



# **Bachelor of Information Technology**

## **(Hons)**

### **BIT305**

### **Final Year Project II**

## **Human Resource Management System**

Adam Waheed (E1100075)  
Mohamed Simadh (E1100097)  
Mariyam Expanded (E1100092)

---

**Project Period: 10th Jan2013 – 24<sup>th</sup> April 2013**

## Abstract

This project is done as a final year project for Bachelors of Information Technology (Hons) program offered by Help University, Malaysia.

The project is undertaken to plan, design and develop a Human Resource Management system, named “TinyHRMS” for Maldives Stock Exchange.

Human Resource Management system provides the information regarding the employees in the company. The system facilitates good interaction / communication facilities between the employees and HR administration. The web pages about an employee are created dynamically based on the user id and password and links are provided to web pages containing information like employee general details. HRMS also has the facility of viewing a detailed report regarding the employee.

## Declaration

I hereby declare that the report presented here as part of the requirement of BIT305 is original and no parts of this report had been plagiarised from any other resources unless those indicated with proper referencing. This report will be the property of Help University and cannot be distributed in any form without the written consent of Help University.

### Student Names

### Signature

Adam Waheed

Mohamed Simadh

Mariyam Expanded

Date: 15<sup>th</sup> April 2013

## Acknowledgement

This project will not be possible without the help of a number of people. We would like to express our deepest gratitude to our lecturer Mr. Ibrahim Waheed for his invaluable guidance and assistance, without which the accomplishment of the task would have never been possible. We also thank Mr. Hussain Naushad for devoting much of his time to supervise us in this project.

We are also thankful to the employees of Maldives Stock Exchange, especially, Mr. Abdulla Musthafa for providing us relevant information and necessary clarifications required for the project.

## Table of Contents

<b>ABSTRACT .....</b>	<b>1</b>
<b>DECLARATION.....</b>	<b>2</b>
<b>ACKNOWLEDGEMENT .....</b>	<b>3</b>
<b>LIST OF FIGURES.....</b>	<b>7</b>
<b>LIST OF TABLES.....</b>	<b>11</b>
<b>LIST OF SYMBOLS AND ABBREVIATIONS .....</b>	<b>16</b>
<b>CHAPTER 1: PROPOSAL .....</b>	<b>17</b>
1.1    PROJECT SUMMARY .....	17
1.2    COMPANY BACKGROUND.....	17
1.2.1 <i>Board of Directors</i> .....	18
1.2.2 <i>Organization Chart of Maldives Stock Exchange</i> .....	18
1.3.    ISSUES WITH CURRENT SYSTEM.....	19
1.4.    BENEFITS AND CONSTRAINTS OF PROPOSED SYSTEM .....	20
1.5.    PROJECT DESCRIPTION.....	21
1.5.1. <i>Overview Structure of TinyHRMS</i> .....	21
1.5.2. <i>Important Modules of the System</i> .....	22
1.5.3. <i>Main Users of the System</i> .....	22
1.5.4. <i>High level Use case Diagram</i> .....	22
1.6.    PROJECT AIMS.....	23
1.7.    PROJECT OBJECTIVES.....	23
1.8.    PROJECT SCOPE.....	24
1.9.    SOFTWARE AND HARDWARE REQUIREMENTS .....	24
1.10.    DEVELOPMENT METHODOLOGY.....	26
1.10.1. <i>Waterfall Model</i> .....	26
1.11.    PROJECT PLAN .....	28
1.11.1 <i>Gantt Chart</i> .....	28
<b>CHAPTER 2: LITERATURE REVIEW .....</b>	<b>29</b>
2.1.    INTRODUCTION .....	29
2.2.    IMPORTANCE OF HRMS .....	29
2.3.    SIMILAR SYSTEM THAT EXIST .....	30
2.3.1. <i>Sage Abra</i> .....	30
2.3.2. <i>Fi-es HRMS</i> .....	30
2.4.    SYSTEM DEVELOPMENT METHODOLOGY .....	30
2.4.1. <i>Comparative Study on Development Methodologies</i> .....	31
2.5.    PATTERN-ORIENTED SOFTWARE ARCHITECTURE .....	33
2.5.1. <i>Presentation-Abstraction-Control (PAC)</i> .....	33
2.5.2. <i>Model View Controller (MVC)</i> .....	34
2.6.    TOOLS FOR DYNAMIC WEBSITES .....	35
2.6.1. <i>Hypertext Pre-processor (PHP)</i> .....	35
2.6.2. <i>Active Server Pages (ASP)</i> .....	36
2.7.    CONCLUSION.....	36
<b>CHAPTER 3: REQUIREMENTS ANALYSIS .....</b>	<b>38</b>
3.1.    INTRODUCTION .....	38
3.2.    REQUIREMENTS GATHERING.....	38
3.2.1 <i>Interviewing</i> .....	38
3.2.2 <i>Use cases</i> .....	38
3.2.3 <i>Observation</i> .....	39
3.3.    REQUIREMENTS SUMMARY .....	39
3.3.1. <i>Modules of HRMS</i> .....	39

