2011

Revision of ReMoTe (Recursively Estimating Multi-Threaded Observation Tool Enterprise) for commercialization

Jeongtaek Hong

Follow this and additional works at: https://scholarworks.lib.csusb.edu/etd-project

Recommended Citation
Hong, Jeongtaek, "Revision of ReMoTe (Recursively Estimating Multi-Threaded Observation Tool Enterprise) for commercialization" (2011). Theses Digitization Project. 3327.
https://scholarworks.lib.csusb.edu/etd-project/3327
REVISION OF ReMoTe (RECURSIVELY ESTIMATING MULTI-THREADED OBSERVATION TOOL ENTERPRISE) FOR COMMERCIALIZATION

A Project
Presented to the
Faculty of
California State University,
San Bernardino

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Computer Science

by
Jeongtaek Hong
June 2011
REVISION OF ReMoTe (RECURSIVELY ESTIMATING MULTI-THREADED OBSERVATION TOOL ENTERPRISE) FOR COMMERCIALIZATION

A Project
Presented to the
Faculty of
California State University,
San Bernardino

by
Jeongtaek Hong
June 2011
Approved by:

13 Jun 2011

Arturo I Concepcion,
Computer Science and Engineering

David Turner

Josephine Mendoza
ABSTRACT

ReMoTe (The Recursively Estimating Multi-threaded Observation Technology Enterprise) is a software process management tool that assists project managers in managing, controlling, tracking, monitoring, and predicting the entire software development. There are several improvements to ReMoTe. First, ReMoTe provides users with a detailed display of the progress of a software project. Second, the user can create Gantt charts, and provide storage and retrieval of artifacts submitted to ReMoTe by all users in the project. Third, the user is reminded of the deadlines by setting start day and end day of project. Fourth, ReMoTe is able to handle multiple projects effectively. Fifth, ReMoTe shows the estimated delivery time in a huge project, and computes the delivery dates and the critical path. Sixth, ReMoTe is improved by providing an easy-to-use interface. It also provides traceability of the software requirements specifications (SRS) in the tool. ReMoTe is always accessible because it is Web-based. With these improvements, it is hoped that ReMoTe is ready for commercialization.
ACKNOWLEDGEMENTS

I would like to thank Dr. Arturo I Concepcion for providing me the most valuable suggestions about direction of this project and lots of information that extended my understanding of the ReMoTe tool and improved the quality of the project, and also thank him for encouraging me to finish the project.

Subsequently, I would like to thank Dr. Josephine Mendoza for helping me to understand database architecture and to encourage me to finish the project.

Specifically, I would like to thank Dr. David A. Turner for his comments and suggestions to understand PHP.

Additionally, I would like to thank Professor Kwon Soo Han for providing me to understand setting up the ReMoTe server properly and securely.

I would like to thank CSE655 class for helping in fixing some of the bugs and crashes of ReMoTe.

In the end, I would like to extend my thanks to my parents and my wife for supporting me during my master study. Especially, I would like to thank my wife for encouraging and supporting me all the time.
TABLE OF CONTENTS

ABSTRACT ........................................ iii
ACKNOWLEDGEMENTS ................................. iv
LIST OF TABLES ...................................... xv
LIST OF FIGURES ..................................... xvii

CHAPTER ONE: INTRODUCTION

1.1 History ........................................... 1
1.2 Purpose of the Project ............................ 2
1.3 Improvements to ReMoTe ........................... 2

CHAPTER TWO: SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Scope ............................................. 5
  2.1.1 Glossary ....................................... 5
  2.1.2 Overview ....................................... 9
2.2 Overall Description ............................... 9
  2.2.1 Product Perspective ......................... 9
  2.2.2 System Interfaces .............................. 9
  2.2.3 User Interfaces ............................... 10
    2.2.3.1 System Administrator .................. 11
    2.2.3.2 CEO
       (Chief Executive Officer) .................. 11
    2.2.3.3 Project Manager ......................... 11
    2.2.3.4 Team Leader
       /Sub-Team Leader .......................... 12
    2.2.3.5 Employee ................................. 13
2.2.4 Hardware Interfaces .................. 13
2.2.5 Software Interfaces ................. 13
   2.2.5.1 Communication Interface ...... 13
   2.2.5.2 Memory and Hardware Constraints ........ 14
   2.2.5.3 Operations .................... 14
   2.2.5.4 Adaptation Requirements ...... 14
2.2.6 Product Functions ................... 14
2.2.7 User Characteristics ................. 17
2.2.8 Constraints .......................... 17
2.2.9 Assumptions and Dependencies ...... 17
2.2.10 Apportioning of Requirements ..... 17
2.3 Specific Requirements .................. 17
   2.3.1 External Interface ................. 17
   2.3.2 User Interface ..................... 18
      2.3.2.1 Login Interface .............. 18
      2.3.2.2 Administrator Interface ...... 20
         2.3.2.2.1 Create Project .......... 21
         2.3.2.2.2 Edit Projects .......... 21
         2.3.2.2.3 Assign Users .......... 22
         2.3.2.2.4 Edit Users ........... 24
         2.3.2.2.5 Edit Messages ....... 25
      2.3.2.3 CEO (Chief Executive Officer) Interface .......... 26
2.3.2.3.1 View Dependencies ... 27
2.3.2.3.2 Set Dependencies ... 29

2.3.2.4 Project Manager/Team Leader /Sub-Team Leader ........... 30
2.3.2.4.1 Project Manager .... 30
2.3.2.4.2 Define Life-Cycle Model ........... 31
2.3.2.4.3 Set Team Names .... 34
2.3.2.4.4 Select Teammates .... 35
2.3.2.4.5 Set Budget .... 35
2.3.2.4.6 Set Man-Hours .... 36

2.3.2.5 Manage My Thread Interface ........... 36
2.3.2.6 Approve TM Thread .... 37
2.3.2.7 Communication Interface .... 38
  2.3.2.7.1 Chat Room .... 38
  2.3.2.7.2 Write/View Message .... 39
  2.3.2.7.3 Set Notes/Tasks .... 41
2.3.2.8 View Project .... 42
2.3.2.9 View Progress chart .... 43

2.3.3 Performance Requirements .... 44
2.3.4 Logical Database Requirements .... 44
2.3.5 Design Constraints .... 44
2.3.6 Software System Attributes .... 44
2.3.6.1 Reliability ...................................... 45
2.3.6.2 Availability ..................................... 45
2.3.6.3 Security .......................................... 45
2.3.6.4 Portability ....................................... 45
2.3.6.5 Maintainability .................................. 45

CHAPTER THREE: ARCHITECTURAL DESIGN

3.1 MVC Pattern ........................................... 46
3.2 Model .................................................. 47

3.2.1 Admin Model ......................................... 48

3.2.1.1 AssignedUserAction .............................. 49
3.2.1.2 AssignUserPageAction ........................... 49
3.2.1.3 CreateProjectAction ............................. 49
3.2.1.4 DeleteMessageAction ............................. 49
3.2.1.5 DeleteUserAction ................................ 49
3.2.1.6 EditMessagesAction .............................. 49
3.2.1.7 EditProjectAction ............................... 49
3.2.1.8 EditUserAction ................................... 49
3.2.1.9 EditUserPageAction ............................... 49
3.2.1.10 LoginAction ..................................... 49

3.2.2 CEO Model .......................................... 50

3.2.2.1 CreateDependenciesAction .................... 50
3.2.2.2 LoginAction ...................................... 50
3.2.2.3 ViewDependenciesAction ....................... 50
3.2.2.4 ViewProjectAction .......... 50
3.2.3 Client Model ................. 51
  3.2.3.1 ApproveThreadAction ...... 53
  3.2.3.2 AddThreadAction .......... 53
  3.2.3.3 CreateThreadAction ...... 53
  3.2.3.4 CostAction ............... 53
  3.2.3.5 CostAnalysisAction ...... 53
  3.2.3.6 DeleteFileAction .......... 53
  3.2.3.7 DeleteMethodAction ...... 53
  3.2.3.8 DefineThreadAction ...... 53
  3.2.3.9 DeleteNoteAction .......... 53
  3.2.3.10 EditMethodAction ....... 52
  3.2.3.11 EditNoteAction .......... 53
  3.2.3.12 ForgotPasswordAction ... 54
  3.2.3.13 GanttChartAction ....... 54
  3.2.3.14 HoursAction ............ 54
  3.2.3.15 LoginAction ............. 54
  3.2.3.16 MessageBoardAction ..... 54
  3.2.3.17 MessageCheck .......... 54
  3.2.3.18 NoteAction ............. 54
  3.2.3.19 NoteListAction .......... 54
  3.2.3.20 OverridePhaseNameAction .. 54
3.2.3.21 RegistrationAction ............ 54
3.2.3.22 SelectProjectAction ............ 54
3.2.3.23 SelectTeamAction ................ 55
3.2.3.24 SetDatesAction .................... 55
3.2.3.25 SetNumberPhasesAction ............ 55
3.2.3.26 SetPictureAction .................. 55
3.2.3.27 SetTeamNamesAction ............... 55
3.2.3.28 SettingsAction ..................... 55
3.2.3.29 StatisticsAction ................... 55
3.2.3.30 ViewMessageAction ................ 55
3.2.3.31 ViewMessagesAction ................ 55
3.2.3.32 ViewProjectAction ................ 55

3.3 View ..................................... 56
           3.3.1 Admin View ..................... 58
           3.3.2 CEO (Chief Executive Officer)
                View .......................... 59
           3.3.3 Client View .................... 59

3.4 Controller .................................. 61
           3.4.1 Admin Controller ................. 61
           3.4.2 CEO (Chief Executive Officer)
                Controller ....................... 62
           3.4.3 Interface Controller ............. 64

3.5 Database .................................. 65
           3.5.1 User Table ....................... 69
3.5.2 UserToProject Table .................. 69
3.5.3 Url Table ......................... 70
3.5.4 Settings Table ..................... 70
3.5.5 Threads Table ..................... 71
3.5.6 Threadblob Table ................. 72
3.5.7 Statistics Table .................. 72
3.5.8 Setmethod Table .................. 73
3.5.9 Definethread Table ............... 73
3.5.10 SetTeamName Table ............... 74
3.5.11 SelectTeam Table ................ 75
3.5.12 Messageboard Table .............. 75
3.5.13 Approvethread Table ............. 76
3.5.14 Note Table ....................... 76
3.5.15 Project Table ................... 77
3.5.16 Cost Table ....................... 77
3.5.17 Hours Table ....................... 78
3.5.18 ChatScript Table ................ 78
3.5.19 ChatUpload Table ................. 79
3.5.20 ChatUploadBlob Table .......... 79
3.5.21 Dependency Table ................. 80

CHAPTER FOUR: TESTING AND EVALUATION

4.1 Interface Testing ..................... 81
4.1.1 Purpose ......................... 81
4.1.2 Procedure ........................................ 82
4.1.3 Expected Results ............................... 82

4.2 Input Testing ........................................ 82
4.2.1 Purpose ............................................ 82
4.2.2 Procedure ........................................ 82
4.2.3 Results ............................................ 83
  4.2.3.4 Invalid Login ................................. 83
  4.2.3.2 Invalid Register ............................. 84
  4.2.3.3 Invalid Project Setting ................. 85
  4.2.3.4 Invalid File ................................. 86
  4.2.3.5 Blank Notes/Tasks ........................ 87
  4.2.3.6 Blank Message ................................ 87
  4.2.3.7 Left Blank .................................... 88
  4.2.3.8 Invalid Register ............................ 89
  4.2.3.9 Code as Input ................................ 90
  4.2.3.10 Invalid Hours and Budget ............ 92
  4.2.3.11 Left Blank on Method .................... 94
  4.2.3.12 Invalid Dependent ....................... 95

4.3 Sequence Testing and Scenario .................. 97
4.3.1 Purpose .......................................... 97
4.3.2 ReMoTe Scenario ................................ 97
  4.3.2.1 Admin Scenario .............................. 98
4.3.2.2 CEO (Chief Executive Officer) Scenario .................. 98
4.3.2.3 Project Manager Scenario .................. 98
4.3.2.4 Team Leader/Sub-Team Leader Scenario .................. 100
4.3.2.5 Software Engineer Scenario .................. 100

