 359](#_Toc332351409)

[7. Defect Log 360](#_Toc332351410)

[8. Test Log 362](#_Toc332351411)

[F. User Manual 362](#_Toc332351412)

[I. Installation Guide 362](#_Toc332351413)

[Users’ Guide 362](#_Toc332351414)

[II. 362](#_Toc332351430)

# Introduction

## Overview

**In this accelerating of technologies and economics, Project Management is growing extremely importance to organizations because it deals effectively with the management of change. Modern economics and business environment are complicated than ever, organizations are beginning to realize that the traditional forms of management cannot adapt efficiently to the dynamics and its resulting chaos**.

In the future, software industries could very well depend on how quickly the procedures and systems of project management are adopted. Therefore, Project Manager will play a very important role within organizations. Our mission is to provide an online system that actively assists those managers in managing their project. The approaches and techniques that are used in the project management process are of interest to all those who wish to be more certain about achieving predetermined targets.

## Scope

This project aims to create an online Software Project Management System. With friendly interface, powerful tools, OOPMS provides powerful and efficient customized service for numerous kinds of managers from small to medium projects. Besides, OOPMS can deploy on J2EE portal servers (JSR168, JSR268).

## Project Title

Project Full name: Online Project Management Suite on Portal Framework

Project code: OOPMS

Software name: OOPMS

## Project start date/finish date

From May-07-2012 to Aug-18-2012

## Team member (Role and Responsibility)

|  |  |  |
| --- | --- | --- |
| Name | Roles | Responsibilities |
| Lê Ngọc Thạch | Project Manager | Supervisor, Assistant, Director |
| Ngô Đức Duy | Team Leader | Control, Audit, Support, Engineer, Training, Customer Support |
| Mạnh Hoàng Trương | Team Member | Engineer, Training, Customer Support |
| Phạm Nguyễn Trường Giang | Team Member | Engineer, Training, Customer Support |
| Tô Công Thanh Hải | Team Member | Engineer, Training, Customer Support |

## Current System (Limit)

Many current systems provide project management services. Still, project managers find it extremely problematic to apply those systems into their real projects. There are numerous reasons making these systems very difficult to use efficiently:

1. Unfriendly Interface
2. Complicated Process
3. Large cost
4. Not open source
5. Not modularization

## Solution

From above issues of current project management systems, we introduce a new system with powerful features for small and medium projects:

* Provide friendly user interface.
* Add more visual items to help user easy to use, reduce ambiguous logic, improve performance.
* Provide useful report function.
* Integrate in mobile phone to support tracking and making decision for managers anytime.

## Product

Include functions:

1. **Dashboard**: show present status of projects that created in OOPMS. Project managers and others senior manager can easily track and monitor status of their project.
2. **Planner**:
   1. Allow users import Microsoft Project file (\*.mpp) into OOPMS and in the future, users also can edit and export \*.mpp files at OOPMS;
   2. Create task and assign task to team members.
3. **Report**: support creating report of project weekly and allow Team leader of project add more comments to the report.
4. **Project Eye**: manage detail information about a specific project.
5. **Time Tracking**: manage working time of team members.
6. **DMS**: manage defects of projects.
7. **Requirement**: include list of requirements (function, non-function). Allow users to keep track status of those requirements
8. **Admin**: Allow admin to manage users and manage system data.

Benefits of product

OOPMS is a system that supports Project managers efficiently.

OOPMS provides as an android application for project managers to monitor general information about project.

At the website, Project manager can setup and manage any software project easily via Dashboard system and Report system that update continuously day by day.

A special benefit of OOPMS system is portability and compatible with J2EE Portal server (JSR168, JSR268). With an android mobile, you can check your project status at any time anywhere.

## Main Acceptable Criteria

The system is 24/7 system.

Accuracy: 90%

Critical bugs:

+ Loss of all data: no

+ Unable to use part of the system’s functionality: not any

Performance:

Response time for a search:

Average2 second / search

Maximum: 3 second/ search

|  |  |  |  |
| --- | --- | --- | --- |
| Item | In progress | | After release |
| Self-Review | Peer Review |
| Document | 2 bug/page (min) | 1bug/page (min) | 2bug/10page(max) |
| Source Code | 20Wdef/1KLOC(min) | 8Wdef/1KLOC(min) | 3Wdef/1KLOC(max) |

*\*Note:*

*Wdef: Weighted defect*

*Type Weight*

*- Bug layout and coding convention 1*

*- Bug logic of code 3*

*- Bug fatal error 5*

## 10. License

OOPMS licensed under the [Apache License, Version 2.0](http://www.apache.org/licenses/LICENSE-2.0) as approved by the Open Source Initiative (OSI), an [OSI-certified](http://www.opensource.org/docs/osd) ("open") and [Gnu/FSF-recognized](http://www.gnu.org/licenses/license-list.html) ("free") license.

## References

* Wikipedia.org
* FMS System

# Software Project Management Plan

## Problem Definition

### Name of this Capstone Project

The official and formal project name is “Online Project Management Suite on Portal Framework”.

### Problem Abstract

In the future, software industries could very well depend on how quickly the procedures and systems of project management are adopted. Therefore, Project Manager will play a very important role within organizations. Our mission is to provide an online system that actively assists those managers in managing their project.

With the increasing number of managers and users, OOPMS must be capable to work well with thousands of users. OOPMS processes a large amount of information, so the system needs to have good performance and accuracy. Any mistakes may lead to a big failure in projects’ statistic and performance.

### Project Overview

#### The Current System

There are many current systems, which provide project management services. Still, project managers find it extremely problematic to apply those systems into their real projects. There are numerous reasons making these systems very difficult to use efficiently:

* + - Unfriendly Interface
    - Complicated Process
    - Large cost
    - Not open source
    - Not modularization

#### The Proposed System

From above issues of current project management systems, we introduce a new system with powerful features for small and medium projects:

* Provide friendly user interface.
* Add more visual items to help user easy to use, reduce ambiguous logic, improve performance.
* Provide useful report function.
* Integrate in mobile phone to support tracking and making decision for managers anytime.

#### Boundaries of the System

The system under development of this Capstone Project will include:

* A complete website version and an Android version.
* All the process documents involved.

#### Development Environment

Below is the list of hardware and software requirements needed for development environment:

Hardware requirements:

* Personal computers for developing with the minimum configuration: 2 Gb of RAM, 20Gb of hard disk, Core 2 Duo 2.0 GHz

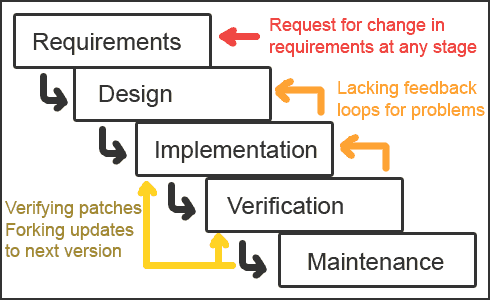
Software requirements:

* Operating system: Windows 7/Windows XP
* IDE: Eclipse
* Dreamweaver
* Microsoft Office
* Microsoft Visio
* Microsoft Project
* DBMS: Oracle 10g Express Edition
* Source Control: SVN

## Project Organization

1. Software Process Model

The process model used for developing this project is waterfall model.



1. Roles and Responsibilities

|  |  |  |  |
| --- | --- | --- | --- |
| No | **Full name** | **Role in Group** | **Responsibilities** |
| 1 | Ngô Đức Duy | Team Leader, Developer, Tester, QA, CM | Managing process  Designing database  Requirement development  Requirement analysis  Architectural design  Project planning and scheduling  Task assignment and tracking  Training  Coding  Testing  Task conducting  Reporting  Customer interaction |
| 2 | Phạm Nguyễn Trường Giang | Developer, Tester, QA, BA | Managing documents  Creating system test cases  Designing database  Requirement development  Requirement analysis  Architectural design  Project planning and scheduling  Coding  Testing  Task conducting  Reporting |
| 3 | Tô Công Thanh Hải | Developer, Tester, QA | Managing documents  Creating system test cases  Designing database  Requirement development  Requirement analysis  Architectural design  Project planning and scheduling  Coding  Testing  Task conducting  Reporting |
| 4 | Mạnh Hoàng Trương | Developer, Tester, QA | Managing documents  Creating system test cases  Designing database  Requirement development  Requirement analysis  Architectural design  Project planning and scheduling  Coding  Testing  Task conducting  Reporting |
| 5 | Lê Ngọc Thạch | Project Manager | Managing process  Architectural design  Project planning and scheduling  Task assignment and tracking  Training  Testing  Customer interaction |

1. Tools and Techniques

* Front-end technologies: Java, Portal(JSF168,268), jQuery, AJAX, Javascript, CSS.
* Back-end technologies: Hibernate.
* Tools: Eclipse, JBoss, Oracle 10g, Microsoft Visio 2010, Microsoft Office 2010, Microsoft Project 2010, Source Version Control, Chrome. Architecture and design patterns: Three Tiers Architecture, Repository.

## Project Management Plan

1. Tasks
2. 1. *Develop Task List*

Description

Develop task, milestone and assign task for team members.

Output

Task list

Deliverables

Deliver task list, project plan document 24/05/2012

Resources Needed

A person for 2 days

Dependencies and Constraints

None

Risks

None

* 1. *Create Software Project Plan*

Description

Create software project plan

Output

Software project management plan (SPM) document

Deliverables

Deliver SPM document 24/05/2012

Resources Needed

One person for 2 days

Dependencies and Constraints

None

Risks

None

* 1. *Create Software Requirements Specification*

Description

Create software requirements specification

Output

Software Requirement Specification (SRS) document

Deliverables

Deliver SRS document 03/06/2012

Resources Needed

Four people for 7 days

Dependencies and Constraints

None

Risks

Missing requirement, changing requirement and scope of this project.

* 1. *Design Database*

Description

Create conceptual, logical, physical database design

Output

Database Design and Database script

Deliverables

Deliver with Software Design Description 10/06/2012

Resources Needed

Four people for 7 days

Dependencies and Constraints

Depends on the completion of SRS

Risks

SRS may not be detailed enough to capture the business rules, causing the database design to be inappropriate or will be changed much in the future.

* 1. *Create Software Design Description*

Description

Design the system in an OOP manner.

Output

Architecture design, detailed design, diagrams and design specification

Deliverables

SDD 22/06/2012

Resources Needed

Four people for 12 days

Dependencies and Constraints

Depends on the completion of SRS

Risks

Misunderstanding SRS and not review carefully will make risk occur.

* 1. *Coding*

Description

Implement the system to reflect SDD

Output

Source code of the project, unit test reports

Deliverables

Executable programs and source code 27/07/2012

Resources Needed

Four people for 35 days

Dependencies and Constraints

Depends on the completion of SRS, SDD, Database design, Coding framework

Risks

* Unit test may not be performed thoroughly causing spending many efforts in system test phase.
* The team don’t have a common place, so it is hard to perform collaboration and CM
* Team members may not easily get used to new technologies.
* Team members are not expert in developing layout and css. This can cause more time for improve layout.
  1. *System Test*

Description

Perform system test for the system

Output

System test report

Deliverables

Software Test Documentation before 12/08/2012

Resources Needed

Four people for 15 days

Dependencies and Constraints

Coding is finished

Risks

* Lack of professional testers in team
* Developers are also responsible for system testing, this may lead to compromise
  1. *Deployment*

Description

Deploy the system, create user manual

Output

Running website with domain and hosting, user manual

Deliverables

Delivery user manual before 18/08/2012

Resources Needed

Four people for 6 days

Dependencies and Constraints

Coding and system test are finished, initial data is inputted

Risks

* Composing document can cause by integrate from many team members’ document.
* Training staff may take more time.

1. Task Sheet: Assignments and Timetable

Refer to OOPMS\_TaskList.mpp

OOPMS 76.85d Mon 5/7/12 Tue 8/21/12 70%

Initiation 2d Mon 5/7/12 Tue 5/8/12 100% All team members

Kick-off meeting 1d Mon 5/7/12 Mon 5/7/12 100%

Introduce overview about Eclipse plug-in development 4h Tue 5/8/12 Tue 5/8/12 3 100%

Introduce overview about Android development 4h Tue 5/8/12 Tue 5/8/12 4 100%

Planning 5d Wed 5/9/12 Tue 5/15/12 2 100%

Develop scope statement 3d Wed 5/9/12 Fri 5/11/12 100% All team members

Create WBS 2d Wed 5/9/12 Thu 5/10/12 100% DuyND

Develop and refine other plans 2d Mon 5/14/12 Tue 5/15/12 7 100% All team members

Solution 28d Wed 5/16/12 Fri 6/22/12 92%

Report 1 4d Wed 5/16/12 Mon 5/21/12 6 100% All team members

Study Requirements + Report 2 4d Tue 5/22/12 Fri 5/25/12 11 100% All team members

Develop functional specifications 5d Mon 5/28/12 Fri 6/1/12 90%

Create SRS 5d Mon 5/28/12 Fri 6/1/12 12 90% All team members

Develop prototype based on functional specifications 13d Mon 6/4/12 Wed 6/20/12 14 87%

Database Design 5d Mon 6/4/12 Fri 6/8/12 85% All team members

ScreenDesign 2d Mon 6/11/12 Tue 6/12/12 16 85% All team members

Activity Diagram 2d Wed 6/13/12 Thu 6/14/12 17 90% All team members

Sequence Diagram 2d Fri 6/15/12 Mon 6/18/12 18 90% All team members

DDD 2d Tue 6/19/12 Wed 6/20/12 19 85% All team members

Review functional specifications (Offline) 1d Thu 6/21/12 Thu 6/21/12 20 100% All team members

Assess Understand the requirement 1d Fri 6/22/12 Fri 6/22/12 21 100% All team members

Construction 36.85d Mon 6/25/12 Tue 8/14/12 10 58%

Development 25.5d Mon 6/25/12 Mon 7/30/12 52%

Create coding plan 4h Mon 6/25/12 Mon 6/25/12 100%

Development 16d Mon 6/25/12 Tue 7/17/12 25 49%

Screen design 5d Mon 6/25/12 Mon 7/2/12 0%

"Pages (Requirement, Portal)" 3d Mon 6/25/12 Thu 6/28/12 0% DuyND

"Pages (DMS, Timesheet)" 3d Mon 6/25/12 Thu 6/28/12 0% TruongMH

"Pages (Project Eyes, Admin, Product)" 3d Mon 6/25/12 Thu 6/28/12 0% HaiTCT

"Pages (Report, Dashboard, Planner)" 3d Mon 6/25/12 Thu 6/28/12 0% GiangPNT

Review Pages 2d Thu 6/28/12 Mon 7/2/12 31 0% All team members

"Database Design, Pages review, mapping" 2d Mon 7/2/12 Wed 7/4/12 32 100% All team members

Functions code 9d Wed 7/4/12 Tue 7/17/12 33 67%

Function 1 Requirement 3d Wed 7/4/12 Mon 7/9/12 100% DuyND

Function 2 DMS 3d Wed 7/4/12 Mon 7/9/12 30% TruongMH

Function 3 Admin 3d Wed 7/4/12 Mon 7/9/12 0% HaiTCT

Function 4 Dashboard 3d Wed 7/4/12 Mon 7/9/12 25% GiangPNT

Function 5 Portal 4d Mon 7/9/12 Fri 7/13/12 35 90% DuyND

Function 6 Timesheet 4d Mon 7/9/12 Fri 7/13/12 36 100% TruongMH

"Function 7 Admin, Product" 4d Mon 7/9/12 Fri 7/13/12 37 80% HaiTCT

Function 8 Planner 4d Mon 7/9/12 Fri 7/13/12 38 95% GiangPNT

Function 9 Project Eye 2d Fri 7/13/12 Tue 7/17/12 39 90% All team members

Function 10 Receipt 2d Fri 7/13/12 Tue 7/17/12 39 25% All team members

Review code offline 1d Tue 7/17/12 Wed 7/18/12 44 0% All team members

Review to update/fix bug for final documents 1d Wed 7/18/12 Thu 7/19/12 45 60%

Modify if needed 1d Thu 7/19/12 Fri 7/20/12 46 60% All team members

Integration 2d Fri 7/20/12 Tue 7/24/12 47 80% All team members

Study and develop Android Version 4d Tue 7/24/12 Mon 7/30/12 48 70% All team members

Testing 12.35d Fri 7/27/12 Tue 8/14/12 49 90% All team members

Create Testing plan + Document 4h Fri 7/27/12 Mon 7/30/12 80%

Unit Testing 5.5d Mon 7/30/12 Tue 8/7/12 100%

Create Test case 1d Mon 7/30/12 Tue 7/31/12 51 100%

Review test case online 1d Tue 7/31/12 Wed 8/1/12 53 100%

Create Test code 1d Wed 8/1/12 Thu 8/2/12 54 100%

Review Test code and Unit Test Statistical Report 4h Thu 8/2/12 Fri 8/3/12 55 100%

"Modify code (fix bug, if any)" 1d Fri 8/3/12 Mon 8/6/12 56 100%

Re-test modified code 4h Mon 8/6/12 Mon 8/6/12 57 100%

"Assess Unit Testing (Write Test Case, Test Code, Test Report)" 4h Mon 8/6/12 Tue 8/7/12 58 100%

Integration Testing 5.5d Tue 8/7/12 Tue 8/14/12 80%

Create Test case 1d Tue 8/7/12 Wed 8/8/12 "48,59" 100%

Review test case online 1d Wed 8/8/12 Thu 8/9/12 61 100%

Create Test code 1d Thu 8/9/12 Fri 8/10/12 62 100%

Review Test code and Unit Test Statistical Report 4h Fri 8/10/12 Fri 8/10/12 63 100%

"Modify code (fix bug, if any)" 1d Fri 8/10/12 Mon 8/13/12 64 90%

Re-test modified code 4h Mon 8/13/12 Tue 8/14/12 65 0%

"Assess Unit Testing (Write Test Case, Test Code, Test Report)" 4h Tue 8/14/12 Tue 8/14/12 66 0%

Monitoring and Controlling 63d Wed 5/16/12 Fri 8/10/12 6 75% DuyND

Termination 5d Tue 8/14/12 Tue 8/21/12 23 0% All team members

Milestone assessment 2d Tue 8/14/12 Thu 8/16/12 0% All team members

Deployment 3d Thu 8/16/12 Tue 8/21/12 70 0%

## Coding Conventions

Refer to <http://code.google.com/p/oopms/wiki/Standard_Java_Coding_Convention>

# Software Requirement Specifications

## User Requirement Specification

### Overview OOPMS User requirement

This section of the document describes all the functions that the PMS application shall provide accordingly to user requirements.