3.3.2. <i>Main Users of the System</i> .....	40
3.3.3. <i>Use Case Diagram</i> .....	43
3.3.4. <i>High-level Use Cases</i> .....	44
3.3.5. <i>Expanded Use Cases</i> .....	55
<b>CHAPTER 4: SYSTEM DESIGN.....</b>	<b>79</b>
4.1. INTRODUCTION .....	79
4.2. DATABASE DESIGN .....	79
4.2.1. <i>System Data Dictionary</i> .....	79
4.2.2. <i>Entity Relational (ER) Diagram</i> .....	84
4.3. LOGICAL DESIGN .....	85
4.3.1. <i>Class Design Diagram</i> .....	85
4.3.2. <i>Web-based Structural Design</i> .....	86
4.4. DYNAMIC DESIGN .....	87
4.4.1. <i>Sequence Diagrams</i> .....	87
4.5. INTERFACE DESIGN .....	112
<b>CHAPTER 5: IMPLEMENTATION &amp; TESTING .....</b>	<b>133</b>
5.1 INTRODUCTION .....	133
5.2 IMPLEMENTATION METHODOLOGY .....	133
5.3 IMPLEMENTATION .....	134
5.3.1 <i>Work Items List</i> .....	134
5.3.2 <i>Work Items List</i> .....	136
5.3.3 <i>Work Items Assignment</i> .....	138
5.4 ISSUES .....	141
5.5 EVALUATION CRITERIA .....	141
5.6 ASSESSMENT .....	141
5.6.1 <i>Assessment Against Objectives</i> .....	141
5.6.2 <i>Assessment against Schedule</i> .....	143
5.6.3 <i>Assessment against Evaluation Criteria Test Results</i> .....	144
5.6.4 <i>Other concerns and deviations</i> .....	144
5.7 TESTING .....	145
5.7.1 <i>Testing Objectives</i> .....	145
5.7.2 <i>Testing Objectives</i> .....	145
5.8 TESTING PHASE CONCLUSION .....	146
<b>CHAPTER 6: PROJECT REVIEW .....</b>	<b>147</b>
6.1. INTRODUCTION .....	147
6.2. AIMS AND OBJECTIVES .....	147
6.3. WORK ITEMS.....	148
6.4. SUCCESSES.....	149
6.4.1. <i>Team Composition and Communication</i> .....	149
6.4.2. <i>Project Pace, Momentum and Support</i> .....	149
6.5. SHORTCOMINGS .....	150
6.5.1. <i>Communication between Stakeholders</i> .....	150
6.5.2. <i>External Events</i> .....	150
6.6. DISCUSSION .....	151
6.6.1. <i>Development of the Project</i> .....	151
6.6.2. <i>Team</i> .....	151
6.7. LEARNING OUTCOMES .....	151
6.7.1. <i>People Skills</i> .....	151
6.7.2. <i>Project Management Skills</i> .....	151
6.7.3. <i>Technical /Computing Skills</i> .....	152
6.8. PROJECT MANAGEMENT .....	152
6.9. COST EFFECTIVENESS .....	153
<b>CHAPTER 7: CONCLUSION .....</b>	<b>154</b>

7.1.	INTRODUCTION .....	154
7.2.	FURTHER IMPROVEMENT .....	154
7.3.	FUTURE WORK.....	155
7.4.	SUMMARY .....	155
<b>REFERENCES .....</b>		<b>156</b>
<b>APPENDIX A: .....</b>		<b>157</b>
<b>MONTHLY PROGRESS REPORTS.....</b>		<b>157</b>
<b>APPENDIX B: .....</b>		<b>161</b>
<b>GANTT CHART .....</b>		<b>161</b>
<b>APPENDIX C: .....</b>		<b>162</b>
<b>HRMS SURVEY REPORT.....</b>		<b>162</b>
<b>APPENDIX D: .....</b>		<b>163</b>
<b>TEST CASES.....</b>		<b>163</b>
	UNIT TESTING .....	163
	ACCEPTANCE TEST.....	213
<b>APPENDIX E: .....</b>		<b>215</b>
<b>MINUTES OF MEETINGS .....</b>		<b>215</b>
<b>APPENDIX F:.....</b>		<b>227</b>
<b>TEAM MEMBERS' CONTRIBUTIONS .....</b>		<b>227</b>
<b>APPENDIX G:.....</b>		<b>228</b>
<b>USER MANUAL .....</b>		<b>228</b>