4.3.3 Screen Shot .................. 102
4.3.3.1 Screen1 Administrator Login .................. 102
4.3.3.2 Screen2 Create Project .................. 103
4.3.3.3 Screen13 Assign Users to Projects .................. 104
4.3.3.4 Screen4 Select Projects .................. 105
4.3.3.5 Screen5 Setup Project .................. 106
4.3.3.6 Screen6 Select Iteration .................. 107
4.3.3.7 Screen7 Set Method/Number of Phases .................. 108
4.3.3.8 Screen8 Set Phase Information .................. 109
4.3.3.9 Screen9 Select Teammates .................. 110
4.3.3.10 Screen10 Approve Team Leader Thread .................. 111
4.3.3.11 Screen11 Manage My Thread .................. 112
4.3.3.12 Screen12 Manage Dependencies .................. 113

CHAPTER FIVE: CONCLUSIONS .................. 114
5.1 Results ........................................ 114
5.2 Future Directions ............................. 118
APPENDIX A: FILE STRUCTURE ................. 120
APPENDIX B: CONFIGURATION AND INSTALLATION
PROCEDURES .................................. 132
REFERENCES .................................... 153
**LIST OF TABLES**

Table 1. Definitions, Acronyms, and Abbreviations ... 5
Table 2. User Table ........................................ 69
Table 3. UserToProject Table .............................. 70
Table 4. Url Table .......................................... 70
Table 5. Settings Table ..................................... 70
Table 6. Threads Table ...................................... 71
Table 7. ThreadBlob Table .................................. 72
Table 8. Statistics Table .................................. 72
Table 9. Setmethod Table .................................. 73
Table 10. Definethread Table .............................. 74
Table 11. SetTeamname Table .............................. 74
Table 12. SelectTeam Table ............................... 75
Table 13. Messageboard Table ............................. 75
Table 14. Approvethread Table ............................ 76
Table 15. Note Table ....................................... 77
Table 16. Project Table ................................... 77
Table 17. Cost Table ....................................... 78
Table 18. Hours Table ...................................... 78
Table 19. ChatScript Table ................................ 79
Table 20. ChatUpload Table ................................ 79
Table 21. ChatUploadBlob Table ......................... 79
Table 22. Dependency Table ............................... 80
LIST OF FIGURES

Figure 1. Deployment Diagram .................................. 10
Figure 2. Use-Case Diagram ...................................... 16
Figure 3. Main Banner ............................................ 18
Figure 4. Login Page for Users. .................................. 18
Figure 5. Login Page for Administrator. ....................... 19
Figure 6. Login Page for
CEO (Chief Executive Officer) ............................. 20
Figure 7. Welcome Screen for System Administrator ......... 20
Figure 8. Administrator Project Settings. ..................... 21
Figure 9. Administrator Edit Project .......................... 22
Figure 10. Administrator Assign Users to Projects ............ 23
Figure 11. Administrator Assign Users to Projects .......... 24
Figure 12. Administrator Edit Users ........................... 24
Figure 13. Administrator Edit Users ........................... 25
Figure 14. Administrator Edit Message Board ................. 25
Figure 15. Administrator Edit Message Board ................. 26
Figure 16. CEO (Chief Executive Officer)
Welcome Screen ............................................. 26
Figure 17. Dependencies Graph .................................. 27
Figure 18. Dependencies Graph Moved Iteration Page ...... 28
Figure 19. Iteration Page Moved Multiple Project ............ 29
Figure 20. CEO (Chief Executive Officer) Setting
 Dependencies and Independencies ......................... 29
Figure 21. Iteration Page Moved Multiple Project . . . 30
Figure 22. Project Manager Setting Page . . . . . . . 31
Figure 23. Set Name and Number of Phases Step 1 . . . 32
Figure 24. Set Name and Number of Phases Step 2 . . . 33
Figure 25. Set Method to Users Step 3 . . . . . . . . . . 34
Figure 26. Set Team Names Screen . . . . . . . . . . 34
Figure 27. Set Budget Screen . . . . . . . . . . . . . . 35
Figure 28. Set Man-Hours Screen . . . . . . . . . . 36
Figure 29. Manage Thread Screen . . . . . . . . . . 37
Figure 30. Approve and Deny Phases Screen . . . . . . 38
Figure 31. Chat Room . . . . . . . . . . . . . . . . . . 39
Figure 32. Create Message Board . . . . . . . . . . 40
Figure 33. Message Board List . . . . . . . . . . . . . 41
Figure 34. Create Notes/Tasks . . . . . . . . . . . . . 42
Figure 35. View Multiple Project . . . . . . . . . . 43
Figure 36. Gantt and Activity Chart . . . . . . . . . . 44
Figure 37. ReMoTe MVC Architecture Design . . . . . 46
Figure 38. ReMoTe Package Diagram . . . . . . . . . 47
Figure 39. Admin Model (Action) Class Package . . . . 48
Figure 40. CEO Model (Action) Class Package . . . . . 50
Figure 41. Client Model (Action) Class Package . . . . 52
Figure 42. GUI Architecture . . . . . . . . . . . . . . 57
Figure 43. Administration Interface Package . . . . . 58
Figure 44. CEO View Class Package .............. 59
Figure 45. Client View Class Package .......... 60
Figure 46. AdminController ..................... 62
Figure 47. CEOController .......................... 63
Figure 48. InterfaceController Class .......... 64
Figure 49. Setup File for ReMoTe's Database ... 66
Figure 50. Database Architecture ................. 67
Figure 51. Entity Relationship Diagram of the Database .............. 68
Figure 52. Invalid Login ............................ 83
Figure 53. Left Blank ............................... 84
Figure 54. Passwords Do Not Match ................. 84
Figure 55. Invalid Email ............................. 85
Figure 56. Number Iterations was Left Blank .... 86
Figure 57. Invalid File ............................... 86
Figure 58. Title or Note was Left Blank ........... 87
Figure 59. Subject or Message was Left Blank .... 88
Figure 60. Note Left Blank on Approval .......... 89
Figure 61. Left Blank on Define Method .......... 90
Figure 62. Code as Input at Message Page ....... 91
Figure 63. Code as Input at Note/Task Page ....... 92
Figure 64. Invalid Costs ............................. 93
Figure 65. Invalid Hours ............................. 94
Figure 66. Invalid Method Selections ............. 95
Figure 67. Estimated Delivery Time ............... 96
Figure 68. Invalid Dependencies Selection ........ 97
Figure 69. Sequence Diagram for All Users ........ 101
Figure 70. Screen1 Administrator Login .......... 102
Figure 71. Screen2 Create Project with Estimated Day ................. 103
Figure 72. Screen3 Assign Users to Projects ........ 104
Figure 73. Screen4 Select Projects ............... 105
Figure 74. Screen5 Setup Project ................. 106
Figure 75. Screen6 Select Iteration .............. 107
Figure 76. Screen7 Set Method/Number of Phases ....... 108
Figure 77. Screen8 Set Phase Information ........ 109
Figure 78. Screen9 Select Teammates ............ 110
Figure 79. Screen10 Approve Team Leader Thread .... 111
Figure 80. Screen11 Manage My Thread .......... 112
Figure 81. Screen12 Manage Dependencies ........ 113
CHAPTER ONE

INTRODUCTION

1.1 History

The Recursively Estimating Multi-threaded Observation Technology Enterprise (ReMoTe) is based on the Recursive Multi-Threaded (RMT) software life-cycle model, that was developed by Scott Simon [1] 1997. RMT monitors the software process through threads of the software life-cycle where each member of the project organization is assigned a thread. In 1998, Chung-Ping Lin [2] developed a computer aided software engineering tool used for monitoring and predicting progress based on RMT. RMT supported software process that is based on object-oriented approach. Yi-Chiun Kuo [4] in 2002 added the multi-database function which is used to support software process management tool in multiple software development sites. In 2006, Darrion DeMelo [5] re-implemented RMT with easy to use interface and useful functionalities using PHP, and created a complete tool to support software process management called ReMoTe (Recursively Estimating Multi-Threaded Observation Tool Enterprise). The students who took the CSE655 and
CSE455 classes supervised by Dr. Concepcion succeeded in several enhancements as class projects.

1.2 Purpose of the Project

The long term goal of the project is to have ReMoTe be commercialized or released as open source at some future date to the general public. In 2007, ReMoTe was chosen to be presented at the Worlds Best Technologies (WBT2007) conference as one of the software technologies that has a potential of being commercialized [3]. The short term goal of the project is to make enhancements to the current project so that it can be used as a resource for project management of software projects. ReMoTe will provide an enhanced user interface that will allow users easy to use functions and setup their projects and define their thread and phases. ReMoTe will be released with current bugs fixed and crashes removed and develop new functions. ReMoTe will have the ability to handle multiple projects, and specify any number of phases for any thread.

1.3 Improvements to ReMoTe

ReMoTe will provide users with a detailed look at the progress of a software project by giving management a
progress report in percentage of individual member, team(s), and the entire project. The project will also generate Gantt charts, and provide storage and retrieval of artifacts submitted to ReMoTe by all users in the project. Also, ReMoTe will help users to be reminded of the deadlines by setting start day and end day of project.

Another improvement to ReMoTe is to be able to handle multiple projects effectively. Once the system administrator assigns a user to several projects, a user is able to choose a list of assigned projects.

ReMoTe would create dependencies graph for multiple projects so that the CEO can view critical path for the projects. The CEO will be able to check dependencies of projects, total development days, iterations, and critical path.

The effectiveness of ReMoTe will be improved by providing an easy-to-use interface that will help all users to define and setup their projects. There are two different user interfaces. One for the users and the other is for the system administrator and the CEO.

The Master Project documents will have six chapters. Chapter One introduces the history and purpose of ReMoTe. Chapter Two will be the Software Requirement Specifications
(SRS) of ReMoTe. Chapter Three will present the architectural design. Chapter Four illustrates testing and evaluation. Chapter Five presents the conclusions of the current ReMoTe project and future directions. The rest of the document includes appendices and references.
CHAPTER TWO
SOFTWARE REQUIREMENTS SPECIFICATION

2.1 Scope

The new version of ReMoTe will provide an improved easy-to-use interface than the previous version. The user interface will provide users with easy to understand usage of ReMoTe to define projects and threads. ReMoTe will have an improved system by releasing bugs and crashes, and help users in monitoring activity and tracking deadline for a project.