### Dashboard

This function provides brief information about all projects in the system:

* Provide overall information about projects ‘status including name, health, cost, progress, quality, efficiency, date, effort.
* Allow user to filter projects by status, category, date, cost, progress, efficiency.
* Allow user to sort projects by name, date, project manager.
* Provide link to project detail information and export feature.

### Planner

This function allow user to keep track of tasks, progress of project:

* Allow project manager to manage tasks: add new task, update, delete, assign, monitor progress, completeness rate, and date.
* Feature filter projects by status, assigned to.
* Allow user to sort status, date, assigned to.
* Import feature using Microsoft Project and export report.
* Team members can update tasks’ progress.

### Report

This function provides report features:

* Project information
* Planner
* DMS
* Timesheet

### Project Eye

This function allows Admin and Project Manager to create new project, edit a project’s information. The function also provides Admin with search function to help them quickly find a project.

**4.1 Manage User**

This function provides Admin with the ability manage users of the PMS system. This function includes search users, create user, and edit user’s information and status.

**4.2 Manage Project Team**

This function provides Admin and Project Manager the ability to manage a project’s team. This function includes search users to add to project’s team, change team member’s role, and remove member from project’s team.

**4.3 Manage Project Module**

This function provides Admin and Project Manager the ability to manage a project’s modules usage. This function will help admin and project manager to choose what modules they want to use in a project.

Here is the list of modules:

* Time Sheet
* Project Planner
* Defect Management
* Report
* Requirement Management
* Risk, Issue Management
* Change Request Management
* Cost Management

**4.4 Change User’s Information**

This function provides users of PMS system the ability to change their user’s information. This also includes changing password function.

**Reset Password**

This function will help users of PMS system when they forget their password. Users provide their username and this function will reset and send their password to their Email in user’s information.

**4.5 Manage Cost**

This function provides Project Managers the ability to control a project’s finance. This function will help them manage budget and tracking expense daily. It also alarms them when their project will be over budget at the time they planning the project’s expense. Project Managers can defy their project’s buffer, the Manage Cost module will notice them the status of the current buffer.

**4.6 Manage Product**

This function provides Project Manager and Team Member the ability to control a project’s products. Project Manager and Team Member can add, modify and delete products.

**4.7 Manage Work Order**

This function provides Project Manager the ability to control a project’s stage and deliverables’ schedule. Project Managers can set how many stages a project has and their duration. They can also set when and which product to deliver in each stage.

**4.8 Manage Risk, Issue, Change Request**

This function provides Project Manager and Team Member the ability to control a project’s risk, issue and change request. They can add, modify and delete risk, issue and change request.

### Timesheet

This function allows manager to keep track of time and effort:

* Allow project manager to monitor timework and effort of team member.
* Feature filter projects by date, status, project.
* Team members can log timesheet as their works daily or weekly.

### DMS

This function allows manager to keep track of defects:

* Allow project manager to monitor defect and fix bug effort of team.
* Feature filter projects by date, status, and project, creator, assigned to.
* Team members can update defect status.

### 7. Admin

**7.1 Admin**

* Manage Project
* Manage User
* Manage Project Team
* Manage Project Module

**7.2 User**

* Change User’s Information
* Reset Password
* Manage Project
* Manage Project Team
* Manage Project Module
* Manage Cost
* Manage Product
* Manage Work Order
* Manage Risk, Issue, Change Request

### Requirements

* Managing feature enables user to store and manage their requirement documents.
* Including: Add, update, remove
* User interface allows user to sort requirement by type, date, priority.
* Display completeness rate updated by user.

## System Requirement Specification (Specific Requirements)

### 2.1 Introduction

In this accelerating of technologies and economics, Project Management is growing extremely importance to organizations because it deals effectively with the management of change. Modern economics and business environment are complicated than ever, organizations are beginning to realize that the traditional forms of management cannot adapt efficiently to the dynamics and its resulting chaos.

In the future, software industries could very well depend on how quickly the procedures and systems of project management are adopted. Therefore, Project Manager will play a very important role within organizations. Our mission is to provide an online system that actively assists those managers in managing their project. The approaches and techniques that are used in the project management process are of interest to all those who wish to be more certain about achieving predetermined targets.

#### 2.1.1 Purpose

This Software Requirements Specification provides a complete description of all the functional requirements, non-functional requirements, constraints and other requirement specification of the “Project Management System”.

#### 2.1.2 Scope

This project aims to create an online Software Project Management System. With friendly interface, powerful tools, OOPMS provides powerful and efficient customized service for numerous kinds of managers from small to medium projects. Besides, OOPMS can deploy on J2EE portal servers (JSR168, JSR268).

#### 2.1.3Definitions, Acronyms, and Abbreviations

Table 1 Abbreviations

|  |  |  |
| --- | --- | --- |
| No. | Key word | Meaning |
| 1 | PM | Project Manager |
| 2 | PMS | Project Management System |
| 3 | ISO | International Standard Organization |

Table: Abbreviations

#### 2.1.4 References

FSOFT PMS System

### 2.2 **Overall Description**

This chapter describes an overview description of this document by listing all the functions, characteristics, constraints of the application.



#### 2.2.1 Product perspective:

The OOPMS develop to manage requirements, tasks, defects, time tracking and report in software projects.

#### 2.2.2 Product functions:

The PMS provides functions as below:

1. Dashboard
2. Planner
3. Report
4. Project Eye
5. Time Tracking
6. DMS
7. Requirement
8. Admin

#### 2.2.3 User characteristics:

The PMS is designed for human operators who are assumed to have the following characteristics:

+ Fully understand the process of Software project management and how it is crucial to business.

+ No special knowledge or skills.

#### 2.2.4 Constraints:

The PMS is a real time system that is responsible for managing project information. It must operate correctly, accurately and in time.

#### 2.2.5 Assumptions and dependencies:

None

### 2.3 **FUNCTIONAL Requirements**

This chapter of the document describes all the functions that the OOPMS application shall provide, and how the system operating these functions using natural language and model, chart.

#### 2.3.1 Dashboard

This function provides brief information about all projects in the system:

* Provide overall information about projects ‘status including name, health, cost, progress, quality, efficiency, date, effort.
* Allow user to filter projects by status, category, date, cost, progress, efficiency.
* Allow user to sort projects by name, date, project manager.
* Provide link to project detail information and export feature.

#### 2.3.2 Planner

This function allow user to keep track of tasks, progress of project:

* Allow project manager to manage tasks: add new task, update, delete, assign, monitor progress, completeness rate, and date.
* Feature filter projects by status, assigned to.
* Allow user to sort status, date, assigned to.
* Import feature using Microsoft Project and export report.

Team member can update task’s progess.

#### 2.3.3 Report

This function provides report features:

* Project information
* Planner
* DMS
* Timesheet

#### 2.3.4 Project Eye

This function allows Admin and Project Manager to create new project, edit a project’s information. The function also provides Admin with search function to help them quickly find a project.

2.3.4.1 Manage User

This function provides Admin with the ability manage users of the PMS system. This function includes search users, create user, and edit user’s information and status.

2.3.4.2 Manage Project Team

This function provides Admin and Project Manager the ability to manage a project’s team. This function includes search users to add to project’s team, change team member’s role, and remove member from project’s team.

2.3.4.3 Manage Project Module

This function provides Admin and Project Manager the ability to manage a project’s modules usage. This function will help admin and project manager to choose what modules they want to use in a project.

Here is the list of modules:

* Time Sheet
* Project Planner
* Defect Management
* Report
* Requirement Management
* Risk, Issue Management
* Change Request Management
* Cost Management

2.3.4.4 Change User’s Information

This function provides users of PMS system the ability to change their user’s information. This also includes changing password function.

Reset Password

This function will help users of PMS system when they forget their password. Users provide their username and this function will reset and send their password to their Email in user’s information.

2.3.4.5 Manage Cost

This function provides Project Managers the ability to control a project’s finance. This function will help them manage budget and tracking expense daily. It also alarms them when their project will be over budget at the time they planning the project’s expense. Project Managers can defy their project’s buffer, the Manage Cost module will notice them the status of the current buffer.

2.3.4.6 Manage Product

This function provides Project Manager and Team Member the ability to control a project’s products. Project Manager and Team Member can add, modify and delete products.

2.3.4.7 Manage Work Order

This function provides Project Manager the ability to control a project’s stage and deliverables’ schedule. Project Managers can set how many stages a project has and their duration. They can also set when and which product to deliver in each stage.

2.3.4.8 Manage Risk, Issue, Change Request

This function provides Project Manager and Team Member the ability to control a project’s risk, issue and change request. They can add, modify and delete risk, issue and change request.

#### 2.3.5 Timesheet

This function allows manager to keep track of time and effort:

* Allow project manager to monitor timework and effort of team member.
* Feature filter projects by date, status, project.
* Team members can log timesheet as their works daily or weekly.

#### 2.3.6 DMS

This function allows manager to keep track of defects:

* Allow project manager to monitor defect and fix bug effort of team.
* Feature filter projects by date, status, and project, creator, assigned to.
* Team members can update defect status.

#### 2.3.7 Admin

2.4.7.1 Admin

* Manage Project
* Manage User
* Manage Project Team
* Manage Project Module

2.3.7.2 User

* Change User’s Information
* Reset Password
* Manage Project
* Manage Project Team
* Manage Project Module
* Manage Cost
* Manage Product
* Manage Work Order
* Manage Risk, Issue, Change Request

#### 2.3.8 Requirements

* Managing feature enables user to store and manage their requirement documents.
* Including: Add, update, remove
* User interface allows user to sort requirement by type, date, priority.
* Display completeness rate updated by user.

### 2.4 Use case model

This chapter describes PMS’s features and functions using abstract Use Cases and detail scenarios.

A use case is a description of a sequence of actions (including its variations) that the system carries out to create an observable result for an actor.

A scenario is a temporal sequence of interaction events among agent instances.

**Main Use Case Diagrams of the PMS**



This main Use Case Diagrams of the PMS show all main functions placing inside the system boundary and all actors that associate with those functions.

#### 2.4.1 General



##### 2.4.1.1 Log-in



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | General\_UC01 | |
| Name | Log-in Use Case | |
| Goal | Authenticate guests then authorize them. | |
| Actors | Guest | |
| Pre-conditions | None | |
| Post-conditions | None | |
| Main Flow | 1. User goes intoPMS Homepage.  3. Users enter user’s information: Username, Password and hit “Log-in” button. | 2. Display Home Screen and request user to log in.  4. Validate user’s information.  5.Display monitor screen to corresponding user. |
| Exception | If username or password is not correct, the PMSwill show error message and ask user to log in again. | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.1.2 Log-out



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | General\_UC02 | |
| Name | Log-out Use Case | |
| Goal | Provide Log-out function for User to check out | |
| Actors | User(include member, admin, project manager) | |
| Pre-conditions | Logged user | |
| Post-conditions | None | |
| Main Flow | 1. Users areusing PMSand click “Log-Out” button. | 2. Log user out and display Home Screen. |
| Exception | None | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.1.3 Change Password



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | General \_UC03 | |
| Name | Change Password Use Case | |
| Goal | This function allows user to change password | |
| Actors | User | |
| Pre-conditions | None | |
| Post-conditions | None | |
| Main Flow | 1. Select link “Change Your Information” | 2. Fill in “Old Password”, “New Password”, “Retype New Password” and then select button “Reset Password” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.1.4 Edit User’s Information



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | General\_UC04 | |
| Name | Edit User Use Case | |
| Goal | This function allows users to edit their profile | |
| Actors | User | |
| Pre-conditions | Logged user | |
| Post-conditions | None | |
| Main Flow | 1. Select link “Change your Information” | 2. Change User’s information then select button “Save Change” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.1.5 Register New Account



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | General\_UC05 | |
| Name | Register New Account Use Case | |
| Goal | This function allows guest to register a new account | |
| Actors | Guest | |
| Pre-conditions | None | |
| Post-conditions | None | |
| Main Flow | 1. User goes into PMS Homepage  3. Fill in User’s information then select button “Submit” | 2. Select link “Register New Account” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.1.6 Reset Password



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | General \_UC06 | |
| Name | Reset Password Use Case | |
| Goal | This function allows user to reset password when they forget password | |
| Actors | Guest | |
| Pre-conditions | None | |
| Post-conditions | None | |
| Main Flow | 1. User goes into PMS Homepage  3. Fill in Username and select button “Reset Password” | 2. Select link “Forget Your Password |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.2 Admin



##### 2.4.2.1 Admin Create New Project



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Admin\_UC01 | |
| Name | Admin Create new Project Use Case | |
| Goal | This function allows admin to create a new project. | |
| Actors | Admin | |
| Pre-conditions | Logged user with admin role | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab Project Management  3. Fill in Project’s information then select button “Submit” | 2. Select link “Create New Project” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.2.2 Admin Search Project



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | ADMIN\_UC02 | |
| Name | Admin Search Project Use Case | |
| Goal | This function allows admin to search project | |
| Actors | Admin | |
| Pre-conditions | Logged user with admin role | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab Project Management  3. Click button “Search” | 2. Select search condition |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.2.3 Admin Edit Project



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | ADMIN\_UC03 | |
| Name | Admin Edit Project Use Case | |
| Goal | This function allows admin to edit information of a project | |
| Actors | Admin | |
| Pre-conditions | Logged user with admin role | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab Project Management  3. Click button “Search”  5. Select link “Change Project Info” | 2. Select search condition  4. Choose the project want to edit  6. Change Project’s information then select button “Save Change” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.2.4 Admin Manage Module



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | ADMIN\_UC04 | |
| Name | Admin Mange Module Use Case | |
| Goal | This function allows admin to manage a project’s modules usage | |
| Actors | Admin | |
| Pre-conditions | Logged user with admin role | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab Project Management  3. Click button “Search”  5. Select link “Modules Management” | 2. Select search condition  4. Choose the project want to edit  6. Choose modules and then select button “OK” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.2.5 Admin Team Management