2.1.1 Glossary

Table 1. Definitions, Acronyms, and Abbreviations

<table>
<thead>
<tr>
<th>Apache</th>
<th>An open source web server. Mostly for Unix, Linux and Solaris platforms.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Browser</td>
<td>A program which allows a person to read hypertext. The browser gives some means of viewing the contents of nodes (or &quot;pages&quot;) and of navigating from one node to another.</td>
</tr>
<tr>
<td>Cookie</td>
<td>Information from a web server, stored on your computer by your web browser. The purpose of a cookie is to provide information about your visit to the website for use by the server during a later visit.</td>
</tr>
<tr>
<td>CSS (Cascading Style Sheets)</td>
<td>a style sheet language used to describe the presentation semantics (the look</td>
</tr>
<tr>
<td><strong>CVS (Concurrent Versions System)</strong></td>
<td>CVS is a source control tool which allows multiple people to simultaneously view and edit code. CVS keeps a history of all changes that have been made to the code, along with who make the change and when it was committed into the repository.</td>
</tr>
<tr>
<td><strong>DAO (Data Access Objects)</strong></td>
<td>The DAO is a group of object which implements the access mechanism required to work with the data source.</td>
</tr>
<tr>
<td><strong>Deployment Diagram</strong></td>
<td>Deployment diagrams serve to model the hardware used in system implementations and the associations between those components.</td>
</tr>
<tr>
<td><strong>ER diagram</strong></td>
<td>Diagrams that use Entity-Relationship model to design or describe database.</td>
</tr>
<tr>
<td><strong>GUI</strong></td>
<td>Graphical User Interface.</td>
</tr>
<tr>
<td><strong>HTML</strong></td>
<td>Acronym for Hypertext Markup Language, the authoring language used to create documents on the World Wide Web.</td>
</tr>
<tr>
<td><strong>HTTP</strong></td>
<td>The standard set of rules for sending text files across the Internet. It requires an HTTP client program at one end, and an HTTP server program at the other end.</td>
</tr>
<tr>
<td><strong>hyperlink</strong></td>
<td>A pointer to another document. Most often a pointer to another web page. A hyperlink is a synonym for a hotlink or a link, and sometimes called a hypertext connection to another document or web page.</td>
</tr>
<tr>
<td><strong>IE</strong></td>
<td>Internet Explorer</td>
</tr>
<tr>
<td><strong>Interface</strong></td>
<td>The communication boundary between two entities, such as a piece of software and a user.</td>
</tr>
<tr>
<td><strong>Iteration</strong></td>
<td>Iteration is the repetition of a process. It describes a specific form of repetition with a mutable state. It also can be considered as a different version of a project.</td>
</tr>
<tr>
<td><strong>Javascript</strong></td>
<td>JavaScript is a scripting language most often used for client-side web...</td>
</tr>
</tbody>
</table>
development. It is able to interact with many HTML elements to make a web page more interactive to the user. "JavaScript" is a trademark of Sun Microsystems. It was used under license for technology invented and implemented by Netscape Communications and current entities such as the Mozilla Foundation. [3]

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linux</td>
<td>A multi-user Unix-style operating system</td>
</tr>
<tr>
<td>Menu Item</td>
<td>The individual element of a menu.</td>
</tr>
<tr>
<td>Mozilla Firefox</td>
<td>A free, cross-platform, graphical web browser that complies with many of today's standards on the worldwide web.</td>
</tr>
<tr>
<td>MS</td>
<td>Microsoft.</td>
</tr>
<tr>
<td>Multi-Database</td>
<td>A multi-database system is an environment which data is stored in two or more database instances are accessible as though these data were in a single instance.</td>
</tr>
<tr>
<td>MySQL</td>
<td>Open-source Structured Query Language database.</td>
</tr>
<tr>
<td>ODBC</td>
<td>A standard for accessing different database systems. The goal of ODBC is to make it possible to access data from any application, regardless of which database management system is handling the data.</td>
</tr>
<tr>
<td>OO</td>
<td>Object Oriented.</td>
</tr>
<tr>
<td>OS</td>
<td>Operating System.</td>
</tr>
<tr>
<td>PHP</td>
<td>Hypertext Preprocessor is a widely used, general-purpose scripting language that was originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document.</td>
</tr>
<tr>
<td>Random Access</td>
<td>The ability to access a random element of a group in equal time.</td>
</tr>
<tr>
<td>Redirect</td>
<td>In web terms: The action when a web page automatically forwards (redirects)</td>
</tr>
</tbody>
</table>
| **ReMoTe** | Recursively Estimating Multi-threaded Observation Technology Enterprise
ReMoTe Utilizes the RMT software life-cycle that supports the monitoring of progress during development and addresses the specific needs of the developing object-oriented software. |
| **RMT** | Recursive Multi-threaded Tool. |
| **Software Requirement Specification** | An SRS is used to describe all the tasks that go into the instigation, scoping, and definition of a new or altered computer system. |
| **SQL** | Structured Query Language |
| **SRS** | An acronym for Software Requirement Specification. |
| **Sub Menu** | A menu subordinate to another menu. |
| **TCP/IP** | A collection of Internet communication protocols between two computers. The TCP protocol is responsible for an error free connection between two computers, while the IP protocol is responsible for the data packets sent over the network. |
| **Thread** | A thread contains a person’s individual artifact(s) for a given iteration for a prototype in their project. |
| **UML** | Unified Modeling Language |
| **Virtual Machine** | Software that creates an environment between the computer platform and the end user, which the end user can operate software. |
| **World Wide Web Consortium** | An international organization that works to define standards for the worldwide web. |
2.1.2 Overview

The following sections contain the product perspectives, functions, user characteristics, interface, and specific requirements.

2.2 Overall Description

2.2.1 Product Perspective

ReMoTe will provide a better easy-to-use interface than the previous version. The new interface will allow users to understand the usage of ReMoTe and define their threads and phases easily. ReMoTe will have an improved system by releasing bugs and crashes, and help users in monitoring activity and tracking deadlines for a project. The project will also support multiple projects and view, indicate total days, graphs with critical path and Gantt charts.

2.2.2 System Interfaces

The system interfaces of ReMoTe displays in Web browser, such as Firefox, Internet Explorer, Google Chrome, Safari, etc. The Web server communicates with the client using http or https. Also, the Oracle database server communicates via PHP and needs ODBC Socket to connect with Apache. ReMoTe using different databases can have multiple
instances and allow managers to view multiple projects at once. Figure 1 shows how clients are able to view ReMoTe's server.

![Figure 1. Deployment Diagram](image)

2.2.3 User Interfaces

The different levels of users will have different user interfaces. There are three login interface pages which are System Administrator, CEO, and Users. The System Administrator can create multiple projects and assign users to projects and roles. The dependencies graph for the CEO includes critical path and delivery times. All users can see the estimated duration time for each of their projects. The following sections detail the user interfaces:
2.2.3.1 System Administrator
   a. Create projects.
   b. Edit Projects.
   c. Assign Users.
   d. Edit Users.
   e. Edit Message(s).

2.2.3.2 CEO (Chief Executive Officer)
   a. View Dependencies.
   b. Set Dependencies.

2.2.3.3 Project Manager
   a. Set Welcome Message.
   b. Project Settings
      - Set Bugzilla Location (optional).
      - Set Number of Iterations in Projects.
      - Set Message to Everyone.
   c. View Project.
   d. View Progress Chart.
   e. Chat Room.
   f. Write Message.
   g. View Messages.
   h. Set Notes/Tasks.
   i. Set Team Names.
   j. Set Budget.
k. Set Hours.

l. Define life-cycle model.
   - Set Name and Number of Phases.
   - Set Phase Information.
   - Set Method to Users (Select Teammates).

m. Approve and Deny members’ Threads.

n. Manage Threads.
   - Uploading and Deleting software artifacts.

2.2.3.4 Team Leader/Sub-Team Leader

a. View Project.

b. View Progress Chart.

c. Chat Room.

d. Write Message.

e. View Messages.

f. Set Notes/Tasks.

g. Set Team Names

h. Define life-cycle model.
   - Set Name and Number of Phases.
   - Set Phase Information.
   - Set Method to Users (Select Teammates).

i. Approve and Deny members’ Threads.

j. Manage Threads.
   - Uploading and Deleting software artifacts.
2.2.3.5 Employee

   a. View Project.
   b. View Progress Chart.
   c. Chat Room.
   d. Write Message.
   e. View Messages.
   f. Set Notes/Tasks.
   g. Manage Threads.

   - Uploading and Deleting software artifacts.

2.2.4 Hardware Interfaces

   There are no hardware interfaces.

2.2.5 Software Interfaces

   The project software interface can run on any Web browser for Windows, Linux, or Mac OS. The current user interface of ReMoTe is designed to run on Windows or Linux platform using Apache and PHP engine. The application platform is PHP, JavaScript, Flash, and CSS.

   2.2.5.1 Communication Interface ReMoTe requires the operating system and Hypertext Pre-Processor (PHP) to manage the communication between the client and the server. The communication interface between PHP and the Microsoft Access database goes through ODBC. In addition, MySQL and
Oracle database will use PHP’s built in function to communicate.

2.2.5.2 Memory and Hardware Constraints The server requires the following components.

For the server with MySQL/Access database:

- 256 MB or greater memory size.
- PII 500 or greater architecture machine.

For the server with Oracle 9i database:

- 512 MB or greater memory size.
- PIII 1 GHz or greater architecture machine.

For the client:

- 128 MB or greater memory size.

2.2.5.3 Operations ReMoTe will operate 24/7, and backup of the database will be done twice a month. Also, maintenance will be done once a month.

2.2.5.4 Adaptation Requirements There is no site adaptation at this time.

2.2.6 Product Functions

The use case diagram describes the users’ functionalities of ReMoTe. The system administrator only creates projects and assigns users to projects with roles. The project manager can only set the project and define life-cycle model and approve/deny members’ threads. The CEO
can view the special graph with critical path analysis of a set of projects. The team leader and sub-team leader will have project managers' functions except setting project. Other users can manage their threads such as adding/deleting software artifacts (See Figure 2.).
Figure 2. Use-Case Diagram
2.2.7 User Characteristics

ReMoTe is designed for the software engineers, including the CEO. The user should be familiar with computer operations over the Web.

2.2.8 Constraints

This project will not support maintainability. It will need to be re-engineered to support maintenance.

2.2.9 Assumptions and Dependencies

There are no assumptions and dependencies.

2.2.10 Apportioning of Requirements

There is no apportioning of requirements.

2.3 Specific Requirements

2.3.1 External Interface

Overall the user interface of ReMoTe will change. ReMoTe will use these interfaces to get information or to update data. The main banner of ReMoTe will appear on top of the user page (See Figure3.).
2.3.2 User Interface

2.3.2.1 Login Interface All users should access the login site for the first time and have a username and password to use the ReMoTe tool. If they do not have them, they need to register so that the admin will assign them to a project with defined roles (see Figure 4, 5, 6).
Figure 5. Login Page for Administrator
2.3.2.2 Administrator Interface Once the system administrator has logged into ReMoTe, they will have the privilege to use the system menu (See Figure 7.).
2.3.2.2.1 Create Project  The system administrator must first create a project with estimated days to the system (See Figure 8.).

Figure 8. Administrator Project Settings

2.3.2.2.2 Edit Projects  The administrator can modify the project information supplied to ReMoTe.
2.3.2.2.3 Assign Users The database should allow ReMoTe to assign users to projects, and store it accordingly. An assigned user to a project can keep track of all records related to his/her role in the project assigned to the user. The following data can be stored or retrieved by ReMoTe within Assign User to Project: User_Id, ProjectName, and Role (See Figure 10 and 11.).
### Figure 10. Administrator Assign Users to Projects

<table>
<thead>
<tr>
<th>Name</th>
<th>Username</th>
<th>Project Name/Type</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>p1Arturo</td>
<td>p1concepcion</td>
<td>ReMoTe/PM</td>
<td>Assign, Delete</td>
</tr>
<tr>
<td>p2Hong</td>
<td>p2taek</td>
<td>ReMoTe/SE</td>
<td>Assign, Delete</td>
</tr>
<tr>
<td>p3Hong</td>
<td>p3taek</td>
<td>ReMoTe/SE</td>
<td>Assign, Delete</td>
</tr>
<tr>
<td>p4Hong</td>
<td>p4taek</td>
<td>ReMoTe/SE</td>
<td>Assign, Delete</td>
</tr>
<tr>
<td>p5Hong</td>
<td>p5taek</td>
<td>ReMoTe/SE</td>
<td>Assign, Delete</td>
</tr>
<tr>
<td>p6Hong</td>
<td>p6taek</td>
<td>ReMoTe/SE</td>
<td>Assign, Delete</td>
</tr>
</tbody>
</table>

- **Assign**
- **Delete**
2.3.2.2.4 Edit Users The administrator can modify the user’s information supplied to ReMoTe (See Figure 12 and 13).
2.3.2.2.5 Edit Messages The administrator can modify the message board written by users (See Figure 14 and 15).
2.3.2.3 CEO (Chief Executive Officer) Interface

Once the CEO has logged onto ReMoTe, they will have the privilege to use Dependencies Menu (See Figure 16.).

Figure 15. Administrator Edit Message Board

Figure 16. CEO (Chief Executive Officer) Welcome Screen
2.3.2.3.1 View Dependencies Once the CEO sets dependencies, they will be able to view dependencies graph with critical path and values (days) (See Figure 17.). Also, when the CEO selects one of the nodes, ReMoTe will show the Iterations that were done (see Figure 18). Clicking the Iterations will move the page (See Figure 19.).