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | ADMIN\_UC05 | |
| Name | Admin Team Management Use Case | |
| Goal | This function allows admin to manage a project’s team members | |
| Actors | Admin | |
| Pre-conditions | Logged user with admin role | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab Project Management  3. Click button “Search”  5. Select link “Team Management” | 2. Select search condition  4. Choose the project want to edit  6. Edit Project’s team member then select button “Save Change” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.2.6 Admin Search User



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | ADMIN\_UC06 | |
| Name | Admin Search User Use Case | |
| Goal | This function allows admin to search user | |
| Actors | Admin | |
| Pre-conditions | Logged user with admin role | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab User Management  3. Click button “Search” | 2. Select search condition |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.2.7 Admin Create User



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | ADMIN\_UC07 | |
| Name | Admin Create User Use Case | |
| Goal | This function allows admin to create new user | |
| Actors | Admin | |
| Pre-conditions | Logged user with admin role | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab User Management  3. Fill in User’s information then select button “Submit” | 2. Select link “Create New User” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.2.8 Admin Edit User



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | ADMIN\_UC08 | |
| Name | Admin Edit User Use Case | |
| Goal | This function allows admin to edit information of an user | |
| Actors | Admin | |
| Pre-conditions | Logged user with admin role | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab User Management  3. Click button “Search”  5. Change User’s information then select button “Save Change” | 2. Select search condition  4. Choose the user want to edit |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.2.9 Admin Change User’s Status



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | ADMIN\_UC09 | |
| Name | Admin Change User’s Status Use Case | |
| Goal | This function allows admin to active, inactive or delete an user | |
| Actors | Admin | |
| Pre-conditions | Logged user with admin role | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab User Management  3. Click button “Search” | 2. Select search condition  4. Select button “Active”, ”Inactive” or “Delete” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.3 Planner



##### 2.4.3.1 Filter Task



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Planner\_UC01 | |
| Name | Filter Task | |
| Goal | This function allows Project Manager and Team Member to filter tasks that are displayed on screen. | |
| Actors | Project Manager; Team Member | |
| Pre-conditions | Users must log in with role “Project Manager” or “Team Member”; Users must go to Planner page | |
| Post-conditions | N/A | |
| Main Flow | 1. Select conditions to filter and click “Search” button. | 2. Show task list after filtering. |
| Exception | N/A | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business rule | N/A | |
| Priority | Normal | |

##### 2.4.3.2 Add Task



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Planner\_UC02 | |
| Name | Add Task | |
| Goal | This function allows Project Manager to plan and add specific tasks to his/her project. | |
| Actors | Project Manager | |
| Pre-conditions | Users must log in with role “Project Manager”; Users must go to Planner page | |
| Post-conditions | N/A | |
| Main Flow | 1. Click “Add New Task” button  3. Fill up task’s information.  6. Click “Save” button. | 2. Redirect to Task Adding Page.  3. Validate task’s information.  5. Display “Complete!” message. |
| Exception | 3A: if task information is not correct, show error message and ask to input again. | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business rule | N/A | |
| Priority | High | |

##### 2.4.3.3 Update Task



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Planner\_UC03 | |
| Name | Update Task | |
| Goal | This function allows Project Manager and Team Members to update status of tasks or change information of tasks. (modifying scope depend on authority of user) | |
| Actors | Project Manager; Team Member | |
| Pre-conditions | Users must log in with role “Project Manager” or “Team Member”; Users must go to Planner page | |
| Post-conditions | N/A | |
| Main Flow | 1. Filer Planner (optional).  2. Click on a task tile.  4. Update task’s information.  6. Click “Save” button. | 3.Redirect to Task Updating Page.  5. Validate task’s information. |
| Exception | 5A: if task information is not correct, show error message and ask to input again. | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business rule | N/A | |
| Priority | High | |

##### 2.4.3.4 Delete Task



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Planner\_UC04 | |
| Name | Delete Task | |
| Goal | This function allows Project Manager delete task on Planner page. | |
| Actors | Project Manager | |
| Pre-conditions | Users must log in with role “Project Manager”; Users must go to Planner page | |
| Post-conditions |  | |
| Main Flow | 1. Select a task, which he/she want to delete.  2. Click “Delete” button. | 3. Display “Delete Successfully” message. |
| Exception | N/A | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business rule | N/A | |
| Priority | High | |

##### 2.4.3.5 Import Task



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Planner\_UC05 | |
| Name | Import Task | |
| Goal | This function allows Project Manager to import task from Microsoft Project file. | |
| Actors | Project Manager | |
| Pre-conditions | Users must log in with role “Project Manager”; Users must go to Planner page | |
| Post-conditions | N/A | |
| Main Flow | 1. User clicks “Import” button.  3. User selects a \*.mpp file from PC and clicks “OK” button. | 2. Display window to user select \*.mpp file.  4. Import tasks in the file to Planner. |
| Exception | 3A: if the file is not correct, show error message and ask to input again. | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business rule | N/A | |
| Priority | Low | |

#### 2.4.4 Project Eye



##### 2.4.4.1 Create Project



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC01 | |
| Name | Create new Project Use Case | |
| Goal | This function allows user to create a new project. | |
| Actors | User | |
| Pre-conditions | Logged user | |
| Post-conditions | None | |
| Main Flow | 1. Select link “Create New Project” | 2. Fill in Project’s information then select button “Submit” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.4.2 Edit Project



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC02 | |
| Name | Edit Project Use Case | |
| Goal | This function allows project managers to edit information of their project | |
| Actors | Project Manager | |
| Pre-conditions | Logged user that is a project manager | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Dash Board"  3. Select link “Change Project Info” | 2. Choose the project want to edit  4. Change Project’s information then select button “Save Change” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.4.3 Manage Module



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC03 | |
| Name | Manage Module Use Case | |
| Goal | This function allows project manager to manage a project’s modules usage | |
| Actors | Project Manager | |
| Pre-conditions | Logged user that is a project manager | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Dash Board”  3. Select link “Modules Management” | 2. Choose the project want to edit  4. Choose modules and then select button “OK” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.4.5 Team Management



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC04 | |
| Name | Team Management Use Case | |
| Goal | This function allows project manager to manage a project’s team members | |
| Actors | Project Manager | |
| Pre-conditions | Logged user that is a project manager | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Dash Board”  3. Select link “Team Management” | 2. Choose the project want to edit  4. Edit Project’s team member then select button “Save Change” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.5 Change Budget



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC05 | |
| Name | Change Budget Use Case | |
| Goal | This function allows project manager to manage a project’s budget | |
| Actors | Project Manager | |
| Pre-conditions | Logged user that is a project manager | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Cost”  3. Select button “OK” on the commit window. | 2. Fill the budget value then select button “Change Budget” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.6 Add Expense



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC06 | |
| Name | Add Expense Use Case | |
| Goal | This function allows project manager to add a project’s expense | |
| Actors | Project Manager | |
| Pre-conditions | Logged user that is a project manager | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Cost”  3. Fill in the expense’s information then select button “Submit” | 2. Select “Add new Expense” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.7 Edit Expense



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC07 | |
| Name | Edit Expense Use Case | |
| Goal | This function allows project manager to edit a project’s expense | |
| Actors | Project Manager | |
| Pre-conditions | Logged user that is a project manager | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Cost”  3. Fill in the expense’s information then select button “Save Change” | 2. Select the expense want to edit |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.8 Delete Expense



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC08 | |
| Name | Delete Expense Use Case | |
| Goal | This function allows project manager to delete a project’s expense | |
| Actors | Project Manager | |
| Pre-conditions | Logged user that is a project manager | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Cost”  3. Select button “Delete” | 2. Select the checkbox at the begin of the expenses’ row you want to delete |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.9 Add Risk, Issue



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC09 | |
| Name | Add Risk, Issue Use Case | |
| Goal | This function allows project manager and team member to add a project’s risk or issue | |
| Actors | Project Manager, Team Member | |
| Pre-conditions | Logged user that is a project manager or team member | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Risk, Issue”  3. Fill in the information then select button “Submit” | 2. Select link “Add new Risk” or “Add new Issue” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.10 Edit Risk, Issue



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC10 | |
| Name | Edit Risk, Issue Use Case | |
| Goal | This function allows project manager and team member to edit a project’s risk, issue | |
| Actors | Project Manager, Team member | |
| Pre-conditions | Logged user that is a project manager or team member | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Risk, Issue”  3. Fill in the information then select button “Save Change” | 2. Select the risk or issue want to edit |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.11 Delete Risk, Issue



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC11 | |
| Name | Delete Risk, Issue Use Case | |
| Goal | This function allows project manager and team member to delete a project’s risk, issue | |
| Actors | Project Manager, Team member | |
| Pre-conditions | Logged user that is a project manager or team member | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Risk, Issue”  3. Select button “Delete” | 2. Select the checkbox at the begin of the row you want to delete |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.12 Add Change Request



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC12 | |
| Name | Add Change Request Use Case | |
| Goal | This function allows project manager and team member to add a project’s change request | |
| Actors | Project Manager, Team Member | |
| Pre-conditions | Logged user that is a project manager or team member | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Change Request”  3. Fill in the request’s information then select button “Submit” | 2. Select link “Add new Change Request” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.13 Edit Change Request



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC13 | |
| Name | Edit Change Request Use Case | |
| Goal | This function allows project manager and team member to edit a project’s change request | |
| Actors | Project Manager, Team member | |
| Pre-conditions | Logged user that is a project manager or team member | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Change Request”  3. Fill in the request’s information then select button “Save Change” | 2. Select the request want to edit |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.14 Delete Change Request



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC14 | |
| Name | Delete Change Request Use Case | |
| Goal | This function allows project manager and team member to delete a project’s change request | |
| Actors | Project Manager, Team member | |
| Pre-conditions | Logged user that is a project manager or team member | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Change Request”  3. Select button “Delete” | 2. Select the checkbox at the begin of the row you want to delete |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.15 Add Product



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC15 | |
| Name | Add Product Use Case | |
| Goal | This function allows project manager and team member to add a project’s product | |
| Actors | Project Manager, Team Member | |
| Pre-conditions | Logged user that is a project manager or team member | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Product”  3. Fill in the product’s information then select button “Submit” | 2. Select link “Add new Product” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

2.4.16 Edit Product  
  


Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC16 | |
| Name | Edit Product Use Case | |
| Goal | This function allows project manager and team member to edit a project’s product | |
| Actors | Project Manager, Team member | |
| Pre-conditions | Logged user that is a project manager or team member | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Product”  3. Fill in the product’s information then select button “Save Change” | 2. Select the product want to edit |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.17 Delete Product



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC17 | |
| Name | Delete Product Use Case | |
| Goal | This function allows project manager and team member to delete a project’s product | |
| Actors | Project Manager, Team member | |
| Pre-conditions | Logged user that is a project manager or team member | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Product”  3. Select button “Delete” | 2. Select the checkbox at the begin of the row you want to delete |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

2.4.18 Add Stage  
  


Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC18 | |
| Name | Add Stage Use Case | |
| Goal | This function allows project manager to add a project’s stage | |
| Actors | Project Manager | |
| Pre-conditions | Logged user that is a project manager | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Work Order”  3. Fill in the stage’s information then select button “Submit” | 2. Select “Add new Stage” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.19 Edit Stage



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC19 | |
| Name | Edit Stage Use Case | |
| Goal | This function allows project manager to edit a project’s stage | |
| Actors | Project Manager | |
| Pre-conditions | Logged user that is a project manager | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Work Order”  3. Fill in the stage’s information then select button “Save Change” | 2. Select the Stage want to edit |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.20 Delete Stage



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC20 | |
| Name | Delete Stage Use Case | |
| Goal | This function allows project manager to delete a project’s stage | |
| Actors | Project Manager | |
| Pre-conditions | Logged user that is a project manager | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Work Order”  3. Select button “Delete” | 2. Select the checkbox at the begin of the stages’ row you want to delete |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.21 Add Deliverable



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC21 | |
| Name | Add Deliverable Use Case | |
| Goal | This function allows project manager to add a project’s deliverable | |
| Actors | Project Manager | |
| Pre-conditions | Logged user that is a project manager | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Work Order”  3. Fill in the deliverable’s information then select button “Submit” | 2. Select “Add new Deliverable” |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.22 Edit Deliverable



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC22 | |
| Name | Edit Deliverable Use Case | |
| Goal | This function allows project manager to edit a project’s deliverable | |
| Actors | Project Manager | |
| Pre-conditions | Logged user that is a project manager | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Work Order”  3. Fill in the deliverable’s information then select button “Save Change” | 2. Select the Deliverable want to edit |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.23 Delete Deliverable



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC23 | |
| Name | Delete Deliverable Use Case | |
| Goal | This function allows project manager to delete a project’s deliverable | |
| Actors | Project Manager | |
| Pre-conditions | Logged user that is a project manager | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Work Order”  3. Select button “Delete” | 2. Select the checkbox at the begin of the deliverables’ row you want to delete |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.24 View Info



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC24 | |
| Name | View Info Use Case | |
| Goal | This function allows project manager and team member to view a project’s information | |
| Actors | Project Manager, Team member | |
| Pre-conditions | Logged user that is a project manager or team member | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Dash Board” | 2. Select a project’s name |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.25 Report



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Project Eye\_UC25 | |
| Name | Report Use Case | |
| Goal | This function allows project manager and team member to view a project’s report | |
| Actors | Project Manager, Team member | |
| Pre-conditions | Logged user that is a project manager or team member | |
| Post-conditions | None | |
| Main Flow | 1. Choose tab “Dash Board”  3. Choose tab “Report” | 2. Select a project’s name |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

7.5 Dashboard



7.5.1 Filter Project



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Dashboard \_UC01 | |
| Name | Filter Project | |
| Goal | This function allows user to filter projects, which appear on Dashboard. Project on system can be filtered by status or category. | |
| Actors | Project Manager | |
| Pre-conditions | Users must log in with role “Project Manager”; Users must go to Dashboard page. | |
| Post-conditions | N/A | |
| Main Flow | 1. Select conditions to filter and Click “Search” button. | 2. Display filtered project list. |
| Exception | N/A | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business rule | N/A | |
| Priority | Normal | |

7.5.2 Export Dashboard



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Dashboard \_UC02 | |
| Name | Export Dashboard | |
| Goal | This function allows Project Manager to export Dashboard to Microsoft Excel file. | |
| Actors | Project Manager | |
| Pre-conditions | Users must log in with role “Project Manager”; Users must go to Dashboard page. | |
| Post-conditions | N/A | |
| Main Flow | 1. Filter project list (optional).  2. User Click on “Export” button. | 3. Export file. |
| Exception | N/A | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business rule | N/A | |
| Priority | Normal | |

7.6 Requirement



#### 2.4.26 Add Requirement



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Requirement \_UC01 | |
| Name | Add Requirement Use Case | |
| Goal | This function allows user add new requirement or CR (Change Request). | |
| Actors | Project Manager | |
| Pre-conditions | Users must log in with role “Project Manager”; | |
| Post-conditions | N/A | |
| Main Flow | 1. Go to Requirement page. Click “Add” button.  3. Fill needed information and click “summit” button. | 2. Display page with required fieldfor user to fill information.    4. Save requirement information. |
| Exception | N/A | |
| Open Issues | How to save requirement documents. (Link or database) | |
| Relationship | N/A | |
| Business rule | N/A | |
| Priority | 5 | |

#### 2.4.27 Update Requirement



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Requirement \_UC02 | |
| Name | Update Requirement Use Case | |
| Goal | This function allows user to updatedocument, status. | |
| Actors | Project Manager, member | |
| Pre-conditions | N/A | |
| Post-conditions | N/A | |
| Main Flow | 1. Go to Requirement page. Choose requirement, Click “Update” button.  3. Fill needed information and click “Save” button. | 2. Display page with information for user to modify.    4. Save updated information. |
| Exception | N/A | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business rule | N/A | |
| Priority | 5 | |

#### 2.4.28 Delete Requirement



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Requirement \_UC03 | |
| Name | Add Requirement Use Case | |
| Goal | This function allows user to delete requirement. | |
| Actors | Project Manager | |
| Pre-conditions | Users must log in with role “Project Manager”; | |
| Post-conditions | N/A | |
| Main Flow | 1. Go to Requirement page. Choose requirement, Click “Delete” button.  3. Click “Yes” button. | 2. Display confirm box.    4. Delete Requirement. |
| Exception | N/A | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business rule | N/A | |
| Priority | 5 | |