![Dependencies Graph](image)

Figure 17. Dependencies Graph
Figure 18. Dependencies Graph moved Iteration page
2.3.2.3.2 Set Dependencies In order to view dependencies graph in the ReMoTe system, CEO must first set up dependencies of the projects (See Figure 20.).
2.3.2.4 Project Manager/Team Leader/Sub-Team Leader

Once the Project Manager/Team Leader/Sub-Team Leader has logged onto ReMoTe, they will be able to select the list of projects (See Figure 21.).

![Figure 21. Iteration page moved multiple projects](image)

2.3.2.4.1 Project Manager

Once the Project Manager selects the project, he/she will be able to manage the project to set up the iteration number, message and Bugzilla location. After the Project Manager completes the project setting, other users can successfully access ReMoTe (See Figure 22.).
2.3.2.4.2 Define Life-Cycle Model The Project Manager and Team Leader can define the life-cycle model for the software engineers' threads. The software engineers should be one level below them. The manager and leaders follow three steps to create a life-cycle:

- Step 1. Set Name and Number of Phases (See Figure 23.).
- Step 2. Set Phase Information (See Figure 24.).
- Step 3. Set Method to Users (See Figure 25.).
Figure 23. Set Name and Number of Phases Step 1
**Figure 24. Set Phase Information Step 2**

<table>
<thead>
<tr>
<th>Phase 1 Name:</th>
<th>coding1</th>
<th>Start Date:</th>
<th>1 Mar 2011</th>
<th>End Date:</th>
<th>1 Apr 2011</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase 2 Name:</td>
<td>coding2</td>
<td>Start Date:</td>
<td>1 Mar 2011</td>
<td>End Date:</td>
<td>1 Apr 2011</td>
</tr>
</tbody>
</table>
2.3.2.4.3 Set Team Names The project manager/team leader/sub-team leader can update the given team-names, and testing team (See Figure 26.).
2.3.2.4.4 Select Teammates The select teammates are located in the life-cycle model step 3. Once the life-cycle model is defined, the managers and team leaders will be given a list of members to select the members of the users' team that will be one level below the software engineers (See Figure 25.).

2.3.2.4.5 Set Budget The project and team leader can supply the budget for the entire project to be distributed to their teams (See Figure 27.).

Figure 27. Set Budget Screen
2.3.2.4.6 Set Man-Hours The project and team leader can supply the man-hours for the entire project to be distributed to their teams (See Figure 28.).

![Set Man-Hours Screen](image)

**Figure 28. Set Man-Hours Screen**

2.3.2.5 Manage My Thread Interface This Interface is used by all users so they can submit to ReMoTe all the software artifacts and deliverables. The software artifacts are uploaded to ReMoTe. Any file types can be uploaded to ReMoTe (See Figure 29.).
2.3.2.6 Approve TM Thread  The Project Manager/Team Leader/Sub-Team Leader will review the artifacts and send a comment on why the file was approved or denied (See Figure 30.).
2.3.2.7 Communication Interface This communication tools help users to contact each other during development.

2.3.2.7.1 Chat Room ReMoTe provides real-time communication with other developers and invites other users to join the chat room (See Figure 31.).
Figure 31. Chat Room

2.3.2.7.2 Write/View Message All members can use this message board to share their opinions or information (See Figure 32 and 33.).
Figure 32. Create Message Board
Figure 33. Message Board List

2.3.2.7.3 Set Notes/Tasks All users can use the notes/tasks system (See Figure 34.).
2.3.2.8 View Project All users can view the entire project and get the overall project progress, the teams' progress, and individual member's progress (See Figure 35.).
2.3.2.9 View Progress Chart All members can use the Gantt chart generator. The Gantt chart can show the defined project plan with threads and phases. Also, the chart shows the total number of days of the entire project will be completed (See Figure 36.).
2.3.3 Performance Requirements

There are no performance requirements in this project.

2.3.4 Logical Database Requirements

There are fourteen tables created to store all the information for ReMoTe. We will use MySQL, Microsoft Access, and Oracle database.

2.3.5 Design Constraints

ReMoTe will follow the MVC design pattern.

2.3.6 Software System Attributes
2.3.6.1 **Reliability** The servers running the ReMoTe will be completely functional.

2.3.6.2 **Availability** The ReMoTe can be used on a Windows 2000 machine IIS Web Server and later deployed to Windows 2003 or Windows 2008 machine IIS Web Server. Users are able to access ReMoTe through the World Wide Web to access.

2.3.6.3 **Security** ReMoTe supports different user access privileges and protects the user’s data.

2.3.6.4 **Portability** The PHP code, the server side program language, will be highly portable across hardware and software requirements.

2.3.6.5 **Maintainability** All modules will be reusable, and the code and documentation will store in CD. We will not support this non-functional requirement.
3.1 MVC Pattern

ReMoTe uses the Model View Controller (MVC) architecture. The Model is an object that represents data. The View displays information about the model. The Controller will cause change of the view and connect the model and the view. The ReMoTe MVC Architecture is shown in Figures 37 and 38.

![Diagram of MVC Architecture]

**Figure 37. ReMoTe MVC Architecture Design**
3.2 Model

The ReMoTe Model has four components: Admin Action, CEO Action, Database (DAO) Action, and User Action. Each action has several classes. The Admin Action has ten classes, and the CEO Action has three classes. The Database (DAO) Action has thirty-two classes, and the Client Action has one class. When the controller class sends their respective action according to what the user supplied, every action will be performed on ReMoTe. All DAO classes will be discussed in Section 3.5 Database.
3.2.1 Admin Model

The Admin Action has ten classes: EditUserAction, LoginAction, EditMessagesAction, DeleteMessageAction, AssignUserPageAction, editUserPageAction, EditProjectAction, DeleteUserAction, CreateProjectAction, and AssignedUserAction (See Figure 39.).
3.2.1.1 **AssignedUserAction** Displays the list of assigned users in the system to allow the administrator to choose a user to be assigned.

3.2.1.2 **AssignUserPageAction** Assigns the user to a project in the system.

3.2.1.3 **CreateProjectAction** Creates the project name.

3.2.1.4 **DeleteMessageAction** Deletes the thread in message board system.

3.2.1.5 **DeleteUserAction** Deletes users in the system.

3.2.1.6 **EditMessagesAction** Edits messages in the message board.

3.2.1.7 **EditProjectAction** Displays the list of current projects in the system to allow the administrator to edit.

3.2.1.8 **EditUserAction** Updates a user’s information, and edits username/password.

3.2.1.9 **EditUserPageAction** Displays the list of users to allow the administrator to edit.

3.2.1.10 **LoginAction** Manages the login for the administration system.
3.2.2 CEO Model

The CEO Action has four classes: LoginAction, ViewProjectAction, ViewDependenciesAction, and CreateDependenciesAction (See Figure 40.).

3.2.2.1 CreateDependenciesAction Displays to the CEO the critical path for all projects in the system.

3.2.2.2 LoginAction Validates the username and password to log onto the system.

3.2.2.3 ViewDependenciesAction Displays the entire projects node with critical path and estimated days to finish.

3.2.2.4 ViewProjectAction Displays multiple projects.
3.2.3 Client Model

The Client Action has thirty-two classes:
ApproveThreadAction, CostAnalysisAction, MessageBoardAction,
CreateThreadAction, CreateThreadAction, DeleteFileAction,
AddThreadAction, SetNumberPhasesAction,
OverridePhaseNamesAction, DefineThreadAction,
DeleteMethodAction, EditMethodAction, RegistrationAction,
GanttChartAction, LoginAction, NoteAction, DeleteNoteAction,
NoteListAction, EditNoteAction, ForgotPasswordAction,
SelectProjectAction, SelectTeamAction, CostAction,
SetDatesAction, HoursAction, SetPictureAction,
SetTeamNamesAction, SettingsAction, StatisticsAction,
ViewMessageAction, ViewMessageAction, ViewProjectAction
(See Figure 41.).
Figure 41. Client Model (Action) Class Package
3.2.3.1 ApproveThreadAction Approves the phases of a user's thread in the assigned project.

3.2.3.2 AddThreadAction Uploads the user's artifacts to ReMoTe.

3.2.3.3 CreateThreadAction Displays to the user the defined phases for the user to submit their software deliverables.

3.2.3.4 CostAction Submits the cost allotted for each individual team in the assigned project.

3.2.3.5 CostAnalysisAction Returns to the user the list of monetary value allotted for the user's team in the assigned project for review.

3.2.3.6 DeleteFileAction Allows the users to delete artifacts in their thread.

3.2.3.7 DeleteMethodAction Deletes a life-cycle method.

3.2.3.8 DefineThreadAction Allows setting of the number of phases and name of the life-cycle.

3.2.3.9 DeleteNoteAction Deletes a note/task defined by the user.

3.2.3.10 EditMethodAction Allows the user to modify the values in their life-cycle.

3.2.3.11 EditNoteAction Allows the users to modify the values in their supplied note/task.
3.2.3.12 ForgotPasswordAction Returns the user’s password that they have forgotten.

3.2.3.13 GanttChartAction Returns the user’s timeline for the phases of their assigned project in a Gantt chart format.

3.2.3.14 HoursAction Submits the man-hours allotted for the team in the assigned project.

3.2.3.15 LoginAction Validates the username and password to login onto ReMoTe.

3.2.3.16 MessageBoardAction Sends to the database a new thread to the assigned project’s message board.

3.2.3.17 MessageCheck Validates to show if the user receive a new message.

3.2.3.18 NoteAction Submits the defined notes/tasks

3.2.3.19 NoteListAction Displays the list of notes/tasks.

3.2.3.20 OverridePhaseNamesAction Allows the user to modify the names of the phases defined in the user defined life-cycle model.

3.2.3.21 RegistrationAction Submits the users contact information to ReMoTe.

3.2.3.22 SelectProjectAction Allows the user to move assigned project.
3.2.3.23 **SelectTeamAction** Allows the user to supply the team manager/Leader/Sub-leader they are responsible for.

3.2.3.24 **SetDatesAction** Sends the database the start date and number of days for a life-cycle.

3.2.3.25 **SetNumberPhasesAction** Submits the number of phases for a life-cycle.

3.2.3.26 **SetPictureAction** Allows the user to upload a photo to the assigned project.

3.2.3.27 **SetTeamNamesAction** Connects to the database and update the setTeamName table.

3.2.3.28 **SettingsAction** Allows the project manager to set the Bugzilla location, number of iterations in the assigned project, and an introduction message for ReMoTe’s users.

3.2.3.29 **statisticsAction** Displays the progress of the assigned project in a bar chart format.

3.2.3.30 **ViewMessageAction** Displays the user an individual message in the message board system.

3.2.3.31 **ViewMessagesAction** Displays the list of messages in the assigned project message board.

3.2.3.32 **ViewProjectAction** Displays the multiple projects.
3.3 View

View gives the presentation of the application based on what the model renders. The HTML files which are the extension *.inc are stored in an HTML folder, so the software engineers will get the benefit of determining the code between a PHP (ends with *.php) or a HTML file (*.inc). The View files are for the interface to render to the users.

There are template files inside, and the user will have two views based on their action: first-time view and submission view. All template files will have the extension *.tpl. When a user visits the first time view, the NULL values are sent to the action class to recognize it. All action class will recognize the NULL values and return input page to the user. When the value in submission view forms are sent to the action class, the action class processes the data. The valid data allows the sending of data to the database or the user is redirected to another page. If the data is invalid, the user will be halted in processing the date and redirected to a page determined by the controller or response to the user's invalid data (See Figure 42.).
Figure 42. GUI Architecture
3.3.1 Admin View

The Admin view has ten classes: editUser, login, intro, editMessages, assignUser, editRegistration, editProject, deleteUser, createProject, assignedUser (See Figure 43.).

Figure 43. Administration Interface Package
3.3.2 CEO (Chief Executive Officer) View

The Admin view has five classes: viewProject, login, intro, viewDependencies, createDependencies (See Figure 44.).