#### 2.4.29 Sort Requirements



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Requirement \_UC04 | |
| Name | Sort Requirements Use Case | |
| Goal | This function allows user to sort, filter requirement by Size, Priority or Date. | |
| Actors | Project Manager, member | |
| Pre-conditions | N/A | |
| Post-conditions | N/A | |
| Main Flow | 1. Go to Requirement page. Choose kind of filter (type, date, priority). | 2. Sort and Display new list. |
| Exception | N/A | |
| Open Issues | Sort by user and last modified? | |
| Relationship | N/A | |
| Business rule | N/A | |
| Priority | 5 | |

7.7 Defect Management System



#### 2.4.30 Search defect



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | DMS\_UC01 | |
| Name | Search defect Use Case | |
| Goal | This function allows users can search defect.  Defect has 3 kinds : open defect, closed defect, leakage | |
| Actors | Member | |
| Pre-conditions | Logged user | |
| Post-conditions | None | |
| Main Flow | 1. Users login into DMS system 2. Click button search | 1. Select search condition |
| Exception | None | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.31 Add defect



Figure 20 Search defect Use Case model

Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | DMS\_UC02 | |
| Name | Add defect Use Case | |
| Goal | This function allows users can add defect.  When defect was added, member become creator, defect change status to opened | |
| Actors | Member | |
| Pre-conditions | Logged user | |
| Post-conditions | None | |
| Main Flow | 1Users login into DMS system | 1. Click button add new |
| Exception | None | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.32 Update defect



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | DMS\_UC03 | |
| Name | Update defect Use Case | |
| Goal | This function allows users can update defect.  Creator can change status of defect from opened to cancelled or corrected.  Assigned member can change status of defect to opened, canceled or closed. | |
| Actors | Member | |
| Pre-conditions | Logged user | |
| Post-conditions | None | |
| Main Flow | 1Users login into DMS system  3..Click button update | 1. Select defect |
| Exception | None | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

#### 2.4.33 Timesheet Module



##### 2.4.33.1 Search Timesheet



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Timesheet\_UC01 | |
| Name | Search Timesheet Use Case | |
| Goal | This function allows users can search timesheet. | |
| Actors | Add Timesheet | |
| Pre-conditions | User logins must be member of project | |
| Post-conditions | None | |
| Main Flow | 1. Users logins to Timesheet system.  3. Click button Search | 2. Select search condition. |
|  |  | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.33.2 Add timesheet



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Timesheet\_UC02 | |
| Name | Add Timesheett Use Case | |
| Goal | This function allows users can add a new timesheet record | |
| Actors | Add Timesheet | |
| Pre-conditions | User logins must be member of project | |
| Post-conditions | None | |
| Main Flow | 1. Users logins to Timesheet system. | 2. Click button add new. |
| Exception | None | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.33.3 Update timesheet



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Timesheet\_UC03 | |
| Name | Update timesheet Use Case | |
| Goal | This function allows users can update timesheet records. | |
| Actors | Member , Project Manager | |
| Pre-conditions | User logins must be member of project | |
| Post-conditions | None | |
| Main Flow | 1. User logins to Timesheet system.  3. Click button update | 2. Select timesheet record to update |
| Exception | None | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.33.4 Delete timesheet



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Timesheet\_UC04 | |
| Name | Delete timesheet Use Case | |
| Goal | This function allows users can delete timesheet records. | |
| Actors | Member , Project Manager | |
| Pre-conditions | User logins must be member of project | |
| Post-conditions | None | |
| Main Flow | 1. User logins to Timesheet system.  3. Click button delete | 2. Select timesheet record to delete |
| Exception | None | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

##### 2.4.33.5 Approve/ Reject timesheet



Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Timesheet\_UC05 | |
| Name | Approve/ Reject timesheet Use Case | |
| Goal | Allow project manager to approves or reject members‘ timesheet records | |
| Actors | Project manager | |
| Pre-conditions | Logged users as role project manager | |
| Post-conditions | None | |
| Main Flow | 1. User logins to Timesheet system  as role project manager  3. Click button approve or reject | 2. Select Timesheet to approve or reject |
| Exception | N/A | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business Rule | N/A | |
| Priority | N/A | |

7.9Report



7.9.1 Print Report

Use Case scenario:

|  |  |  |
| --- | --- | --- |
| User Case ID | Report\_UC01 | |
| Name | Print Report | |
| Goal | This function allows Project Manager and Team Member to view and print report about planner, timesheet, DMS or overall of the project. | |
| Actors | Project Manager; Team Member | |
| Pre-conditions | Users must log in with role “Project Manager” or “Team Member” ; Users must go to Report page | |
| Post-conditions | N/A | |
| Main Flow | 1. Select report by clicking on name of report (planner, timesheet, DMS...).  3. User clicks “Print” button. | 2. Display Report  4. Export Report. |
| Exception | N/A | |
| Open Issues | N/A | |
| Relationship | N/A | |
| Business rule | N/A | |
| Priority | High | |

### 2.5 NON-FUNCTIONAL Requirements

This chapter of the document describes all the non-functions requirement of the OOPMS.

The PMS’s usability, availability, performance are very important factors to ensure that the system operate effectively and keep hundred miners safe.

#### 2.5.1 Usability

The OOPMS usability is the key factor to ensure that the system run exactly as well as the operators can control the system efficiently.

The OOPMS application shall provide clear, friendly and easy interface to operate so that system users have to spend no more than one hour learning to use the system.

##### 2.5.1.1 Background knowledge

OOPMS users assume to have very basic knowledge at using computer systems.

##### 2.5.1.2 Training

Operators’ training time requirement: 60 minutes.

#### 2.5.2 Reliability

##### 2.5.2.1 Availability

Whenever the system is during operation time no matter how many people working inside, the OOPMS has to be ON.

##### 2.5.2.2 Mean Time between Failures (MTBF)

Mean Time between Failures (MTBF): more than 6 months.

##### 2.5.2.3 Mean Time to Repair (MTTR)

Mean Time To Repair (MTTR): less than 48 hours.

##### 2.5.2.4 Accuracy

Accuracy: 100%

##### 2.5.2.5 Maximum Bugs and Defect Rate

Maximum Bugs and Defect Rate: 0.3 bugs per thousand lines of code (0.3 bugs/KLOC).

##### 2.5.2.6 Critical Bugs

Critical bugs:

+ Loss of Log: No

+ Unable to operate any function: No

#### 2.5.3 Performance

##### 2.5.3.1 Response Time

Response time for a respond:

Average: 50 milliseconds

Maximum: 80 milliseconds

##### 2.5.3.2 Capacity

Only one operator at the same time.

##### 2.5.3.3 Resource utilization

Memory:

+ 512MB of RAM

Operating System:

+ Microsoft Windows XP or newer version

+ Mac OS X 10.0 or newer version

+ Linux 3.5 or newer version

#### 2.5.4 Supportability

##### 2.5.4.1 Coding standards

According to “Standard Java Coding Convention”

– 09be-HD/PM/HDCV/FSOFT - Version 1/1.

##### 2.5.4.2 Maintenance Utilities

Support working hour’s phone call technical support: 8:00 to 16:00 from Monday to Friday.

##### 2.5.5 Design Constraints

Coding standard:

+ This application developed in Java programming language, version J2EE 6.

Software process requirements:

+ The software process shall confront to the CMMI 5 standard.

Developmental tools:

+ This application developed using Eclipse from Sun Microsystems.

##### 2.5.5.1 Software Languages

GUI, Help documents, all other support documents are in English.

##### 2.5.5.2 Software Process Requirement

The software process has to meet the CMMI 5 standard.

##### 2.5.5.3 Development Tools

The OOPMS developed using Eclipse 3.6, SQA activity using FSOFT template and j-unit.

#### 2.5.6 On-line User Documentation and Help System Requirements

The deployment of the application shall be provided by a technical agent include full help document for user.

#### 2.5.7 Purchased Components

None

#### 2.5.8 Interfaces

##### 2.5.8.1 User Interfaces

Refer to Detail Design.

##### 2.5.8.2Hardware Interfaces

N/A

#### 2.5.9 Licensing Requirements

The OOPMS is only applicable for Project Management Department, which legally buy this product. No other use is legal.

#### 2.5.10 Legal, Copyright, and Other Notices

OOPMS licenses under the [Apache License, Version 2.0](http://www.apache.org/licenses/LICENSE-2.0) as approved by the Open Source Initiative (OSI), an [OSI-certified](http://www.opensource.org/docs/osd) ("open") and [Gnu/FSF-recognized](http://www.gnu.org/licenses/license-list.html) ("free") license.

#### 2.5.11 Applicable Standards

The system shall confront to the ISO 90003:2001 standard.

# Software Design Description

## Introduction

### 1.1 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.

### 1.2 Scope

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

### 1.3 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.
* **Database Design**: This section describesin detail how data is structured and manipulated in this application.
* **CRC Card Model:** This is to describe modules’ responsibilities and its cooperators.
* **Application Security**: This section describles security matrix in detail
* **Detail Function Design**: This section describe in detail how features are developed and work.
* **Interface Design:** This section describesin detail how UI is designed in general ( layout , theme ).
* **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.

### 1.4 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 |

### 1.5 References

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

Table 1.1: List of References

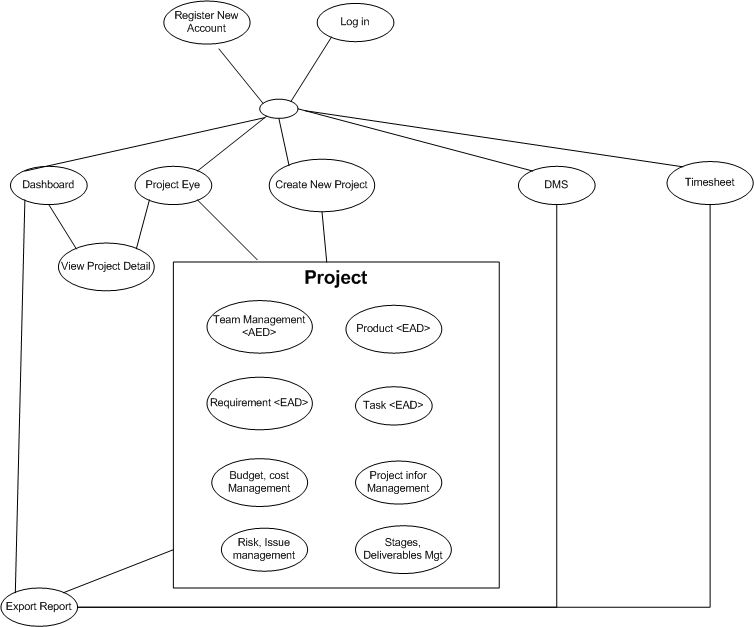
## Architecture design

### 2.1 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

#### 2.2.1 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.

#### 2.2.2 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.

#### 2.2.3 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.

#### 2.2.4 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



#### 2.3.1 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 .

#### 2.3.2 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.

#### 2.3.4 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.

#### 2.3.5 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*.

#### 2.3.6 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.

#### 2.3.7 Utils

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

#### 2.3.8 Logging

This package includes implemented logging classes.

## Technical Solutions

### 3.1 Exception handling mechanism

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

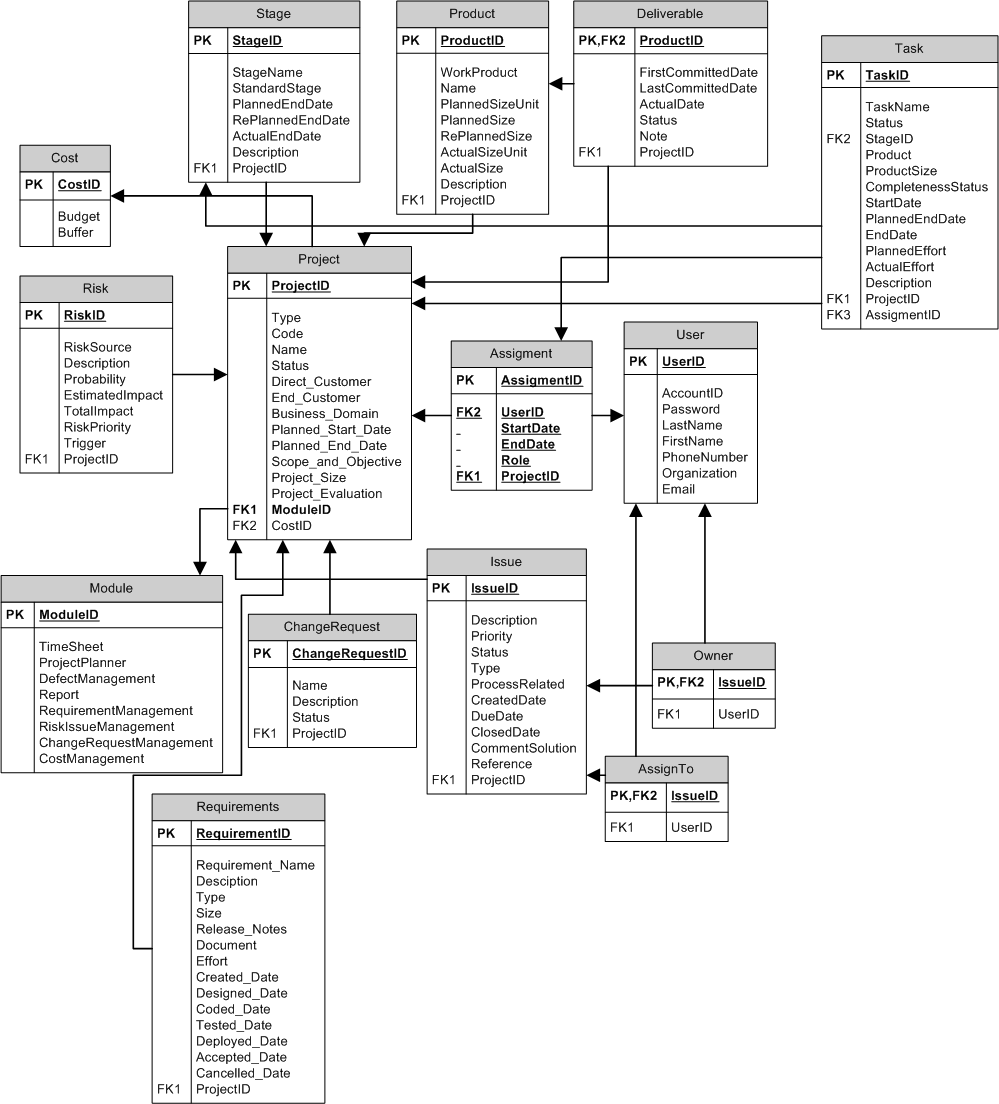
Whenever an exception raise , system will log exception detail to Windows event and redirect user to Error page to display standard error message!

### 3.2 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

### 4.1 Entity Relationship Diagram



### 4.2 Schema

**Overview**

|  |  |  |
| --- | --- | --- |
| **#** | **Name** | **Description** |
| 1 | Project | Project with its information: name, type, code.. |
| 2 | Stage | Stage of software development process |
| 3 | Product | Product from development process |
| 4 | Deliverable | Product to be summited to customers |
| 5 | Task | Task assign to team members |
| 6 | Cost | Cost management |
| 7 | Risk | Risk management |
| 8 | Assignment | Assignment to user |
| 9 | User | User |
| 10 | Module | Module of project |
| 11 | ChangeRequest | Change Request management |
| 12 | Issue | Issue Management |
| 13 | Owner | Owner of issue |
| 14 | AssignTo | Assigned member of issue |
| 15 | Requirement | Requirement Management |

### 4.3 Detail Schema

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Project table** | | | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | | **Null** | **Unique** | **P/F Key** | **Description** | |
| 1 | ProjectID | CHAR | 10 | |  |  | PK | PK – ID of project | |
| 2 | Type | NVARCHAR | 20 | |  |  |  | Type of project | |
| 3 | Code | NVARCHAR | 20 | |  |  |  | Project code | |
| 4 | Name | NVARCHAR | 100 | |  |  |  | Name of project | | of requirement | |  | |  | |  |  |
| 5 | Status | NVARCHAR | 10 | |  |  |  | Status of project | |  | |  | |  | |  |  |
| 6 | DirectCustomer | NVARCHAR | 20 | | X |  |  | Direct customer of project | |  | |  | |  | |  |  |
| 7 | EndCustomer | NVARCHAR | 20 | | X |  |  | End customer of project | |  | |  | |  | |  |  |
| 8 | BusinessDomain | NVARCHAR | 20 | | X |  |  | Business domain of project | |  | |  | |  | |  |  |
| 9 | PlannedStartDate | DATETIME |  | |  |  |  | Planned start date of project | |  | |  | |  | |  |  |
| 10 | PlannedEndDate | DATETIME |  | |  |  |  | Planned end date of project | |  | |  | |  | |  |  |
| 11 | ScopeAndObjective | NVARCHAR | 200 | | X |  |  | Scope and objective of project | |  | |  | |  | |  |  |
| 12 | ProjectSize | INTEGER |  | | X |  |  | Size of project | |  | |  | |  | |  |  |
| 13 | ProjectEvaluation | INTEGER |  | | X |  |  | Project Evaluation | |  | |  | |  | |  |  |
| 14 | ModuleID | CHAR | 10 | |  |  | FK | FK ModuleID of project | |  | |  | |  | |  |  |
| 15 | CostID | CHAR | 10 | |  |  | FK | FK CostID of project | |  | |  | |  | |  |  |
|  | | | |  | | | | |  | |  | |  | |
| **Stage table** | | | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | | **Null** | **Unique** | **P/F Key** | **Description** | |
| 1 | StageID | CHAR | 10 | |  |  | PK | PK – ID of stage | |
| 2 | StageName | NVARCHAR | 20 | |  |  |  | Name of stage | |
| 3 | StandardStage | NVARCHAR | 20 | |  |  |  | Standard Stage | |
| 4 | PlannedEndDate | DATETIME |  | |  |  |  | Planned end date of stage | |
| 5 | RePlannedEndDate | DATETIME |  | | X |  |  | Re planned end date of stage | |
| 6 | ActualEndDate | DATETIME |  | | X |  |  | Actual end date of stage | |
| 7 | Description | NVARCHAR | 200 | | X |  |  | Description of stage | |
| 8 | DelFlag | BOOLEAN |  | |  |  |  | Deleted flag of stage | |
| 9 | ProjectID | CHAR | 10 | |  |  | FK | FK ProjectID of stage | |
|  | | | |  | | | | |  | |
| **Product table** | | | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | | **Null** | **Unique** | **P/F Key** | **Description** | |
| 1 | ProductID | CHAR | 10 | |  |  | PK | PK – ID of product | |
| 2 | WorkProduct | NVARCHAR | 20 | |  |  |  | Work of product | |
| 3 | Name | NVARCHAR | 20 | |  |  |  | Name of product | |
| 4 | PlannedSizeUnit | NVARCHAR | 20 | |  |  |  | Planned size unit of product | | of requirement | |  | |  | |  |  |
| 5 | PlannedSize | INTEGER |  | |  |  |  | Planned size of product | |  | |  | |  | |  |  |
| 6 | RePlannedSize | INTEGER |  | | X |  |  | Re planned size of product | |  | |  | |  | |  |  |
| 7 | ActualSizeUnit | NVARCHAR | 20 | | X |  |  | Actual size unit of product | |  | |  | |  | |  |  |
| 8 | ActualSize | INTEGER |  | | X |  |  | Actual size of product | |  | |  | |  | |  |  |
| 9 | Description | NVARCHAR | 200 | | X |  |  | Description of product | |  | |  | |  | |  |  |
| 10 | DelFlag | BOOLEAN |  | |  |  |  | Deleted flag of product | |
| 11 | ProjectID | CHAR | 10 | |  |  | FK | FK ProjectID of product | |  | |  | |  | |  |  |
|  | | | |  | | | | |  | |  | |  | |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Deliverable table** | | | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | | **Null** | **Unique** | **P/F Key** | **Description** | |
| 1 | ProductID | CHAR | 10 | |  |  | PK/FK | PK – ID of deliverable | |
| 2 | FirstCommittedDate | DATETIME |  | |  |  |  | First committed date of deliverable | |
| 3 | LastCommittedDate | DATETIME |  | | X |  |  | Last committed date of deliverable | |
| 4 | ActualDate | DATETIME |  | | X |  |  | Actual date of deliverable | | of requirement | |  | |  | |  |  |
| 5 | Status | NVARCHAR | 20 | |  |  |  | Status of deliverable | |  | |  | |  | |  |  |
| 6 | Note | NVARCHAR | 200 | | X |  |  | Note of deliverable | |  | |  | |  | |  |  |
| 7 | DelFlag | BOOLEAN |  | |  |  |  | Deleted flag of deliverable | |
| 8 | ProjectID | CHAR | 10 | |  |  | FK | FK ProjectID of deliverable | |  | |  | |  | |  |  |
|  | | | |  | | | | |  | |  | |  | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk table** | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | **Null** | **Unique** | **P/F Key** | **Description** |
| 1 | RiskID | CHAR | 10 |  |  | PK | PK – ID of risk |
| 2 | RiskSource | NVARCHAR | 50 |  |  |  | Source of risk |
| 3 | Description | NVARCHAR | 200 |  |  |  | Description of risk |
| 4 | Probability | INTEGER |  |  |  |  | Probability of risk | of requirement |  |  |  |  |
| 5 | EstimatedImpact | NVARCHAR | 10 |  |  |  | Estimated impact of risk |  |  |  |  |  |
| 6 | TotalImpact | INTEGER |  | X |  |  | Total Impact of risk |  |  |  |  |  |
| 7 | RiskPriority | INTEGER |  |  |  |  | Priority of risk |  |  |  |  |  |
| 8 | Trigger | NVARCHAR | 200 | X |  |  | Trigger of risk |  |  |  |  |  |
| 9 | DelFlag | BOOLEAN |  |  |  |  | Deleted flag of risk |
| 10 | ProjectID | CHAR | 10 |  |  | FK | FK ProjectID of risk |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Issue table** | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | **Null** | **Unique** | **P/F Key** | **Description** |
| 1 | IssueID | CHAR | 10 |  |  | PK | PK – ID of issue |
| 2 | Description | NVARCHAR | 20 |  |  |  | Description of issue |
| 3 | Priority | NVARCHAR | 20 |  |  |  | Priority of issue |
| 4 | Status | NVARCHAR | 10 |  |  |  | Status of issue | of requirement |  |  |  |  |
| 5 | Type | NVARCHAR | 10 |  |  |  | Type of issue |  |  |  |  |  |
| 6 | ProcessRelated | NVARCHAR | 20 |  |  |  | Process related to issue |  |  |  |  |  |
| 7 | CreatedDate | NVARCHAR | 20 |  |  |  | Create date of issue |  |  |  |  |  |
| 8 | DueDate | NVARCHAR | 20 |  |  |  | Due date of issue |  |  |  |  |  |
| 9 | ClosedDate | DATETIME |  | X |  |  | Closed date of issue |  |  |  |  |  |
| 10 | CommentSolution | DATETIME |  | X |  |  | Comment solution for issue |  |  |  |  |  |
| 11 | Reference | NVARCHAR | 200 | X |  |  | Reference of issue |  |  |  |  |  |
| 12 | DelFlag | BOOLEAN |  |  |  |  | Deleted flag of issue |
| 13 | ProjectID | CHAR | 10 |  |  | FK | FK ProjectID of issue |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ChangeRequest table** | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | **Null** | **Unique** | **P/F Key** | **Description** |
| 1 | ChangeRequestID | CHAR | 10 |  |  | PK | PK – ID of change request |
| 2 | Name | NVARCHAR | 20 |  |  |  | Name of change request |
| 3 | Description | NVARCHAR | 200 |  |  |  | Description of change request |
| 4 | Status | NVARCHAR | 10 |  |  |  | Status of change request | of requirement |  |  |  |  |
| 5 | DelFlag | BOOLEAN |  |  |  |  | Deleted flag of change request |
| 6 | ProjectID | CHAR | 10 |  |  | FK | FK ProjectID of issue |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Module table** | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | **Null** | **Unique** | **P/F Key** | **Description** |
| 1 | ModuleID | CHAR | 10 |  |  | PK | PK – ID of Module |
| 2 | TimeSheet | BOOLEAN |  |  |  |  | Usage of time sheet |
| 3 | ProjectPlanner | BOOLEAN |  |  |  |  | Usage of project planner |
| 4 | DefectManagement | BOOLEAN |  |  |  |  | Usage of defect management | of requirement |  |  |  |  |
| 5 | Report | BOOLEAN |  |  |  |  | Usage of report |  |  |  |  |  |
| 6 | RequirementManagement | BOOLEAN |  |  |  |  | Usage of requirement management |  |  |  |  |  |
| 7 | RiskIssueManagement | BOOLEAN |  |  |  |  | Usage of risk issue management |  |  |  |  |  |
| 8 | ChangeRequestManagement | BOOLEAN |  |  |  |  | Usage of change request management |  |  |  |  |  |
| 9 | CostManagement | BOOLEAN |  |  |  |  | Usage of cost management |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Owner table** | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | **Null** | **Unique** | **P/F Key** | **Description** |
| 1 | IssueID | CHAR | 10 |  |  | PK/FK | PK/FK – ID of Owner From IssueID of Issue table |
| 2 | UserID | CHAR | 10 |  |  | FK | FK UserID of Owner |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **AssignTo table** | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | **Null** | **Unique** | **P/F Key** | **Description** |
| 1 | IssueID | CHAR | 10 |  |  | PK/FK | PK/FK – ID of AssignTo From IssueID of Issue table |
| 2 | UserID | CHAR | 10 |  |  | FK | FK UserID of AssignTo |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **User table** | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | **Null** | **Unique** | **P/F Key** | **Description** |
| 1 | UserID | CHAR | 10 |  |  | PK | PK – ID of user |
| 2 | AccountID | NVARCHAR | 20 |  | X |  | AccountID of user |
| 3 | Password | NVARCHAR | 20 |  |  |  | Password of user |
| 4 | LastName | NVARCHAR | 10 |  |  |  | Last name of user | of requirement |  |  |  |  |
| 5 | FirstName | NVARCHAR | 20 |  |  |  | First name of user |  |  |  |  |  |
| 6 | PhoneNumber | NVARCHAR | 20 | X |  |  | Phone number of user |  |  |  |  |  |
| 7 | Organization | NVARCHAR | 20 | X |  |  | Organization of user |  |  |  |  |  |
| 8 | Email | NVARCHAR | 20 | X |  |  | Email of user |  |  |  |  |  |
| 9 | DelFlag | BOOLEAN |  |  |  |  | Deleted flag of user |
| 10 | Status | BOOLEAN |  |  |  |  | Status of user (active/inactive) |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Assigment table** | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | **Null** | **Unique** | **P/F Key** | **Description** |
| 1 | AssignmentID | CHAR | 10 |  |  | PK | PK – ID of assignment |
| 2 | UserID | CHAR | 10 |  |  | FK | FK UserID of assignment |
| 3 | StartDate | DATETIME |  |  |  |  | Start date of assignment |
| 4 | EndDate | DATETIME |  | X |  |  | End date of assignment | of requirement |  |  |  |  |
| 5 | Role | NVARCHAR | 20 |  |  |  | Role of user in this assignment |  |  |  |  |  |
| 6 | DelFlag | BOOLEAN |  |  |  |  | Deleted flag of assigment |
| 7 | ProjectID | CHAR | 10 |  |  | FK | FK ProjectID of assignment |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Cost table** | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | **Null** | **Unique** | **P/F Key** | **Description** |
| 1 | CostID | CHAR | 10 |  |  | PK | PK – ID of cost |
| 2 | Budget | INTEGER |  |  |  |  | Budget of project |
| 3 | Buffer | INTEGER |  |  |  |  | Buffer of project |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Task table** | | | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | | **Null** | **Unique** | **P/F Key** | **Description** | |
| 1 | TaskID | CHAR | 10 | |  |  | PK | ID of task | |
| 2 | TaskName | NVARCHAR | 100 | |  |  |  | Name of task | |
| 3 | Status | INTEGER |  | |  |  |  | Status of task | |
| 4 | AssignmentID | CHAR | 10 | | X |  | FK | ID of assignment table (to get users belong to the project) | | of requirement | |  | |  | |  |  |
| 5 | StageID | CHAR | 10 | | X |  | FK | ID of stage that task belong to | |  | |  | |  | |  |  |
| 6 | Product | INTEGER |  | | X |  |  | Product of task (LOC, page…) | |  | |  | |  | |  |  |
| 7 | ProductSize | INTEGER |  | | X |  |  | Size of product | |  | |  | |  | |  |  |
| 8 | CompletenessStatus | INTEGER |  | | X |  |  | Number of completed products | |  | |  | |  | |  |  |
| 9 | StartDate | DATETIME |  | | X |  |  | Start date of task | |  | |  | |  | |  |  |
| 10 | PlannedEndDate | DATETIME |  | | X |  |  | Planned end date of task | |  | |  | |  | |  |  |
| 11 | EndDate | DATETIME |  | | X |  |  | Actual end date of task | |  | |  | |  | |  |  |
| 12 | PlannedEffort | INTEGER |  | | X |  |  | Planned effort of task | |  | |  | |  | |  |  |
| 13 | ActualEffort | INTEGER |  | | X |  |  | Actual effort of task | |  | |  | |  | |  |  |
| 14 | Description | NVARCHAR | 200 | | X |  |  | Description of task | |  | |  | |  | |  |  |
| 15 | ProjectID | CHAR | 10 | |  |  | FK | ID of project that task belong to | |  | |  | |  | |  |  |
| 16 | Active | BOOLEAN |  | |  |  |  | Deleted state of task | |  | |  | |  | |  |  |
|  | | | |  | | | | |  | |  | |  | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Requirements table** | | | | | | | |
| **No** | **Field name** | **Type** | **Max Length** | **Null** | **Unique** | **P/F Key** | **Description** |
| 1 | RequirementID | String | 10 |  |  | P | PK – ID of requirement |
| 2 | Description | String | 200 |  |  |  | Short description of requirement |
| 3 | Type | String | (new, change request) |  |  |  | Type of requirement |
| 4 | Size | int | (1,2,3,4,5) |  |  |  | Size of requirement |
| 5 | Release\_Notes | String | 200 | X |  |  | Notes of requirement |
| 6 | Document | String | 200 |  |  |  | Doc link of requirement |
| 7 | Effort | Int | N/A | X |  |  | Effort of requirement |
| 8 | Created\_Date | Date | N/A | X |  |  | Created date of requirement |
| 9 | Designed\_Date | Date | N/A | X |  |  | Designed date of requirement |
| 10 | Coded\_Date | Date | N/A | X |  |  | Coded date of requirement |
| 11 | Tested\_Date | Date | N/A | X |  |  | Tested date of requirement |
| 12 | Deployed\_Date | Date | N/A | X |  |  | Deployed date of requirement |
| 13 | Accepted\_Date | Date | N/A | X |  |  | Accepted date of requirement |
| 14 | Cancelled\_Date | Date | N/A | X |  |  | Cancelled date of requirement |
| 15 | ProjectID | String | 10 |  |  | F | FK ProjectID of requirement |
| 16 | Status | String | (open, designed, coded, tested, accepted, deployed, released, cancelled) |  |  |  | Status of requirement |
| 17 | Active | Boolean |  |  |  |  | Active or inactive |

## CRC Card Model

**Basic User Scenario**

1. Project Manager create new project.
2. PM manages team; add Products, Tasks, and Requirements…
3. Project Management includes schedule, progress, cost, risk, issue…
4. PM, member use DMS, Timesheet system.
5. Control, monitor, and update Project Status.