![CEO View Class Package](image)

Figure 44. CEO View Class Package

3.3.3 Client View

The client view has twenty-nine classes: approveThreads, costAnalysis, createMessage, createThread, definePhase, defineThread, editMethod, editRegistration, ganttChart, intro, login, noteList, password, register, selectProject, selectTeam, setCost, setDates, setHours, setNumberPhases, setPicture, setProjectWeights, setTeamNames, settings, setViewProjects, statistics, viewMessage, viewMessages, viewProject (See Figure 45.).
Figure 45. Client View Class Package
3.4 Controller

The Controller is an event provided by the user that causes change of the View. For every action a user sends to the model, the controller determines the layout of the response the user will receive based on the action of the user. That is, the controller identifies the operation and sends the information to the related action class. There are three classes in the controller: AdminController, CEOController, and InterfaceController.

3.4.1 Admin Controller

When a user clicks on a hyperlink on the Admin system, an operation will be sent to the Admin controller class. The Admin controller class then sends the user the respective actions according to what the user has supplied (See Figure 46).
3.4.2 CEO (Chief Executive Officer) Controller

When the CEO clicks on a hyperlink on the CEO system, an operation will be sent to the CEO controller class. The CEO controller class then sends the CEO the respective action according to what the CEO has supplied (See Figure 47.).
Figure 47. CEOController Class
3.4.3 Interface Controller

When a client clicks on a hyperlink on the client system, an operation will be sent to the client controller class. The client controller class then sends the client the respective actions according to what the client has supplied (See Figure 48.).

![Diagram of interface controller class](image)

Figure 48. InterfaceController Class
3.5 Database

The database in the ReMoTe system is able to handle storing or retrieving software artifacts and allow the communication between three different databases: MySQL, Microsoft Access, and Oracle. In order to use the database, the administrator and the CEO have to set up in the ReMoTe’s setup file (See Figure 49 for Setup File.).
<?php

/*****************************************************/
Thank you for choosing the ReMoTe tool as your project management tool. This is setup file the Administrator/IT to edit for the ReMoTe to work.
*****************************************************/

//put username to get into database
$usr = "root";

//put password to get into database
$pwd = "myPassword";

//Title of database
$db = "myDatabase";

//database type
// 1 == mySQL
// 2 == Microsoft Access
// 3 == Oracle
// 4 == Experimental
$db_type = "1";

///host of database
$host = "localhost";

//company name (Please use text only)
$companyname = "Master";

//Set username and password for administrator to create project managers
$admin_name="Admin";
$admin_pass="Admin";

//Set username and password for CEO
$CEO_name="ceo";
$CEO_pass="ceo";

******************************************************************************
People involved
Darrion DeMelo
David Hollingsworth
Chaz Lee
Norman Loenandi
******************************************************************************/
?>

Figure 49. Setup File for ReMoTe’s Database
The Database Access Object, call on the ConnectDAO class, to handle all of the queries to the different databases with all other DAO classes (See Figure 50 and 51.).

Figure 50. Database Architecture
Figure 51. Entity Relationship Diagram of the Database
3.5.1 User Table

The user table provides the detailed personal information of users (See Table 2.).

Table 2. User Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MySQL</td>
<td>Access</td>
</tr>
<tr>
<td>ID</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>username</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>password</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>firstName</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>lastName</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>middleInitial</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>phone</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>email</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>role</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>projectName</td>
<td>Varchar(255)</td>
<td>text</td>
</tr>
<tr>
<td>valid</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>address</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>pictureName</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>pictureType</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>pictureSize</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>pictureContent</td>
<td>blob</td>
<td>memo</td>
</tr>
</tbody>
</table>

3.5.2 UserToProject Table

UserToProject Table provides the users assigned to projects their roles: Project Manager or Software Engineer (See Table 3.).
Table 3. UserToProject Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>User_Id</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>Project_Name</td>
<td>Varchar (255)</td>
<td>text</td>
</tr>
<tr>
<td>Role</td>
<td>text</td>
<td>text</td>
</tr>
</tbody>
</table>

3.5.3 Url Table

The URL Table provides the URL information for the URL currently being used (See Table 4.)

Table 4. Url Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>text</td>
<td>text</td>
</tr>
</tbody>
</table>

3.5.4 Settings Table

The Setting Table provides the information the project manager has to set up, such as the number of iterations, message to the whole team and the location of Bugzilla which is a URL (See Table 5).

Table 5. Settings Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Comments</th>
</tr>
</thead>
</table>

70
### Table 6. Threads Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>MySQL</th>
<th>Access</th>
<th>Oracle</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td>PRIMARY</td>
</tr>
<tr>
<td>companyName</td>
<td>text</td>
<td>text</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td>projectName</td>
<td>Varchar(255)</td>
<td>text</td>
<td>varchar(255)</td>
<td>MUL</td>
</tr>
<tr>
<td>bugzilla</td>
<td>text</td>
<td>text</td>
<td>varchar(255)</td>
<td>Location of bugzilla</td>
</tr>
<tr>
<td>message</td>
<td>text</td>
<td>text</td>
<td>varchar(255)</td>
<td>Message to all the team numbers</td>
</tr>
<tr>
<td>numberIterations</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td></td>
</tr>
</tbody>
</table>

#### 3.5.5 Threads Table

The Thread Table provides all the information of the file which the user submits (See Threads Table 6.).
3.5.6 ThreadBlob Table

The ThreadBlob Table stored in the database is in the form of a Binary large Object to handle the user’s software artifacts (See ThreadBlob Table 7.).

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MySQL</td>
<td>Access</td>
</tr>
<tr>
<td>ID</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>content</td>
<td>blob</td>
<td>memo</td>
</tr>
<tr>
<td>fileId</td>
<td>int</td>
<td>int</td>
</tr>
</tbody>
</table>

3.5.7 Statistics Table

The Statistics Table contains all of the progress information for every individual user, the user’s team and the overall project (See statistics Table 8.).

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MySQL</td>
<td>Access</td>
</tr>
<tr>
<td>ID</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>iteration</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>companyName</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>projectName</td>
<td>Varchar (255)</td>
<td>text</td>
</tr>
<tr>
<td>percentage</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>userId</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>date</td>
<td>text</td>
<td>text</td>
</tr>
</tbody>
</table>
3.5.8 Setmethod Table

The Setmethod Table will supply the team's life-cycle model/method for the team. The project manager/team leader/sub-team leader must supply the team's life-cycle model/method in order for the team to submit software artifacts (See setmethod Table 9.).

Table 9. Setmethod Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>MySQL</th>
<th>Access</th>
<th>Oracle</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td>PRI</td>
</tr>
<tr>
<td>name</td>
<td>text</td>
<td>text</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td>iteration</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td>MUL</td>
</tr>
<tr>
<td>userId</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td></td>
</tr>
<tr>
<td>numberPhases</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td></td>
</tr>
<tr>
<td>projectName</td>
<td>varchar(255)</td>
<td>text</td>
<td>varchar (255)</td>
<td></td>
</tr>
</tbody>
</table>

3.5.9 Definethread Table

The Definethread Table provides the detail data for the estimated delivery day such as project start year, start month, start day and estimated accomplish period (See Definethread Table 10.).
Table 10. Definethread Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MySQL</td>
<td>Access</td>
</tr>
<tr>
<td>ID</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>iteration</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>userId</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>phaseNumber</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>phaseDays</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>startMonth</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>projectName</td>
<td>varchar (255)</td>
<td>text</td>
</tr>
<tr>
<td>methodId</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>startYear</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>startDay</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>phaseName</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>endDate</td>
<td>varchar (255)</td>
<td>text</td>
</tr>
<tr>
<td>endMonth</td>
<td>int</td>
<td>Int</td>
</tr>
<tr>
<td>endDay</td>
<td>int</td>
<td>int</td>
</tr>
</tbody>
</table>

3.5.10 SetTeamName Table

The SetTeamNames Table provides the team name and other necessary information (See SetTeamName Table 11.).

Table 11. SetTeamname Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MySQL</td>
<td>Access</td>
</tr>
<tr>
<td>ID</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>companyName</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>projectName</td>
<td>Varchar (255)</td>
<td>text</td>
</tr>
<tr>
<td>iteration</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>teamId</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>teamName</td>
<td>int</td>
<td>int</td>
</tr>
</tbody>
</table>
3.5.11 SelectTeam Table

The SelectTeam Table provides information of the user who has been selected by the project manager or team leader/sub-team leader (See SelectTeam Table 12.).

Table 12. SelectTeam Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Extra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Name</td>
<td>MySQL</td>
<td>Access</td>
</tr>
<tr>
<td>ID</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>managerId</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>subUserId</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>projectName</td>
<td>varchar(255)</td>
<td>text</td>
</tr>
<tr>
<td>iteration</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>methodId</td>
<td>int</td>
<td>int</td>
</tr>
</tbody>
</table>

3.5.12 Messageboard Table

The MessageBoard Table provides the information of the message board such as date, broadcast, replay ID, subject and the content of the message (See messageboard Table 13.).

Table 13. Messageboard Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Name</td>
<td>MySQL</td>
<td>Access</td>
</tr>
<tr>
<td>ID</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>userId</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>date</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>projectName</td>
<td>varchar</td>
<td>text</td>
</tr>
<tr>
<td>Field Name</td>
<td>Type</td>
<td>MySQL</td>
</tr>
<tr>
<td>------------</td>
<td>------</td>
<td>-------</td>
</tr>
<tr>
<td>broadcast</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>replyId</td>
<td>int</td>
<td>int</td>
</tr>
<tr>
<td>message</td>
<td>text</td>
<td>text</td>
</tr>
<tr>
<td>subject</td>
<td>text</td>
<td>text</td>
</tr>
</tbody>
</table>

3.5.13 Approvethread Table

The Approvethread Table has the user's approval state for each phase of the assigned project defined by their project manager/team leader/sub-team leader (See Approvethread Table 14.).

Table 14. Approvethread Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>MySQL</th>
<th>Access</th>
<th>Oracle</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>int</td>
<td>int</td>
<td>number</td>
</tr>
<tr>
<td>approve</td>
<td>int</td>
<td>int</td>
<td>int</td>
<td>number</td>
</tr>
<tr>
<td>iteration</td>
<td>int</td>
<td>int</td>
<td>int</td>
<td>number</td>
</tr>
<tr>
<td>userId</td>
<td>int</td>
<td>int</td>
<td>int</td>
<td>number</td>
</tr>
<tr>
<td>phaseTotal</td>
<td>int</td>
<td>int</td>
<td>int</td>
<td>number</td>
</tr>
<tr>
<td>viewNote</td>
<td>int</td>
<td>int</td>
<td>int</td>
<td>number</td>
</tr>
<tr>
<td>phaseId</td>
<td>int</td>
<td>int</td>
<td>int</td>
<td>number</td>
</tr>
<tr>
<td>Date</td>
<td>int</td>
<td>int</td>
<td>int</td>
<td>number</td>
</tr>
<tr>
<td>projectName</td>
<td>varchar(255)</td>
<td>text</td>
<td>varchar(255)</td>
<td></td>
</tr>
</tbody>
</table>

3.5.14 Note Table

The Note Table allows the team members to know the notes/tasks (See Note Table 15.).
Table 15. Note Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>number</td>
</tr>
<tr>
<td>projectName</td>
<td>Varchar (255)</td>
<td>varchar(255)</td>
</tr>
<tr>
<td>iteration</td>
<td>int</td>
<td>number</td>
</tr>
<tr>
<td>note</td>
<td>text</td>
<td>varchar(255)</td>
</tr>
<tr>
<td>title</td>
<td>text</td>
<td>varchar(255)</td>
</tr>
<tr>
<td>userId</td>
<td>int</td>
<td>number</td>
</tr>
<tr>
<td>dateCreated</td>
<td>int</td>
<td>number</td>
</tr>
</tbody>
</table>

3.5.15 Project Table

The Project Table has project name supplied by the System administrator and time (day). It will be used when the dependency graph computes the critical path (See Project Table 16.).