Notes: These are not fully documented user stories. More is needed, but these should give you an idea.

|  |  |
| --- | --- |
| **Dashboard Controller Class** | |
| Responsibilities | Collaborators |
| 1. Provide overall information about projects ‘status including name, health, cost, progress, quality, efficiency, date, effort. 2. Allow user to filter projects by status, category, date, cost, progress, efficiency. 3. Allow user to sort projects by name, date, project manager. 4. Provide link to project detail information and export feature. | Project Class  Planner Class  Cost Class  DMS Class  Timesheet Class |

|  |  |
| --- | --- |
| **Planner Controller Class** | |
| Responsibilities | Collaborators |
| 1. Allow project manager to manage tasks: add new task, update, delete, assign, monitor progress, completeness rate, and date. 2. Feature filter projects by status, assigned to. 3. Allow user to sort status, date, assigned to. 4. Import feature using Microsoft Project and export report. 5. Team members can update tasks’ progress. | Project Class  User Class  Stage Class  Product Class |

|  |  |
| --- | --- |
| **Project Controller Class** | |
| Responsibilities | Collaborators |
| 1. Team Management 2. Project Module Management 3. Cost 4. Product 5. Schedule 6. Progress 7. Risk, Issue, Change Request | Stage Class  Product Class  User Class  Risk Class  Issue Class  Requirement Class  Deliverable Class |

|  |  |
| --- | --- |
| **Report Controller Class** | |
| Responsibilities | Collaborators |
| 1. Project information 2. Planner 3. DMS 4. Timesheet | Project Class  Planner Class  DMS Class  Timesheet Class |

|  |  |
| --- | --- |
| **Timesheet Controller Class** | |
| Responsibilities | Collaborators |
| 1. Allow project manager to monitor timework and effort of team member. 2. Feature filter projects by date, status, project. 3. Team members can log timesheet as their works daily or weekly. | Project Class  User Class  Product Class  Stage Class |

|  |  |
| --- | --- |
| **DMS Controller Class** | |
| Responsibilities | Collaborators |
| 1. Allow project manager to monitor defect and fix bug effort of team. 2. Feature filter projects by date, status, and project, creator, assigned to. 3. Team members can update defect status. | Project Class  User Class |

|  |  |
| --- | --- |
| **Requirement Controller Class** | |
| Responsibilities | Collaborators |
| 1. Managing feature enables user to store and manage their requirement documents. 2. Including: Add, update, remove, sort 3. User interface allows user to sort requirement by type, date, priority. 4. Display completeness rate updated by user. | Project Class  Product Class |

|  |  |
| --- | --- |
| **User Admin Controller Class** | |
| Responsibilities | Collaborators |
| 1. Manage Project 2. Manage User 3. Manage Project Team 4. Manage Project Module 5. Change User’s Information 6. Reset Password 7. Manage Project 8. Manage Project Team 9. Manage Project Module 10. Manage Cost 11. Manage Product 12. Manage Work Order 13. Manage Risk, Issue, Change Request | User Class  Project Class |

|  |  |
| --- | --- |
| **Dashboard Class** | |
| Responsibilities | Collaborators |
| Represent information of projects on system. |  |

|  |  |
| --- | --- |
| **Planner Class** | |
| Responsibilities | Collaborators |
| Representation information of project’s tasks |  |

|  |  |
| --- | --- |
| **Project Class** | |
| Responsibilities | Collaborators |
| Representation information of project |  |

|  |  |
| --- | --- |
| **Report Class** | |
| Responsibilities | Collaborators |
| Representation information of project’s reports |  |

|  |  |
| --- | --- |
| **Timesheet Class** | |
| Responsibilities | Collaborators |
| Representation information of project’s timesheet |  |

|  |  |
| --- | --- |
| **DMS Class** | |
| Responsibilities | Collaborators |
| Representation information of project’s DMS |  |

|  |  |
| --- | --- |
| **Requirement Class** | |
| Responsibilities | Collaborators |
| Representation information of project’s requirements |  |

|  |  |
| --- | --- |
| **User Class** | |
| Responsibilities | Collaborators |
| Representation information of user on system |  |

|  |  |
| --- | --- |
| **Product Class** | |
| Responsibilities | Collaborators |
| Representation information of products of project |  |

|  |  |
| --- | --- |
| **Stage Class** | |
| Responsibilities | Collaborators |
| Representation information of stages of project |  |

|  |  |
| --- | --- |
| **Risk Class** | |
| Responsibilities | Collaborators |
| Representation information of risks of project |  |

|  |  |
| --- | --- |
| **Issue Class** | |
| Responsibilities | Collaborators |
| Representation information of issues of project | Project Class |

|  |  |
| --- | --- |
| **Deliverable Class** | |
| Responsibilities | Collaborators |
| Representation state of products of project |  |

## Application Security

### 6.1 User Permission

|  |  |
| --- | --- |
| **Name** | **Permission** |
| System Admin | New, Read , Edit Users, Project |
| User | Read, Edit |
| Project Full Control | New, Read, Edit,Delete |
| No Acess | N/A |

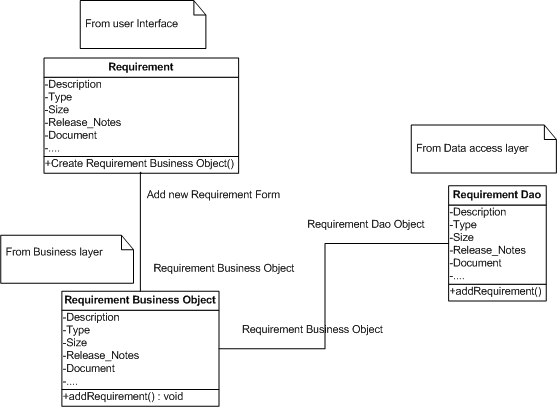
### 6.2 Security Group

|  |  |
| --- | --- |
| Name | **Description** |
| [Setup.Admin] | Administrator is responsible to manage system, users, projects. |
| [User] | User can be member or PM of a project. |
| [Project.Manager] | PMs are allowed to have full access to their projects. |

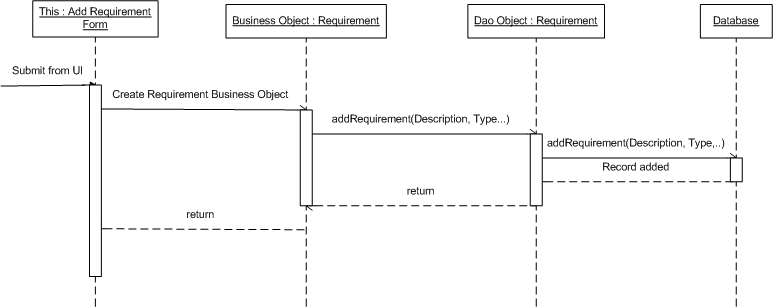
## Details function design

## Requirement \_UC01 - Add Requirement Use Case

### Class Diagram

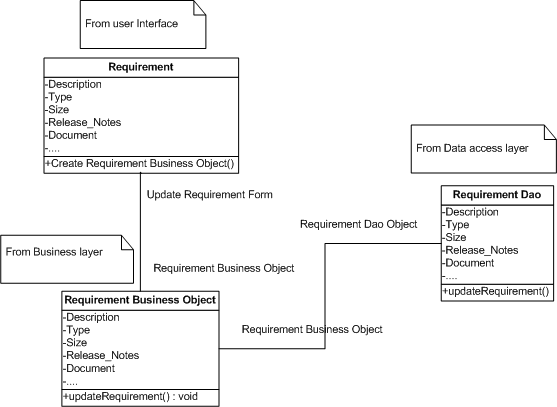


### Sequence flow

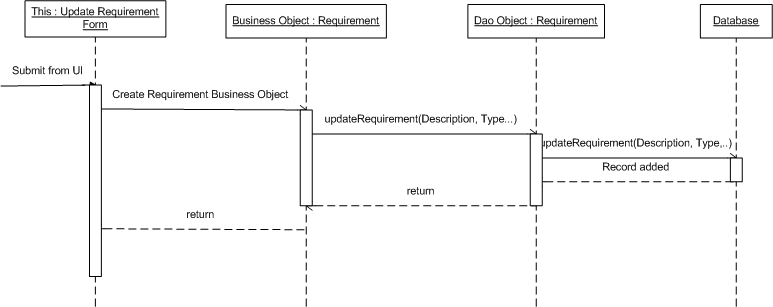


## Requirement \_UC02 - Update Requirement Use Case

### Class Diagram

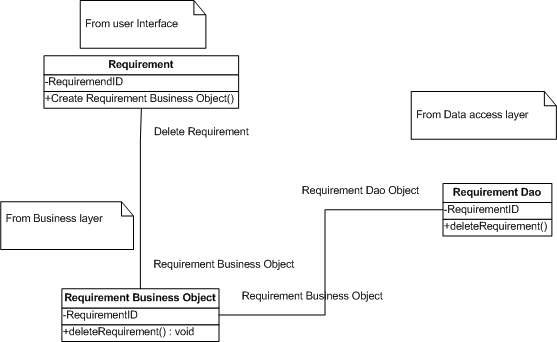


### Sequence flow

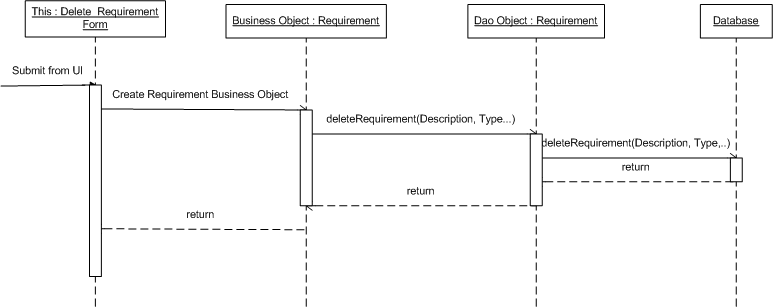


## Requirement \_UC03 - Delete Requirement Use Case

### Class Diagram

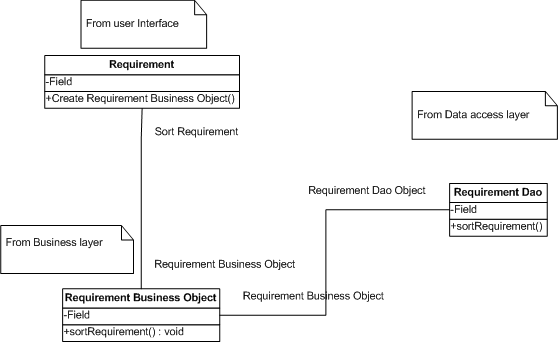


### Sequence flow



## Requirement \_UC04 - Sort Requirements Use Case

### Class Diagram



### Sequence flow



## Admin \_UC01 - Admin Create new Project Use Case

### Class Diagram



### Sequence flow



## Admin \_UC02 - Admin Search Project Use Case

### Class Diagram



### Sequence flow



## Admin \_UC03 - Admin Edit Project Use Case

### Class Diagram



### Sequence flow



## Admin \_UC04 - Admin Mange Module Use Case

### Class Diagram



### Sequence flow



## Admin \_UC05 - Admin Team Management Use Case

### Class Diagram



### Sequence flow



## Admin \_UC06 - Admin Search User Use Case

### Class Diagram



### Sequence flow



## Admin \_UC07 - Admin Create User Use Case

### Class Diagram



### Sequence flow



## Admin \_UC08 - Admin Edit User Use Case

### Class Diagram



### Sequence flow



## Admin \_UC09 - Admin Change User’s Status Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC01 - Create new Project Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC02 - Edit Project Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC03 - Manage Module Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC04 - Team Management Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC05 - Add Risk Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC06 - Edit Risk Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC07 - Delete Risk Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC08 - Add Issue Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC09 - Edit Issue Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC10 - Delete Issue Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC11 - Add Change Request Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC12 - Edit Change Request Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC13 - Delete Change Request Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC14 - Add Product Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC15 - Edit Product Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC16 - Delete Product Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC17 - Add Stage Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC18 - Edit Stage Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC19 - Delete Stage Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC20 - Add Deliverable Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC21 - Edit Deliverable Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC22 - Delete Deliverable Use Case

### Class Diagram



### Sequence flow



## ProjectEye \_UC23 - View Info Use Case

### Class Diagram



### Sequence flow



## Planner\_UC01- Filter Task Use Case

### Class Diagram



### Sequence Flow



## Planner\_UC02 - Add Task Use Case

### Class Diagram



### Sequence Flow



## Planner\_UC03 - Update Task Use Case

### Class Diagram



### Sequence Flow



## Planner\_UC04 - Delete Task Use Case

### Class Diagram



### Sequence Flow



## Planner\_UC05 - Import Task Use Case

### Class Diagram



### Sequence Flow



## Dashboard \_UC01 - Filter Project Use Case

### Class Diagram



### Sequence Flow



## Dashboard \_UC02 - Export Dashboard Use Case

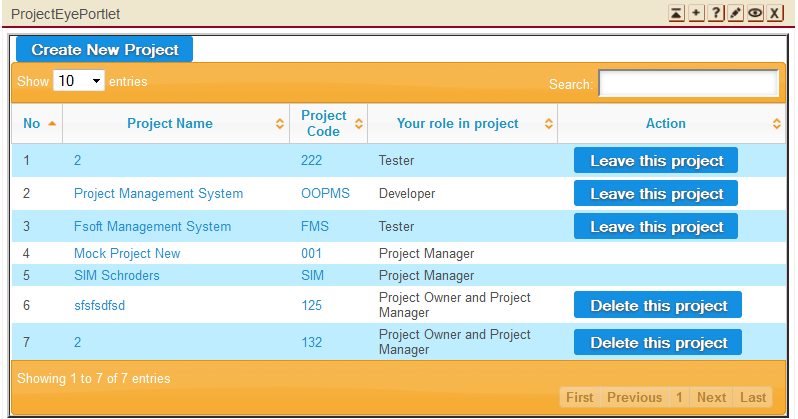
### Class Diagram

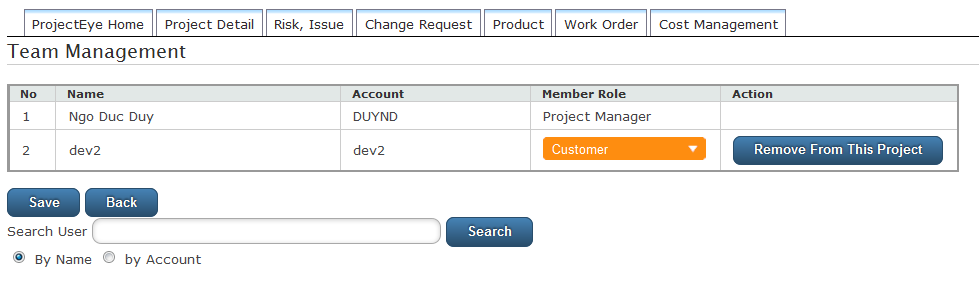


### Sequence Flow



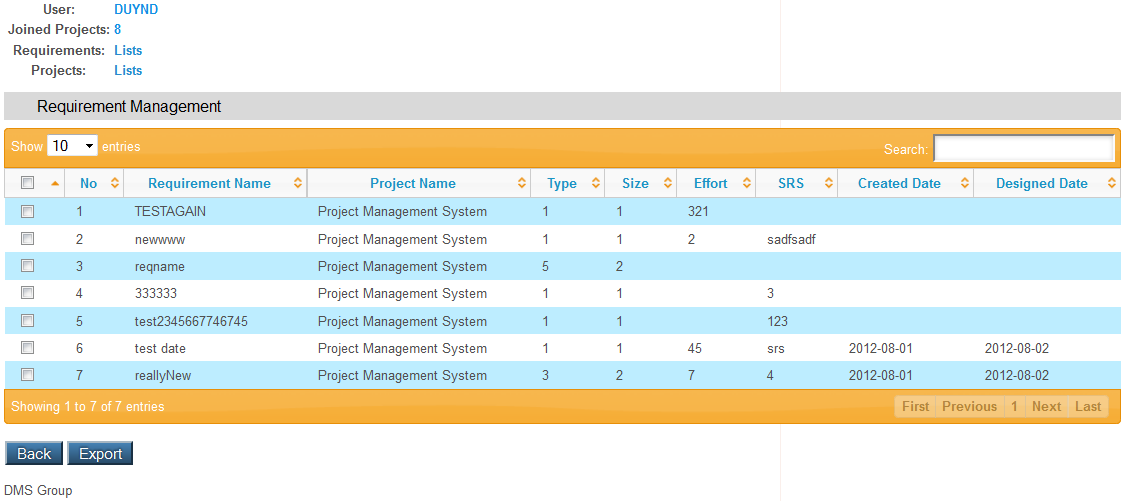
## Interface Design

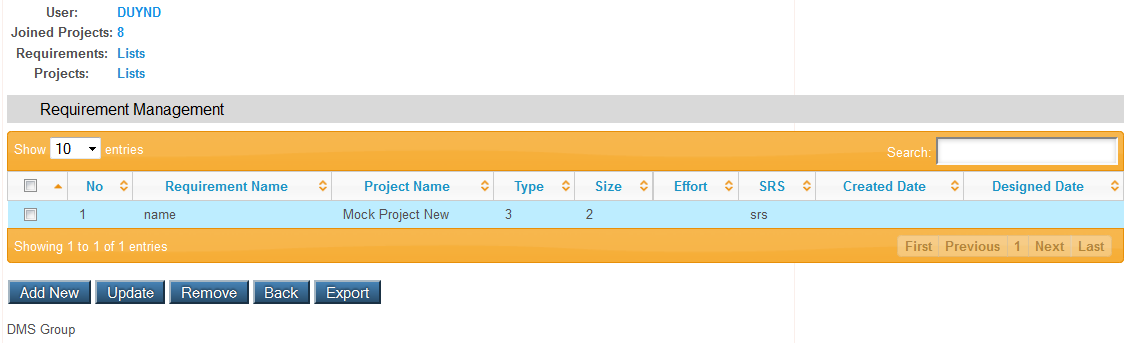
8.1 Project Eye Portlet  
  


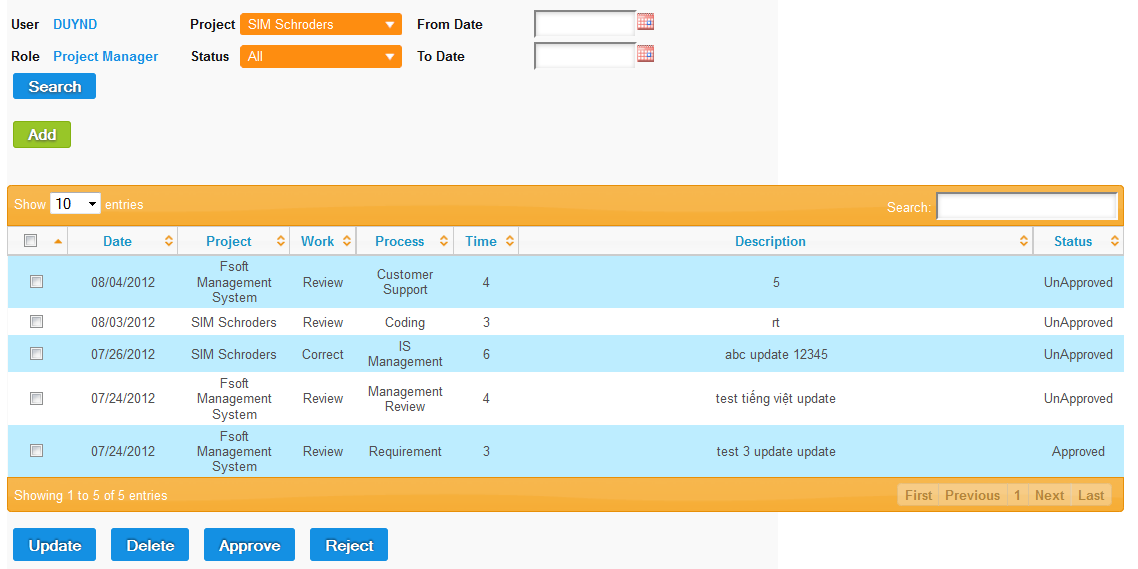
8.2 Team Management  


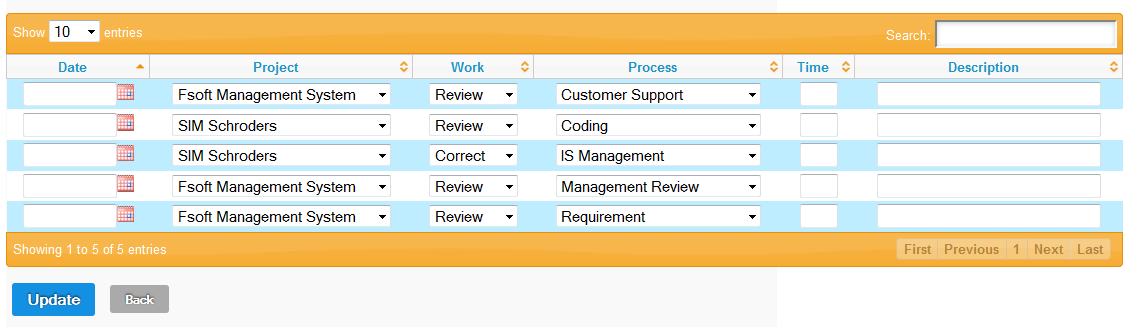
### 8.3 Planner Portlet

### 8.4 Requirement Portlet





8.5 Timesheet Portlet  
  




## Configuration

Refer to Deployment Guide.

## Packaging and Deployment

Refer to Deployment Guide.

# Software Test Documentation

## System Overview

In this accelerating of technologies and economics, Project Management is growing extremely importance to organizations because it deals effectively with the management of change. Modern economics and business environment are complicated than ever, organizations are beginning to realize that the traditional forms of management cannot adapt efficiently to the dynamics and its resulting chaos.

In the future, software industries could very well depend on how quickly the procedures and systems of project management are adopted. Therefore, Project Manager will play a very important role within organizations. Our mission is to provide an online system that actively assists those managers in managing their project. The approaches and techniques that are used in the project management process are of interest to all those who wish to be more certain about achieving predetermined targets.

## Test Approach

The testing for this project will consist of Unit, System/Integration (combined) and Acceptance test levels. It is hope that there will be at least one full time independent test person for system/integration testing. The test manager with the development team’s participation will do most testing activity.

Unit Testing worked out by the developers and approved by the implementing team leader. Developers will manage all unit test information (sample, output, data printouts, and defect information…) and it can show when team required.

The test manager will perform System/Integration Testing and implementing team leader with assistance from the individual developers as required.

The test manager and development team leader will perform acceptance Testing. The acceptance test takes place for a period of 2 weeks after completion of the System/Integration test process. Programs will enter into Acceptance test after all critical and major defects corrected. A program may have one major defect as long as it does not obstruct testing of the program (There is a work around for the error). Prior to completion of acceptance testing all open critical and major defects must be corrected.

## Features to be tested

### Dashboard

Test all functions in Dashboard page includes

* List of projects.
* Display projects’ health.
* Display projects’ basic information.
* Export report

### Planner

Test all functions in Planner page includes

* List of tasks.
* Display assignment.
* Add and assign task.
* Change task information.
* Feature sort, update, delete
* Export report

### Report

Test all functions in Report page includes

* List type of report to choose.
* Link to export chose kind of report.

### Project Eye

Test all functions in Project Eye page includes

* List of projects.
* Display projects’ information.
* Add new project.
* Edit projects’ information.
* Feature sort, update, delete
* Team management (add, update, delete user).
* Product management (add, update, delete).

Work Order management (insert, update delete stage, deliverable).

Change Request (insert, update, delete)

* Risk, Issue (insert, update, delete)

### Timesheet

Test all functions in Timesheet page includes

* List of timesheet.
* Display timesheet.
* Add timesheet.
* Feature sort, update, delete
* Export report

### DMS

Test all functions in DMS page includes

* List of defects.
* Add, assign defect.
* Feature sort, update, delete
* Export report

### Requirement

Test all functions in Requirement page includes

* List of requirements.
* Add requirement.
* Feature sort, update, delete
* Export report.

### Admin

Test all functions in Admin page includes

* List of users.
* Reset password for user.
* New users.
* Feature update, delete user
* Create, Update, delete project

### Android

* View list or project with status.
* View detail status of a project.

## Features not to be tested

The following is a list of the areas that will NOT be tested in this system:



### Text formatting

There was not enough time and this feature is not so important, so we pass this case and assume that all the text is format correctly.

### Image formatting

There was not enough time and this feature is not so important, so we pass this case and assume that the entire image is not need to format correctly.

### Performance

There was not enough time so we cannot optimize source code and improve performance. We can only ensure the system work well but not at its best performance.

### Network Connection & Security

With each user, it will be different. We only ensure that this system will affect to network.

### Performance

There was not enough time so we cannot optimize source code and improve performance. We can only ensure the system work well but not at its best performance.

## Test Case

### Project Eye

#### 5.1.1 Check view project list

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check view project list | N/A | 1. Log in | 1. Able to view list of projects. |

#### 5.1.2 Create project

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Create Project | 1. Log In | 1. Log in 2. Click “Create Project” | 1. Place to input information. 2. Validation 3. Submit Ok |

#### 5.1.3 Team management

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Team Management | 1. Log In 2. Project Manager Role | 1. Log in   2a. Go to project which User is PM.  2b. or Create new project. | 1. Able to search, add, remove team member. |

#### 5.1.4 Create Product

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Create product | 1. Log In as PM of a project | 1. Go to project that user is PM. 2. Create product link | 1. Place to input information. 2. Validation 3. Submit Ok |

#### Delete Product

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Delete product | 1. Log In as PM of a project | 1 Go to project that user is PM.  2 Choose product to Delete | 1. Product deleted |

#### 5.1.6 Update Product

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Update product | 1. Log In as PM of a project | 1. Go to project that user is PM.   Choose product to update | 1. View current information and input new ones. 2. Validation 3. Submit Ok |

#### 5.1.7 Create Risk

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Create risk | 1. Log In as PM of a project | 1 Go to project that user is PM.  2 Create risk link | 1. Place to input information. 2. Validation 3. Submit Ok |

#### 5.1.8 Delete Risk

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Delete risk | Log In as PM of a project | 1. Go to project that user is PM. 2. Choose risk to Delete | 1. risk deleted |

#### 5.1.9 Update Risk

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Update risk | Log In as PM of a project | 1. Go to project that user is PM. 2. Choose risk to update | 1. View current information and input new ones. 2. Validation 3. Submit Ok |

#### 5.1.10 Create Issue

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Create issue | Log In as PM of a project | 1. Go to project that user is PM. 2. Create issue link | 1. Place to input information. 2. Validation 3. Submit Ok |

#### 5.1.11 Delete Issue

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Delete issue | Log In as PM of a project | 1. Go to project that user is PM. 2. Choose issue to Delete | 1 issue deleted |

#### 5.1.12 Update Issue

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Update issue | Log In as PM of a project | 1. Go to project that user is PM. 2. Choose issue to update | 1. View current information and input new ones. 2. Validation 3. Submit Ok |

#### 5.1.13 Create Change

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Create change | Log In as PM of a project | 1. Go to project that user is PM. 2. Create change link | 1. Place to input information. 2. Validation 3. Submit Ok |

#### 5.1.14 Delete Change

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Delete change | Log In as PM of a project | 1. Go to project that user is PM. 2. Choose change to Delete | 1. change deleted |

#### 5.1.15 Update Change

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Update change | Log In as PM of a project | 1. Go to project that user is PM. 2. Choose change to update | 1. View current information and input new ones. 2. Validation 3. Submit Ok |

### Planner

#### 5.2.1 Check view project list

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check view project list | Member of one project | 1. Log in | 1. Able to view list of projects that logged-in user is member. |

#### 5.2.2 Check view tasks list of a project

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check view task list | 1. Logged in | 1. Click on one project link | 1. Able to view list of task of that project |

#### 5.2.3 Check PM can Add/Delete/Assign Task

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check PM can ADA task | 1 Logged in  2 User is PM of at least one project | 1. Log in 2. Click on project link that user is PM | 1. Able to view Add, assign, delete buttons. |

#### 5.2.4 Check Add Task

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check add task | 1 Logged in  2 User is PM of at least one project | 1. Log in 2. Click on project link that user is PM 3. Click add button | 1. Place to input information. 2. Validation 3. Submit Ok |

#### 5.2.5 Check Delete Task

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check delete task | 1. Logged in 2. User is PM of at least one project | 1. Log in 2. Click on project link that user is PM 3. Tick one task and Click delete button | Task deleted |

#### 5.2.6 Check Assign Task

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check assign task | 1. Logged in 2. User is PM of at least one project | 1. Log in 2. Click on project link that user is PM 3. Tick at least one task and Click assign button | 1 task is assigned |

#### 5.2.7 Check team member can update Task

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check member can update task | 1. Logged in 2. User is member of at least one project | 1. Log in 2. Click on project link that user is member 3. Tick at least one task and Click update button | 1. Search action on change of the box. |

#### 5.2.8 Check Update Task

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check update task | 1. Logged in 2. User is member of at least one project | 1. Log in 2. Click on project link that user is member 3. Tick at least one task and Click update button | 1. See current values, Place to input information. 2. Validation 3. Submit Ok |

#### Check Search Task

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check search task | 1. Logged in | 1. Log in 2. View list task and type in to search box | 1. Search action on change of the box. |

#### 5.2.10 Check Sort Task

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check sort task | 1. Logged in | 1. Log in  2. View list task and click on any column header | 1 tasks are sorted by clicked column. |

### Requirement

#### 5.3.1 Check view project list

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check view project list | Member of one project | 1. Log in | 1. Able to view list of projects that logged-in user is member. |

#### 5.3.2 Check view requirements list of a project

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check view requirement list | 1. Logged in | 1. Click on one project link | 1. Able to view list of requirements of that project |

#### 5.3.3 Check PM can Add/Update/Delete Requirement

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check PM can AUD requirement | 1 Logged in  2 User is PM of at least one project | 1. Log in 2. Click on project link that user is PM | 1. Able to view Add, Update, Remove buttons. |

#### 5.3.4 Check Add requirement

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check add requirement | 1 Logged in  2 User is PM of at least one project | 1. Log in 2. Click on project link that user is PM 3. Click add button | 1. Place to input information. 2. Validation 3. Submit Ok |

#### 5.3.5 Check Update requirement

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check update requirement | 1. Logged in 2. User is PM of at least one project | 1. Log in 2. Click on project link that user is PM 3. Tick one requirement and Click update button | 1. See current value and able to input new information. 2. Validation 3. Submit Ok |

#### 5.3.6 Check Remove requirement

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check remove requirement | 1. Logged in 2. User is PM of at least one project | 1. Log in 2. Click on project link that user is PM 3. Tick at least one requirement and Click remove button | 1 Requirements are removed and the list refresh. |

#### 5.3.7 Check Search requirement

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check search requirement | 1. Logged in | 1. Log in 2. View list requirements and type in to search box | 1. Search action on change of the box. |

#### 5.3.8 Check Sort requirement

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check sort requirement | 1. Logged in | 1. Log in  2. View list requirement and click on any column header | 1 Requirements are sorted by clicked column. |

### 5.4 Timesheet

#### 5.4.1 Check PM view all timesheet list

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check PM view timesheet list | N/A | Log in as PM | 1. Able to view all list timesheet. |

#### 5.4.2 Check team members view their own timesheet

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check team member view timesheet list | N/A | Log in as team member | 1. Able to view list timesheet of their own. |

#### 5.4.3 Check team members can Add/Update/Delete their own timesheet

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check team member can AUD timesheet | N/A | Log in as team member | Able to view Add, Update, Delete buttons of their own timesheet. |

#### 5.4.4 Check PM can Delete all timesheet

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check PM delete timesheet | N/A | Log in as PM | Able to view Delete buttons of all timesheet. |

#### 5.4.5 Check PM can Approve/Reject all timesheet

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check PM approve/reject timesheet | N/A | Log in as PM | Able to view Approve/Reject buttons of all timesheet. |

#### 5.4.6 Check Add timesheet

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check add timesheet | User is team member of at least one project | 1. Log in 2. Click add button | 1. Place to input information. 2. Validation 3. Submit Ok |

#### 5.4.7 Check update timesheet

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check update timesheet | User is team member of at least one project | 1. Log in 2. Tick at least one timesheet and Click update button | 1. View current value and Place to input new information.  2.Validation  3.Submit Ok |

#### 5.4.8 Check Delete timesheet

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check delete timesheet | User is team member or PM | 1. Log in  2. Tick at least one timesheet and Click delete button | 1 PM is able to delete all timesheet of that project / team member can delete their own timesheet. |

#### 5.4.9 Check approve timesheet

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check approve timesheet | User is PM of one project | 1. Log in 2. Tick timesheet and click Approve button. | timesheet approved and status changed. |

#### 5.4.10 Check reject timesheet

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check reject timesheet | User is PM of one project | 1. Log in 2. Tick timesheet and click reject button. | timesheet rejected and status changed. |

#### 5.4.11 Check Search timesheet

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check search timesheet | Logged in | 1. Log in 2. View list timesheet and type in to search box | Search action on change of the box. |

#### 5.4.12 Check Sort timesheet

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check sort timesheet | Logged in | 1. Log in  2. View list timesheet and click on any column header | 1 timesheet are sorted by clicked column. |