Table 16. Project Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>number</td>
</tr>
<tr>
<td>projectName</td>
<td>text</td>
<td>varchar(240)</td>
</tr>
<tr>
<td>time</td>
<td>int</td>
<td>number</td>
</tr>
</tbody>
</table>

3.5.16 Cost Table

The Cost Table contains all of the cost allocated for a team (See Cost Table 17.).
Table 17. Cost Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>MySQL</th>
<th>Access</th>
<th>Oracle</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td>PRI</td>
</tr>
<tr>
<td>projectName</td>
<td>varchar(255)</td>
<td>text</td>
<td>varchar(255)</td>
<td>MUL</td>
</tr>
<tr>
<td>iteration</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td>MUL</td>
</tr>
<tr>
<td>cost</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td></td>
</tr>
<tr>
<td>userId</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td></td>
</tr>
<tr>
<td>managerId</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td></td>
</tr>
</tbody>
</table>

3.5.17 Hours Table

The Hours Table has all of the man-hours required for an assigned team (See Hours Table 18.).

Table 18. Hours Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>MySQL</th>
<th>Access</th>
<th>Oracle</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td>PRI</td>
</tr>
<tr>
<td>projectName</td>
<td>varchar(255)</td>
<td>text</td>
<td>varchar(255)</td>
<td>MUL</td>
</tr>
<tr>
<td>iteration</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td>MUL</td>
</tr>
<tr>
<td>hours</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td></td>
</tr>
<tr>
<td>userId</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td></td>
</tr>
<tr>
<td>managerId</td>
<td>int</td>
<td>int</td>
<td>number</td>
<td></td>
</tr>
</tbody>
</table>

3.5.18 ChatScript Table

The ChatScript Table contains the user's text written in to the chat room (See ChatScript Table 19.).
Table 19. ChatScript Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>MySQL</th>
<th>Access</th>
<th>Oracle</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td></td>
<td>int</td>
<td>number</td>
<td>PRI</td>
</tr>
<tr>
<td>theNick</td>
<td>text</td>
<td></td>
<td>text</td>
<td>varchar(255)</td>
<td></td>
</tr>
<tr>
<td>projectName</td>
<td>varchar(255)</td>
<td>int</td>
<td>int</td>
<td>varchar(255)</td>
<td>MUL</td>
</tr>
<tr>
<td>chatTime</td>
<td>int</td>
<td></td>
<td>int</td>
<td>number</td>
<td></td>
</tr>
<tr>
<td>theText</td>
<td>text</td>
<td></td>
<td>text</td>
<td>varchar(255)</td>
<td></td>
</tr>
</tbody>
</table>

3.5.19 ChatUpload Table

The ChatUpload Table contains the data information, i.e. file name, file size, etc (See ChatUpload Table 20.).

Table 20. ChatUpload Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>MySQL</th>
<th>Access</th>
<th>Oracle</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td></td>
<td>int</td>
<td>number</td>
<td>PRI</td>
</tr>
<tr>
<td>fileSize</td>
<td>int</td>
<td></td>
<td>int</td>
<td>number</td>
<td></td>
</tr>
<tr>
<td>projectName</td>
<td>int</td>
<td></td>
<td>int</td>
<td>number</td>
<td>MUL</td>
</tr>
<tr>
<td>chatTime</td>
<td>int</td>
<td></td>
<td>int</td>
<td>number</td>
<td></td>
</tr>
</tbody>
</table>

3.5.20 ChatUploadBlob Table

The ChatUploadBlob stores all data in the form of a BLOB (See ChatUploadBlob Table 21.).

Table 21. ChatUploadBlob Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Comments</th>
</tr>
</thead>
</table>

3.5.21 Dependency Table

The Dependency Table contains all projects' time and dependencies (See Dependency Table 22.).

Table 22. Dependency Table

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Type</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>number</td>
</tr>
<tr>
<td>dependentProject</td>
<td>varchar(255)</td>
<td>text</td>
</tr>
<tr>
<td>independentProject</td>
<td>varchar(255)</td>
<td>int</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field Name</th>
<th>MySQL</th>
<th>Access</th>
<th>Oracle</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>int</td>
<td>int</td>
<td>number PRI</td>
</tr>
<tr>
<td>dependentProject</td>
<td>varchar</td>
<td>text</td>
<td>varchar(255)</td>
</tr>
<tr>
<td>independentProject</td>
<td>varchar</td>
<td>int</td>
<td>varchar(255)</td>
</tr>
</tbody>
</table>
CHAPTER FOUR

TESTING AND EVALUATION

In this chapter we will show how testing was done to ReMoTe and evaluate whether the software is ready for production.

4.1 Interface Testing

4.1.1 Purpose

The following are the tests made to verify the following requirements for each Web page:

1. Links
   a. All necessary links exist on page.
   b. All links work correctly.

2. Web Pages
   a. Verify all Web pages are used.
   b. Verify no additional Web pages are needed.

3. JavaScript
   a. Verify all JavaScript functions work based on user interactions.

4. Stylesheet
   a. Verify all interface design of the CSS.

5. HTML
   a. Verify all tables and images in pages are properly working
4.1.2 Procedure

1. Open Web browser such as Internet Explorer, Mozilla Firefox, Google Chrome, etc.

2. Open Web pages and perform the following checks.
   a. Insure the banner is correct for the page.
   b. Insure proper look of the interface.
   c. Check grammar and spelling.
   d. Verify all hyperlinks are valid.
   e. Verify all necessary links to other pages exist.

4.1.3 Expected Results

Each Web page, Internet Explorer, Mozilla Firefox, and Google Chrome, passed the test

4.2 Input Testing

4.2.1 Purpose

The following tests will check whether all pages can handle errors, and report how the system will display the errors to the user.

4.2.2 Procedure

1. Open a browser.

2. Open each page and put invalid data or blank fields, and check how the system handles the invalid information.
4.2.3 Results

4.2.3.1 Invalid Login When the users supply invalid login or left blank, ReMoTe system will report that the login is invalid (See Figure 52.) or left blank (See Figure 53.).

![Figure 52 Invalid Login](image-url)
4.2.3.2 Invalid Register When the user supplies a different password or invalid email, the system will report passwords that do not match (See Figure 54.) or email invalid (See Figure 55.).
4.2.3.3 Invalid Project Setting When the Project Manager does not provide information on the number of iterations, all other users cannot continue using ReMoTe system. The system will report “number Iterations was left black” (See Figure 56.).
Figure 56 Number Iterations was Left blank

4.2.3.4 Invalid File When the user submits file without attach file, ReMoTe will report invalid File (See Figure 57.).

Figure 57 Invalid File
4.2.3.5 Blank Notes/Tasks When the user does not provide the title or description, the system will report error (See Figure 58.).

Figure 58 Title or Note was Left blank

4.2.3.6 Blank Message When the user does not provide the subject or message, the system will report error (See Figure 59.).
Figure 59 Subject or Message was Left blank

4.2.3.7 Left Blank When the Project Manager or Team Leader/Sub-Team Leader leave the notes blank, the system will report that note was left blank (See Figure 60.).
Figure 60 Notes Left Blank on Approval

4.2.3.8 Invalid Register When the Project Manager or Team Leader/Sub-Team Leader does not provide the name or number of phases, the system will report that name or number of phases was left blank (See Figure 61.).
Figure 61 Left Blank on Define Method

4.2.3.9 Code as Input When the user provides input that contains any coding, ReMoTe will reject the input for security purposes (See Figure 62 and 63.).
Figure 62 Code as Input at Message page
Figure 63 Code as Input at Note/Task page

4.2.3.10 Invalid Hours and Budget If the amount or hours assigned to the teams does not match the given data, the values will be rejected and will need to try again with the correct values (See Figure 64 and 65.).
<table>
<thead>
<tr>
<th>ID</th>
<th>Team Name</th>
<th>Budget</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>profesion1</td>
<td>$10000</td>
</tr>
<tr>
<td>2</td>
<td>profesion2</td>
<td>$0</td>
</tr>
<tr>
<td>3</td>
<td>profesion3</td>
<td>$0</td>
</tr>
<tr>
<td>4</td>
<td>profesion4</td>
<td>$0</td>
</tr>
</tbody>
</table>

**Figure 64 Invalid Costs**
4.2.3.11 Left Blank on Method When the Project Manager or Team Leader/Sub-Team Leader does not select the method, the system will report that method was left blank (See Figure 66.).
Figure 66 Invalid Method Selections

4.2.3.12 Invalid Dependent When the CEO does not select Dependent and Independent project, ReMoTe will report the error and select a different project (see Figure 68). ReMoTe can compute the latest time (completion time) for the entire project. Figure 67 shows the estimated delivery time for a huge project. Using the critical path analysis algorithm in the example, the delivery of 55 days and the path from ReMoTe node to Project2 node to project3 node are computed. When the estimate delivery time was changed by the Administrator, ReMoTe would automatically re-compute the delivery path.
Figure 67 Estimated Delivery Time
4.3 Sequence Testing and Scenario

4.3.1 Purpose

This test will analyze whether ReMoTe system properly handles on a given user's scenario. Also, the sequence diagram will display sequence of four users (System Administrator, CEO, Project Manager, Team Leader/Sub-Team Leader, and Engineer) (See Picture 69.).

4.3.2 ReMoTe Scenario

The System Administrator has different Login interface as ReMoTE users.
4.3.2.1 Admin Scenario Once Admin log in to ReMoTe, he/she must create project first and assign user to project with roles, and then all user (Project Manager, Team Leader/Sub-Team Leader) can use ReMoTe system by their roles.

Major Work (Sequence):

1. Create Project with estimated day (See Figure 70 screen 2.).
2. Assign user to project with roles (See Figure 71 screen 3.).

4.3.2.2 CEO (Chief Executive Officer) Scenario Once the CEO log in to the ReMoTe system, the CEO must set dependent and independent up to display critical path and values for each project.

Major Work (Sequence): (See Figure 80 screen 12.)

1. Login to the CEO page.
2. Set dependencies to display the critical graph.
3. View dependencies with critical path for entire project and each project value.

4.3.2.3 Project Manager Scenario Once user allocates the project manager, he/she can log in to the system as project manager. After he/she select project on the list, the project manager must set project up for their team, and
then the ReMoTe will provide prototype to select iterations. The iterations will be maximum 25 to select by user. When the project manager gets iteration, the thread settings will be activated by the ReMoTe, and defines team’s thread, and phases, or allocates users to team thread and phases.

**Major Work (Sequence):**

1. Login to the ReMoTe system (See Figure 69 screen1.).

2. Select project on the list what projects assigned from Admin (See Figure 72 screen4).

3. Setup the project (Bugzilla Location, Number of Iterations, and Message) (See Figure 73 screen5.).

4. Select Prototype (Select iterations) (See Figure 74 Screen6.).

5. Thread Settings:
   
   a. Define Team Threads (See Figure. 75 Screen7.).

   b. Select Teammates (See Figure 76 Screen8.).

   c. Approve Team’s Treads (This option will display after set thread up) (See Figure 77 Screen9.).
6. Approve/Deny team’s Thread (See Figure 78 Screen10.).

4.3.2.4 Team Leader/Sub-Team Leader Scenario The leader users will have same scenario as Project Manager except number 3. Setup the project. The Setup the Project option is only for the project manager.

4.3.2.5 Software Engineer Scenario The software engineer will have Manage My Thread option that submits their artifacts (See Figure 79 Screen 11.).
Picture 69. Sequence Diagram for All users
4.3.3 Screen Shot

This Screen shot will help users to understand Sequence Diagram (See Picture 69.).

4.3.3.1 Screen1 Administrator Login Log-in site for System Administrator (See Picture 70.).

Figure 70. Screen1 Administrator Login
4.3.3.2 Screen2 Create Project The system administrator has to create project with estimated days (See Picture 71.).

Figure 71. Screen2 Create Project with Estimated Day
4.3.3.3 Screen3 Assign Users to Projects The system administrator have to assign users to projects (See Picture 72.).

Figure 72. Screen3 Assign Users to Projects
4.3.3.4 Screen4 Select Projects All users have to select the list of project (See Picture 73.).