### 5.5 DMS

#### 5.5.1 Check view Defect list

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check view Defect list | Team member of a project | 1. Log in 2. Click on project link | 1. Able to view list of Defect. |

#### 5.5.2 Check create Defect

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check create Defect | Team member of a project | 1. Log in 2. Click on project link | 1. Able to create |

#### 5.5.3 Check assign Defect

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check assign Defect | Team member of a project | 1. Log in 2. Click on project link | 1. Able to assign |

#### 5.5.4 Check update Defect

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check update Defect | Team member of a project | 1. Log in 2. Click on project link | 1. Able to update |

#### 5.5.5 Check delete Defect

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check delete Defect | Team member of a project | 1. Log in 2. Click on project link | 1. Able to delete |

#### 5.5.6 Check search Defect

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check search Defect | Team member of a project | 1. Log in 2. Click on project link | 1. Able to search |

### 5.6 Dashboard

#### 5.6.1 Check view project status list

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check view project status list |  | 1. Log in | 1. Able to view list status of projects that logged-in user is member. |

#### 5.6.2 Check view detail status of a project

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check view detail project status | 1. Logged in | 1. Click on one project link | 1. Able to view detail status of that project |

### Admin

#### 5.7.1 Create User

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Create User | System Admin | 1. Log in as System admin. 2. Go to Create User | 1. Input information. 2. Submit ok |

#### 5.7.1 Update User

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Update User | System Admin | 1. Log in as System admin. 2. Go to Update User | 1. Input information. 2. Submit ok |

#### 5.7.1 Delete User

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Delete User | System Admin | 1. Log in as System admin.  2. Go to Delete User | 1. Submit ok |

### 5.8 Report

#### 5.8.2 Check Report Project Status

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check report project status | N/A | 1 Go to report.  2 Choose report project status | 1 Get report printed out |

#### 5.8.2 Check Report Task Status

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check report task status | N/A | 1 Go to report.  2 Choose report task status | 1 Get report printed out |

#### 5.8.3 Check Report Timesheet Status

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check report timesheet status | N/A | 1 Go to report.  2 Choose report timesheet status | 1 Get report printed out |

#### 5.8.4 Check Report Requirement Status

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check report requirement status | N/A | 1 Go to report.  2 Choose report requirement status | 1 Get report printed out |

#### 5.8.5 Check Report DMS Status

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check report DMS status | N/A | 1 Go to report.  2 Choose report DMS status | 1 Get report printed out |

### 5.9 Android

#### 5.9.1 Check view project status list

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check view project status list |  | 1. Log in | 1. Able to view list status of projects that logged-in user is member. |

#### 5.9.2 Check view detail status of a project

|  |  |  |  |
| --- | --- | --- | --- |
| **Content** | **Precondition** | **Test case procedure** | **Expected output** |
| Check view detail project status | 1. Logged in | 1. Click on one project link | 1. Able to view detail status of that project |

## Checklists



### Checklist of Validation

|  |  |  |  |
| --- | --- | --- | --- |
| Question | Yes | No | N/A |
| 1. Does a failure of validation on every field cause a sensible user error message? | x |  |  |
| 2. Is the user required to fix entries which have failed validation tests? | X |  |  |
| 3. Have any fields got multiple validation rules and if so are all rules being applied? | X |  |  |
| 4. If the user enters an invalid value and clicks on the SAVE button (i.e. does not TAB off the field) is the invalid entry identified and highlighted correctly with an error message? | X |  |  |
| 5. Is validation consistently applied at screen level unless specifically required at field level? | X |  |  |
| 6. For all numeric fields check whether negative numbers can and should be able to be entered. | X |  |  |
| 7. For all numeric fields check the minimum and maximum values and also some mid-range values allowable? | X |  |  |
| 8. For all character/alphanumeric fields check the field to ensure that there is a character limit specified and that this limit is exactly correct for the specified database size? | X |  |  |
| 9. Do all mandatory fields require user input? | X |  |  |
| 10. If any of the database columns don't allow null values then the corresponding screen fields must be mandatory. (If any field which initially was mandatory has become optional then check whether null values are allowed in this field.) | X |  |  |

### Submission Checklist

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Question | | Yes | No | N/A |
| TABLES AND FIGURES | | | | |
| 1 | Does every table column, including the stub column, have a heading? | X |  |  |
| 2 | Are all tables referred to in text? | X |  |  |
| 3 | Are the elements in the figures large enough to remain legible after the figure has been reduced to the width of a journal column or page? | X |  |  |
| 4 | Is lettering in a figure no smaller than 8 points and no larger than 14 points? | X |  |  |
| 5 | Are the figures being submitted in a file format acceptable to the publisher? | X |  |  |
| 6 | Has the figure been prepared at a resolution sufficient to produce a high-quality image? | X |  |  |
| 7 | Are all figures numbered consecutively with Arabic numerals? | X |  |  |
| 8 | Are all figures and tables mentioned in the text and numbered in the order in which they are mentioned? | X |  |  |
| REFERENCES | | | | |
| 1 | Are references cited both in text and in the reference list? |  |  | x |
| 2 | Do the text citations and reference list entries agree both in spelling and in date? |  |  | x |
| 3 | Are journal titles in the reference list spelled out fully? |  | x |  |
| 4 | Are the references (both in the parenthetical text citations and in the reference list) ordered alphabetically by the authors’ surnames? |  | x |  |
| 5 | Are inclusive page numbers for all articles or chapters in books provided in the reference list? |  | x |  |
| 6 | Are references to studies included in your meta-analysis preceded by an asterisk? |  | x |  |
| FORMAT | | | | |
| 1 | Have you checked the journal’s website for instructions to authors regarding specific formatting requirements for submission? |  | x |  |
| 2 | Is the entire manuscript—including quotations, references, author note, content footnotes, and figure captions—double-spaced? Is the manuscript neatly prepared? |  | x |  |
| 3 | Are the margins at least 1 in. (2.54 cm)? |  |  | x |
| 4 | Are the title pages, abstracts, references, appendices, content footnotes, tables, and figures on separate pages (with only one table or figure per page)? Are the figure captions on the same page as the figures? Are manuscript elements ordered in sequence, with the text pages between the abstract and the references? |  |  | x |
| 5 | Are all pages numbered in sequence, starting with the title page? | x |  |  |
| PARAGRAPHS AND HEADINGS | | | | |
| 1 | Is each paragraph longer than a single sentence but not longer than one manuscript page? |  |  | x |
| 2 | Do the levels of headings accurately reflect the organization of the paper? | x |  |  |
| 3 | Do all headings of the same level appear in the same format? | x |  |  |

## 7. Defect Log

Refer to OOPMS\_Defects.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Type | Status | Priority | Owner | Summary |
| 1 | Defect | Fixed | Low | truongmh60003 | [TimeSheetPortlet] Import is never used |
| 2 | Defect | Accepted | Critical | truongmh60003 | [TimeSheetPortlet] Missing important features |
| 3 | Defect | New | High | thachln | [TimeSheetPortlet] Not re-use the logon user of the portal |
| 4 | Defect | Fixed | Low | tocongthanhhai | [ProjectEyePortlet] Java build path of project is configured not well |
| 5 | Defect | New | High | tocongthanhhai | [ProjectEyePortlet] Done re-use DAO layer in DaoCommon |
| 6 | Defect | Fixed | Low | tocongthanhhai | [ProjectEyePortlet] Output file of Log4j is configured not good |
| 7 | Defect | Fixed | Medium | tocongthanhhai | [ProjectEyePortlet] There is no use CSS of uPortal |
| 8 | Defect | Fixed | Low | tocongthanhhai | [ProjectEyePortlet] Could not make distribution by ANT tool |
| 9 | Defect | Fixed | Medium | tocongthanhhai | [ProjectEyePortlet] Unnecessary folder in SVN |
| 10 | Defect | Fixed | Medium | tocongthanhhai | [ProjectEyePortlet] Unclear error message |
| 11 | Defect | Invalid | Medium | tocongthanhhai | [ProjectEyePortlet] Encapsulate properties of Project into a object |
| 12 | Defect | Fixed | High | tocongthanhhai | [ProjectEyePortlet] No data validation and display error in Create project screen |
| 13 | Defect | New | Medium | tocongthanhhai | [ProjectEyePortlet] Error when logon user has not existed in database |
| 14 | Defect | Fixed | Low | giang.phamnguyen | [Planner] Could not make distribution |
| 15 | Defect | Fixed | Medium | giang.phamnguyen | [Planner] Java build path of project is configured not well |
| 16 | Defect | Fixed | Low | giang.phamnguyen | [Planner] Project name/Project folder is not true |
| 17 | Defect | Fixed | Medium | giang.phamnguyen | [Planner] Home page of the Planner Portlet is not good |
| 18 | Defect | Fixed | Medium | giang.phamnguyen | [Planner] Unclear error message |
| 19 | Defect | Fixed | Critical | duynd.vn | [RequirementPortlet] Add button doesnot work properly |
| 20 | Defect | Fixed | Medium | duynd.vn | [RequirementPortlet] Have not apply validation |
| 21 | Defect | Fixed | Medium | duynd.vn | [RequirementPortlet] CSS problem |
| 22 | Defect | Fixed | Low | duynd.vn | [RequirementPortlet] There are still test button, for test purpose |
| 23 | Defect | Fixed | Medium | tocongthanhhai | [ProjectEyePortlet] Have not validate create project |
| 24 | Defect | Accepted | High | duynd.vn | [RequirementPortlet] Have not use Daocommon |
| 25 | Defect | Fixed | Medium | tocongthanhhai | [ProjectEyePortlet] Unable to update project detail |
| 26 | Defect | Fixed | High | duynd.vn | [RequirementPortlet] Init screen is not right |
| 27 | Defect | Fixed | Medium | giang.phamnguyen | [Planner] Do not reuse table Tasks of FMS |
| 28 | Defect | New | Medium | giang.phamnguyen | [Planner] Do not use DaoCommon |
| 29 | Defect | Fixed | Medium | tocongthanhhai | [ProjectEyePortlet] Initial data for screen "Create project" is not good |
| 30 | Defect | Fixed | Medium | duynd.vn | [RequirementPortlet] Java build path of project is configured not well |
| 31 | Defect | Fixed | Low | duynd.vn | [RequirementPortlet] Folder 'resouce' is placed not correctly |
| 32 | Defect | Fixed | Medium | duynd.vn | [RequirementPortlet] Error occurred but no log file |
| 33 | Defect | Duplicate | Medium | duynd.vn | [ProjectEyePortlet] Homepage is not good. |
| 34 | Defect | Fixed | Low | tocongthanhhai | [ProjectEyePortlet] Validation messages are not friendly |
| 35 | Defect | New | Low | tocongthanhhai | [ProjectEyePortlet] Project detail page is not good |
| 36 | Defect | Fixed | Low | tocongthanhhai | [ProjectEyePortlet] In screen "Create Project", Cancel button is no applied |
| 37 | Defect | Fixed | Low | tocongthanhhai | [ProjectEyePortlet] In screen "Create Project", could not type the Planned End Date completely |
| 38 | Defect | Fixed | Low | tocongthanhhai | [ProjectEyePortlet] In screen "Create Project", no process escapse HTML character |
| 39 | Defect | New | Medium | tocongthanhhai | [ProjectEyePortlet] Existing projects are not displayed in the Home screen |
| 40 | Defect | Fixed | Medium | duynd.vn | [RequirementPortlet] CSS does not fit when deploy to uPortal |
| 41 | Defect | Fixed | Low | truongmh60003 | [TimesheetPortlet] Not proper default date when add timesheet |
| 42 | Defect | Fixed | Medium | truongmh60003 | [TimesheetPortlet] Unable to update Date, Project |
| 43 | Defect | Fixed | High | truongmh60003 | [TimesheetPortlet] Add timesheet problem - Unable to add timesheet for old date (yesterday), able to add timesheet for future (tomorrow) |
| 44 | Defect | Fixed | Low | truongmh60003 | [TimesheetPortlet] Able to update approved timesheet |
| 45 | Defect | Fixed | Low | truongmh60003 | [TimesheetPortlet] Unable to view all timesheet of all project from search feature |
| 46 | Defect | Fixed | Low | truongmh60003 | [TimeSheetPortlet] Java build path of project is configured not well |
| 47 | Review | New | Low | truongmh60003 | [TimeSheetPortlet] Folder 'resouce' is placed not correctly |
| 48 | Defect | Fixed | Medium | duynd.vn | [RequirementPortlet] Use menu bar |
| 49 | Defect | Fixed | Medium | duynd.vn | [RequirementPortlet] Layout of screen "Add requirement" is very bad |
| 50 | Review | Fixed | Medium | giang.phamnguyen | [Planner] Could not click on links of projects |
| 51 | Defect | Fixed | Medium | duynd.vn | [RequirementPortlet] User can see non-unauthorized projects |
| 52 | Defect | Fixed | Medium | duynd.vn | [RequirementPortlet] Could not go back the first screen |
| 53 | Defect | Fixed | Medium | tocongthanhhai | [ProjectEyePortlet] Could not compile source code |
| 54 | Defect | Fixed | Medium | duynd.vn | [RequirementPortlet] List of requirements of project is incorrect |
| 55 | Defect | New | Medium | thachln | [ProjectEyePortlet] Click on the item with HAND CURSOR but no action |
| 56 | Defect | Fixed | Medium | tocongthanhhai | [ProjectEyePortlet] There is no confirmation screen before delete |
| 57 | Defect | Fixed | Medium | tocongthanhhai | [ProjectEyePortlet] Size of Project Code is not matched with size of field in database |
| 58 | Defect | New | Medium | truongmh60003 | [TimeSheetPortlet] The error occurred but no logging file |
| 59 | Defect | New | Medium | tocongthanhhai | [ProjectEyePortlet] Could not perform "Create New Project" in uPortal |
| 60 | Defect | New | Medium | tocongthanhhai | [ProjectEyePortlet] Layout of screen Create Project is not good |
| 61 | Defect | New | Medium | tocongthanhhai | [ProjectEyePortlet] Developer of the project but could not view the project |
| 62 | Defect | New | Medium | giang.phamnguyen | [Planner] Properties in /PlannerPortlet/resource/messages.properties are used or not? |
| 63 | Defect | New | Medium | giang.phamnguyen | [Planner] Non-unauthenticated projects are displayed for user |
| 64 | Defect | New | Low | giang.phamnguyen | [Planner] Naming Controller is not good |
| 65 | Defect | New | Low | truongmh60003 | [RequirementPortlet] There are two resource messages |
| 66 | Defect | Fixed | Low | duynd.vn | [RequirementPortlet] Error in JSP |
| 67 | Defect | New | Medium | truongmh60003 | [TimeSheetPortlet] Could not compile source code |
| 68 | Defect | Fixed | Medium | tocongthanhhai | [ProjectEyePortlet] Paging in workOrder, Product |
| 69 | Defect | Started | Medium | duynd.vn | [RequirementPortlet] Portlet is displayed badly on uPortal |
| 70 | Defect | Accepted | Medium | duynd.vn | [RequirementPortlet] Only PM of a project can view Add button but when click add, he/she is able to add req to other project |
| 71 | Defect | New | Low | truongmh60003 | [TimeSheetPortlet] Duplicated libraries |
| 72 | Defect | New | High | truongmh60003 | [TimeSheetPortlet] Error in the first screen. |
| 73 | Defect | New | High | truongmh60003 | [TimeSheetPortlet] Duplicate models with entities DaoCommon |
| 74 | Defect | New | Medium | giang.phamnguyen | [PlannerPortlet] Dialog box with unfriendly message |

## Test Log

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Module code | Pass | | | Fail | | Untested | | N/A | | Number of test cases | |
| Total | |  |  | |  | |  | | 76 | |
| Project Eye | |  |  | |  | |  | | 28 | |
| Planner | | 28 |  | |  | |  | | 10 | |
| Requirement | | 8 |  | |  | |  | | 8 | |
| Timesheet | | 12 |  | |  | |  | | 12 | |
| DMS | |  |  | |  | |  | | 6 | |
| Dashboard | |  |  | |  | |  | | 2 | |
| Admin | |  |  | |  | |  | | 3 | |
| Report | |  |  | |  | |  | | 5 | |
| Android | |  |  | |  | |  | | 2 | |

# User Manual

## Installation Guide

Refer Installation Guide Document

## Users’ Guide

Refer Users’ Guide Document