Figure 73. Screen4 Select Projects
4.3.3.5 Screen5 Setup Project The Project Manager/Team Leader/Sub-Team Leader can manage the project to set up the iteration number, message and Bugzilla location (See Picture 74.).

Figure 74. Screen5 Setup Project
4.3.3.6 Screen6 Select Iteration  All users have to select the list of iterations (See Picture 75.).

Figure 75. Screen6 Select Iteration
4.3.3.7 Screen 7 Set Method/Number of Phases Step 1 of the Life-Cycle Model is setting name and number of phases (See Picture 76.).

Figure 76. Screen 7 Set Method/Number of Phases
4.3.3.8 Screen8 Set Phase Information Step 2 of the Life-Cycle Model is setting phase Information (See Picture 77.).

Figure 77. Screen8 Set Phase Information
4.3.3.9 Screen 9 Select Teammates

The Project manager/Team Leader/Sub-Team Leader can select users to defined methods (See Picture 78.).

Figure 78. Screen 9 Select Teammates
4.3.3.10 Screen10 Approve Team Leader Thread  The Project manager/Team Leader/Sub-Team Leader can review the artifacts and send a comment on why the file was approved or denied (See Picture 79.).

Figure 79. Screen10 Approve Team Leader Thread
4.3.3.11 Screen11 Manage My Thread All users can submit to ReMoTe all the software artifacts and deliverables (See Picture 80.).

Figure 80. Screen11 Manage My Thread
4.3.3.12 Screen12 Manage Dependencies CEO have to first set up dependencies of the projects in order to view dependencies graph in the ReMoTe system (See Picture 81.).

Figure 81. Screen12 Manage Dependencies
CHAPTER FIVE
CONCLUSIONS

5.1 Results

We need tools, such as ReMoTe, because there are big Software Engineer Projects that failed (See Table 24.).

Table 23. Software Hall of Shame

<table>
<thead>
<tr>
<th>YEAR</th>
<th>COMPANY</th>
<th>OUTCOME (COSTS IN US $)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td>Hudson Bay Co. [Canada]</td>
<td>Problems with inventory system contribute to $33.3 million loss.</td>
</tr>
<tr>
<td>2004-05</td>
<td>UK Inland Revenue</td>
<td>Software errors contribute to $3.45 billion tax-credit overpayment.</td>
</tr>
<tr>
<td>2004</td>
<td>Avis Europe PLC [UK]</td>
<td>Enterprise resource planning (ERP) system canceled after $54.5 million is spent.</td>
</tr>
<tr>
<td>2004</td>
<td>Ford Motor Co.</td>
<td>Purchasing system abandoned after deployment costing approximately $400 million.</td>
</tr>
<tr>
<td>2004</td>
<td>Hewlett-Packard Co.</td>
<td>Problems with ERP system contribute to $160 million loss.</td>
</tr>
<tr>
<td>2003-04</td>
<td>AT&amp;T Wireless</td>
<td>Customer relations management (CRM) upgrades problems lead to revenue loss of $100 million.</td>
</tr>
<tr>
<td>2002</td>
<td>McDonald’s Corp.</td>
<td>The Innovate information-purchasing system canceled after $70 million is spent.</td>
</tr>
<tr>
<td>Year</td>
<td>Company/Department</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>--------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>2002</td>
<td>Sydney Water Corp. [Australia]</td>
<td>Billing system canceled after $33.2 million is spent.</td>
</tr>
<tr>
<td>2002</td>
<td>CIGNA Corp.</td>
<td>Problems with CRM system contribute to $445 million loss.</td>
</tr>
<tr>
<td>2001</td>
<td>Nike Inc.</td>
<td>Problems with supply-chain management system contribute to $100 million loss.</td>
</tr>
<tr>
<td>2001</td>
<td>Kmart Corp.</td>
<td>Supply-chain management system canceled after $130 million is spent.</td>
</tr>
<tr>
<td>1999</td>
<td>United Way</td>
<td>Administrative processing system canceled after $12 million is spent.</td>
</tr>
<tr>
<td>1999</td>
<td>State of Mississippi</td>
<td>Tax system canceled after $11.2 million is spent; state receives $185 million damages.</td>
</tr>
<tr>
<td>1999</td>
<td>Hershey Foods Corp.</td>
<td>Problems with ERP system contribute to $151 million loss.</td>
</tr>
<tr>
<td>1998</td>
<td>Snap-on Inc.</td>
<td>Problems with order-entry system contribute to revenue loss of $50 million.</td>
</tr>
<tr>
<td>1997</td>
<td>U.S. Internal Revenue Service</td>
<td>Tax modernization effort canceled after $4 billion is spent.</td>
</tr>
<tr>
<td>1997</td>
<td>State of Washington</td>
<td>Department of Motor Vehicle (DMV) system canceled after $40 million is spent.</td>
</tr>
<tr>
<td>1997</td>
<td>Oxford Health Plans Inc.</td>
<td>Billing and claims system problems contribute to quarterly loss; stock plummets, leading to $3.4 billion loss in corporate value.</td>
</tr>
<tr>
<td>1996</td>
<td>Arianespace [France]</td>
<td>Software specification and design errors cause $350 million Ariane 5 rocket to explode.</td>
</tr>
<tr>
<td>1996</td>
<td>FoxMeyer Drug Co.</td>
<td>$40 million ERP system abandoned after deployment,</td>
</tr>
<tr>
<td>Year</td>
<td>Company or System</td>
<td>Notes</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>1995</td>
<td>Toronto Stock Exchange [Canada]</td>
<td>Electronic trading system canceled after $25.5 million is spent.</td>
</tr>
<tr>
<td>1994</td>
<td>U.S. Federal Aviation Administration</td>
<td>Advanced Automation System canceled after $2.6 billion is spent.</td>
</tr>
<tr>
<td>1994</td>
<td>State of California</td>
<td>DMV system canceled after $44 million is spent.</td>
</tr>
<tr>
<td>1994</td>
<td>Chemical Bank</td>
<td>Software error causes a total of $15 million to be deducted from 100,000 customer accounts.</td>
</tr>
<tr>
<td>1993</td>
<td>Allstate Insurance Co.</td>
<td>Office automation system abandoned after deployment, costing $130 million.</td>
</tr>
<tr>
<td>1993</td>
<td>Greyhound Line Inc.</td>
<td>Bus reservation system crashes repeatedly upon introduction, contributing to revenue loss of $61 million.</td>
</tr>
</tbody>
</table>

Software development companies need to monitor software development progress and manage the software management process. This could be done by using ReMoTe. ReMoTe provides easy access to users based on a Web-based
application. Project managers, team leaders, or sub-team leaders can monitor the project’s progress remotely and manage software engineers who are subordinates to project managers or team leaders.

ReMoTe can handle all thread event information on a software project anywhere where one can use the World Wide Web. The user can immediately see the effects of changes at the project level. Also, the user can view project details on any team’s task level. ReMoTe will display both individual’s and team’s percentage of progress. The management long term development of a huge and complex software project can be done easily by having CVS link that support team work. In addition, the overall project progress is calculated and displayed graphically. Moreover, ReMoTe will show the estimated delivery time of a huge project, and compute the delivery dates and the critical path by using the critical path analysis algorithm.

Comparing ReMoTe with Microsoft Project, ReMoTe is more detailed on personal progress management that is built by each member of the project and more unique functionalities when it comes to managing a software project (See Table 23.).
Table 24. Comparison of ReMoTe and Microsoft Project

<table>
<thead>
<tr>
<th>Features</th>
<th>ReMoTe</th>
<th>MS Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run on any platform (Linux, Windows, ...)</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Critical path analysis with graph</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Group communicated functionality</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Customize plans</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Scheduling Events</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Displaying individual progress</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Estimating overall progress of software project</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Scheduling events</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Organizing and retrieval of software artifacts</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Handle multiple iterations of a project</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

5.2 Future Directions

Include improvement of sequence of creation of threads:

1. Project manager defines his/her thread then assigns threads to team leader.

2. Team leader defines his/her thread then assigns threads to sub-team leader.

3. Sub-team leader defines his/her thread then assigns threads to software engineer.

4. Each thread at any level can have many iterations.
The function in the future that can be added is having a user have different roles in same project, so ReMoTe will display both individuals and teams percentage of progress of the same user with different roles.

Also, there is another improvement that can be made, which is re-engineering the MVC design. When ReMoTe needs to be maintained, it will be easier to do this by following strictly the separation of Model, View, and Controller components.
APPENDIX A

FILE STRUCTURE
The following is the file structure of ReMoTe showing where the files are located.

ReMoTe  //root directory
|-- DOCS //documentation files
     |-- CHANGES
     |-- TODO
         `-- masters
             |-- PreliminaryProposalFormat.doc
             |-- ReMoTe_SRS.doc
             `-- Summer.doc
|-- GanttChartProject.php
|-- README.txt //overall description of installation
|-- ShowImage.php
|-- bugzilla // bugzilla installation files
     |-- css
         |-- colors.css
         |-- layout.css
         |-- navigation.css
     |-- flash
         |-- header.fla
     |-- images
         |-- banner.jpg
         |-- CSUSB_logo_white_footer.gif
     |-- js
         |-- clientFunctions.js
         |-- horizontal_menu.js
         |-- jquery.js
         |-- mootoolssvn.js
         |-- old_jquery.js
         |-- script.js
     |-- sdmenu
         |-- bottoml.gif
         |-- collapsed.gif
         |-- expanded.gif
         |-- linkarrowl.gif
         |-- sdmenu.css
         |-- sdmenu.js
         |-- title.gif
         |-- toptitle.gif
|-- WEB-INF //source code
     |-- html //interface code
         |-- adminPages //administrator interface
             |-- acceptUser.inc
             |-- assignedUser.inc
| assignUser.inc |
| createMessage.inc |
| createProject.inc |
| deleteUser.inc |
| editMessages.inc |
| editRegistration.inc |
| editProject.inc |
| editUser.inc |
| intro.inc |
| layout //administrator template files |
| admin.tpl |
| logout.tpl |
| login.inc |
| register.inc |
| CEOPages //CEO interface |
| intro.inc |
| login.inc |
| viewDependencies.inc |
| viewProject.inc |
| layout //CEO template files |
| CEO.tpl |
| logout.tpl |
| clientPages //user interface |
| approveThreads.inc |
| costAnalysis.inc |
| createMessage.inc |
| createThread.inc |
| defineThread.inc |
| editMethod.inc |
| editRegistration.inc |
| editViewProjects.inc |
| ganttChart.inc |
| intro.inc |
| layout //user template files |
| client.tpl |
| logout.tpl |
| redirect.tpl |
| login.inc |
| noteList.inc |
| password.inc |
| redirectAdmin.inc |
| register.inc |
| selectTeam.inc |
| setCost.inc |
```
| | | |-- setDates.inc
| | | |-- setHours.inc
| | | |-- setNumberPhases.inc
| | | |-- setPicture.inc
| | | |-- setTeamNames.inc
| | | |-- settings.inc
| | | |-- statistics.inc
| | | |-- viewMessage.inc
| | | |-- viewMessages.inc
| | | `-- viewProject.inc
| `-- threadPages //thread output interface
| | |-- compress.inc
| | |-- createNotes.inc
| | |-- cvsgrab.inc
| | |-- cvsgrab.php
| | |-- cvsoutput.php
| | |-- ganttChart.inc
| | |-- rugzilla.php
| | |-- showFile.inc
| | |-- showFile.php
| | |-- showPhase.php
| | |-- showTeamList.inc
| | |-- statistics.inc
| | |-- threadLayout
| | | |-- bugzilla.tpl
| | | `-- thread.tpl
| `-- thread_top.php
|-- src //php source code
| |-- admin //administrator action files
| | |-- AcceptUserAction.php
| | |-- AssignedUserAction.php
| | |-- AssignUserAction.php
| | |-- CreateProjectAction.php
| | |-- DeleteMessageAction.php
| | |-- DeleteProjectAction.php
| | |-- DeleteUserAction.php
| | |-- EditMessageAction.php
| | |-- EditMessagesAction.php
| | |-- EditUserAction.php
| | |-- EditUserPageAction.php
| | |-- LoginAction.php
| | |-- RegistrationAction.php
| | |-- CEO //CEO action files
| | |-- CreateDependenciesAction.php
| | |-- LoginAction.php
```
|-- User.php
|-- UserDAO.php
|-- urlDAO.php
|-- UserToProject.php
|-- UserToProjectDAO.php
|-- thread //thread output action files
|-- GanttChartAction.php
|-- ShowFileAction.php
|-- ShowItem.php
|-- ShowNotesAction.php
|-- ShowTeamListAction.php
|-- StatisticsAction.php
|-- util
`-- DebugLogger.php
`-- static //non-chaning files i.e. images, scripts
|-- css //stylesheets
| |-- admin.css
| |-- CEO.css
| `-- clientDefault.css
|-- help //user help folder
| |-- About.php
| |-- README.html
| |-- frameMain.html
| |-- helpFiles
| | |-- FAQ.html
| | |-- apprrovethreads.html
| | |-- chatroom.html
| | |-- createcvs.html
| | |-- cvstutorial.html
| | |-- deadlines.html
| | |-- images
| | | |-- GanttChart.jpg
| | | |-- accept.jpg
| | | |-- addmessage.jpg
| | | |-- addthread.jpg
| | | |-- addthreadl.jpg
| | | |-- adminlogin.jpg
| | | |-- adminside.jpg
| | | |-- approveThreadsSmall.jpg
| | | |-- change.jpg
| | | |-- chat.jpg
| | | |-- colorsheme.jpg
| | | |-- contact.jpg
| | | |-- createMessageSmall.jpg
| | | | |-- databaseid.jpg
|-- chat.php //root chatroom file
|-- chatShareItem.php
|-- chatUpload.php
|-- costAnalysis.php
|-- index.html
|-- inviteUserChat.php
|-- main.php //user root file
|-- settings //ReMoTe settings
  |-- Admin
  |-- `-- setup.conf
  |-- chaz
  |-- `-- setup.conf
  |-- david
  |-- `-- setup.php
  |-- db
  |  |-- `-- ReMoTebkup.sql
  |  |-- `-- remote_algo2005.sql
  |-- ddemelo
  |  |-- `-- setup.php
  |-- jhong
  |  |-- `-- setup.php
  |-- norman
  |  |-- `-- setup.conf
  |-- phil
  |  |-- `-- setup.php
  |-- remoteDemo
  |  |-- `-- setup.php
  |  |-- `-- setup.php -> ddemelo/setup.php
  `-- statistics.php
APPENDIX B

Configuration and Installation Procedures
WAMP and LAMP Configuration and Installation Procedures

WAMP and LAMP is a viable general purpose web server and open source.

WAMP (Windows, Apache, MySQL, and PHP/Perl/Python) is a form of mini-server that can run on any Windows Operating System. LAMP (Linux, Apache HTTP Server, MySQL and PHP/Perl/Python) is one of the most important servers that can run on Linux Operating System.

(1) WAMP Server

(a) Download WAMP:


There are 32bits and 64bits WAMP files (See Figure 82.). Before download WAMP file, you need to check your Operating System (See Figure 83.).
Downloads

WampServer is an open source project, free to use (GPL licence). If you think our work deserves it and you want to help us, you can make a donation with paypal.

**WARNING**: do not try to install WampServer 2 over WAMP5. If WAMP5 is installed on your computer, save your data, uninstall it and delete the WAMP5 directory before installing WampServer 2.

<table>
<thead>
<tr>
<th>DOWNLOAD</th>
<th>WampServer 2.1e (32 bits)</th>
<th>DOWNLOAD</th>
<th>WampServer 2.1d (64 bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(december 27 2010)</td>
<td></td>
<td>(december 27 2010)</td>
</tr>
<tr>
<td>Apache</td>
<td>2.2.17</td>
<td>Apache</td>
<td>2.2.17</td>
</tr>
<tr>
<td>Php</td>
<td>5.3.5</td>
<td>Php</td>
<td>5.3.4</td>
</tr>
<tr>
<td>Mysql</td>
<td>5.5.8</td>
<td>Mysql</td>
<td>5.1.53</td>
</tr>
<tr>
<td>PhpMyadmin</td>
<td>3.2.0.1</td>
<td>PhpMyadmin</td>
<td>3.2.0.1</td>
</tr>
<tr>
<td>SQLBuddy</td>
<td>1.3.2</td>
<td>SQLBuddy</td>
<td>1.3.2</td>
</tr>
<tr>
<td>XDebug</td>
<td>2.1.0-5.3</td>
<td>XDebug</td>
<td>2.1.0-5.3</td>
</tr>
<tr>
<td>webGrind</td>
<td>1.0</td>
<td>webGrind</td>
<td>1.0</td>
</tr>
<tr>
<td>XDC</td>
<td>1.5</td>
<td>XDC</td>
<td>1.5</td>
</tr>
<tr>
<td>taille: 36Mo</td>
<td></td>
<td>taille: 36Mo</td>
<td></td>
</tr>
</tbody>
</table>

Figure 82. Download WAMP
Figure 83. Check System Type.
Welcome to the WampServer 2 Setup Wizard

This will install WampServer 2.1 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

Figure 84. Setup WampServer1.
Figure 85. Setup WampServer2.
Figure 86. Setup WampServer 3.
Figure 87. Setup WampServer 4.
Figure 88. Setup WampServer5.
Completing the WampServer 2 Setup Wizard

Setup has finished installing WampServer 2 on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

Launch WampServer 2 now

Figure 89. Setup WampServer6.
Figure 90. Server test (http://localhost/).
Figure 91. Upload test html file.

Figure 92. Check test file using Web-Browsers.
(2) Installation CentOS

Download and Installation Procedure:

- Download CentOS from
  
  http://isoredirect.centos.org/centos/5isos/i386/

- Insert CentOS CD into CDROM Drive, and boot the system with the CDROM.

- Type linux text from the boot: prompt.

- Select “skip” from Median check.

- Press OK.

- Select English and press OK.

- Select us and press OK.

- Select “Remove all partitions on selected drives and create default layout” and press OK.

- Press Yes.

- Press Yes.

- Press OK.

- Press OK (Boot Loader).

- Press OK.

- Set Boot Loader Password and press OK.

- Press OK.

- Select MBR and press OK.
• Press Yes on Network setup menu.

• Select the following and press OK.
  
  [*] Active on boot
  
  [*] Enable IPv4 support

• Select Manual address configuration.

  IP: 192.168.1.# / NM 255.255.255.0

Press OK.

• Sdfsdf Enter the following entries:

  Gateway: 192.168.1.1
  
  Primary DNS: 192.168.1.2
  
  Secondary DNS: 139.182.2.1

• Hostname setup: select manually and enter the name of the host.

• Set Time Zone: (Example: Los Angeles) Unselect UTC

• Set Root Password

• Package Selection:

  Unselect everything from the menu

  Select the "Custom software Selection"

  and press OK.

• Package Group Selection:

  Unselect everything from the menu press OK.
• Installation to begin: press OK.

(3) Update System

• Update System

  # rpm - import /etc/pki/rpm-gpg/RPM-GPG-KEY*
  # yum -y update

  Reboot the system after update completed.

• Turn off the unnecessary services

  # chkconfig -list | grep 3:on | cut -c1-16 > off

  Edit the file "off" and remove the daemon that you want to keep from off list:

  # vi off

• Delete following list from off list

  Network
  Iptables
  Syslog
  Crond
  Sshd
# for I in $(cat off);do chkconfig $i off;service $i stop;done

- Modify /etc/hosts file
  
  # Do not remove the following line, or various programs
  # that require network functionality will fail.
  127.0.0.1 localhost.localdomain localhost :
  :: localhost6.localhostmain6 localhost6

  #echo "$(

- Time Configuration
  
  # yum -y install ntp
  # ntpdate pool.ntp.org
  # service ntpd start
  # chkconfig ntpd on
  # hwclock -w
  # Date:hwclock -r

- Secure Shell (SSH) configuration
  
  Edit /etc/ssh/sshd_config
Protocol2
PermitRootLogin no
MaxAuthTries3
PermitEmptyPasswords no
X11Forwarding no
PermitTunnel no

# service sshd start

• Create an administrative sudo user
  # useradd -G wheel -c "Full Name" loginname
  # passwd loginname
  # visudo

• Edit: Remove #from the line below
  %wheel ALL=(ALL) NOPASSWD: ALL

• Install and configure Denyhosts

Download DenyHosts
  # wget
  http://downloads.sourceforge.net/denyhosts/DenyHosts-2.6-
python2.4.noarch.rpm?modtime=1165494114&big_mirror
r=0.

- Installation:
  
  # rpm -Uvh DenyHOsts-2.6-python2.4.noarch.rpm

- Configuration:
  
  # cp /usr/share/denyhosts/denyhosts.cfg-dist/usr/share/denyhosts/denyhosts.cfg
  # vi /usr/share/denyhosts/denyhosts.cfg
  
  ####### THESE SETTINGS ARE REQUIRED #######
  
  SECURE_LOG = /var/log/secure
  HOSTS_DENY = /etc/hosts.deny
  PURGE_DENY =
  BLOCK_SERVICE = sshd
  DENY_THRESHOLD_INVALID = 5
  DENY_THRESHOLD_VALID = 10
  DENY_THRESHOLD_ROOT = 1
  DENY_THRESHOLD_RESTRICTED = 1
  WORK_DIR = /usr/share/denyhosts/data
  SUSPICIOUS_LOGIN_REPORT_ALLOWED_HOSTS=YES
  HOSTNAME_LOOKUP=YES

149
LOCK_FILE = /var/lock/subsys/denyhosts

4) LAMP Server

LAMP (Linux, Apache, MySQL, PHP) Server

Installation:

# yum -y install mysql mysql-server php php-gd php-mbstring php-mysql php-pear-Mail mod_auth_mysql
httpd mod_ssl openssl crypto-utils

- Configuration:

  # rm -vf /etc/pki/tls/private/*.key

  # rm -vf /etc/pki/tls/certs/*.crt

Generate CA Certificate and Private key:

  # genkey -days 2048 remote.ias.csusb.edu

  Next -> next -> No ->

  US
  California
  San Bernardino
  Remote.cse.csusb
  IT
  www.remote.ias.csusb.edu
Next -> Encrypt the private key -> Next ->
Passphrase(twice) -> Next

• Edit: /etc/httpd/conf.d/ssl.conf
  SSLCertificateFile
  /etc/pki/tls/certs/remote.ias.csusb.edu
  SSLCertificateKeyFile
  /etc/pki/tls/private/www.remote.ias.csusb.edu.key
  DocumentRoot /var/www/html
  ServerName www.remote.ias.csusb.edu:443

• Edit /etc/hosts: (or Add DNS entry for
  www.remote.ias.csusb.edu)
  # echo "IP (**.***.***.***)
  www.remote.ias.csusb.edu" >>/etc/hosts

• Check php mysql configuration:
  #echo "<?php phpinfo(); ?>">>
  /var/www/html/test.php
  #echo "Welcome to Virtuallab">>
  /var/www/html/index.php
  #service httpd restart
• Get rid of passphrase from httpd startup

```bash
# cd /etc/pki/tls/private
# cp www.remote.ias.csusb.edu.key
www.remote.ias.csusb.edu.key.org
# openssl rsa -in www.remote.ias.csusb.edu.key.org
-out www.remote.csusb.edu.key
# chmod 400 www.remote.ias.csusb.edu.key
# service httpd restart
# chkconfig httpd on
# rm -vf www.remote.ias.csusb.edu.key.org
```

• Web Server Testing

```bash
# echo "IP (**.**.**.**) www.remote.ias.csusb.edu" >> /etc/hosts
Browse your http and https sites:
http://www.remote.ias.csusb.edu and
https://www.remote.ias.csusb.edu
Browse Your http and https sites:
http://www.remote.ias.csusb.edu/test.php
```
REFERENCES


http://www.w3schools.com/w3c/default.asp


http://www.wbtshowcase.com/wbt/web.nsf/pages/presentingtech.html#07