## List of Figures

FIGURE 1:LOW-LEVEL ORGANIZATION CHART OF MSE .....	18
FIGURE 2 OVERVIEW OF TINYHRMS .....	21
FIGURE 3: HIGH-LEVEL USE CASE DIAGRAM.....	22
FIGURE 4 WATERFALL MODEL.....	27
FIGURE 5 USE CASE DIAGRAM OF TINYHRMS .....	43
FIGURE 6 ENTITY RELATIONAL (ER) DIAGRAM OF TINYHRMS .....	84
FIGURE 7 CLASS DESIGN DIAGRAM .....	85
FIGURE 8 WEB-BASED STRUCTURAL DESIGN.....	86
FIGURE 9 SEQUENCE DIAGRAM CREATE USER.....	87
FIGURE 10 SEQUENCE DIAGRAM LOG-IN .....	88
FIGURE 11 SEQUENCE DIAGRAM FORGOT PASSWORD .....	88
FIGURE 12 SEQUENCE DIAGRAM EDIT PROFILE .....	89
FIGURE 13 SEQUENCE DIAGRAM DELETE USER .....	89
FIGURE 14 SEQUENCE DIAGRAM UPDATE USER .....	90
FIGURE 15 SEQUENCE DIAGRAM ASSIGN ROLE TO THE USER.....	90
FIGURE 16 SEQUENCE DIAGRAM WITHDRAW USER ROLE .....	91
FIGURE 17 SEQUENCE DIAGRAM CREATE NEW DEPARTMENT.....	91
FIGURE 18 SEQUENCE DIAGRAM UPDATE DEPARTMENT INFORMATION .....	92
FIGURE 19 SEQUENCE DIAGRAM DELETE DEPARTMENT.....	92
FIGURE 20 SEQUENCE DIAGRAM CREATE NEW EMPLOYEE .....	93
FIGURE 21 SEQUENCE DIAGRAM UPDATE EMPLOYEE INFORMATION .....	93
FIGURE 22 SEQUENCE DIAGRAM DELETE EMPLOYEE .....	94
FIGURE 23 SEQUENCE DIAGRAM CREATE ALLOWANCE.....	94
FIGURE 24 SEQUENCE DIAGRAM UPDATE ALLOWANCE .....	95
FIGURE 25 SEQUENCE DIAGRAM DELETE ALLOWANCE.....	95
FIGURE 26 SEQUENCE DIAGRAM ASSIGN ALLOWANCE .....	96
FIGURE 27 SEQUENCE DIAGRAM CREATE NEW LEAVE.....	96
FIGURE 28 SEQUENCE DIAGRAM DELETE LEAVE.....	97
FIGURE 29 SEQUENCE DIAGRAM UPDATE LEAVE INFORMATION.....	97
FIGURE 30 SEQUENCE DIAGRAM APPROVE LEAVE .....	98
FIGURE 31 SEQUENCE DIAGRAM LEAVE REQUEST .....	98
FIGURE 32 SEQUENCE DIAGRAM AMEND LEAVE REQUEST .....	99
FIGURE 33 SEQUENCE DIAGRAM CHECK IN.....	99
FIGURE 34 SEQUENCE DIAGRAM VIEW ATTENDANCE .....	100
FIGURE 35 SEQUENCE DIAGRAM CREATE DESIGNATION .....	100
FIGURE 36 SEQUENCE DIAGRAM UPDATE DESIGNATION .....	101

FIGURE 37 SEQUENCE DIAGRAM DELETE DESIGNATION.....	101
FIGURE 38 SEQUENCE DIAGRAM PAYROLL PROCESS.....	102
FIGURE 39 SEQUENCE DIAGRAM UPDATE ATTENDANCE .....	102
FIGURE 40 SEQUENCE DIAGRAM ASSIGN EMPLOYEE TO THE DEPARTMENT .....	103
FIGURE 41 SEQUENCE DIAGRAM WITHDRAW LEAVE.....	103
FIGURE 42 SEQUENCE DIAGRAM UPDATE DEPARTMENT INFORMATION .....	104
FIGURE 43 SEQUENCE DIAGRAM UPDATE CALENDAR EVENT.....	104
FIGURE 44 SEQUENCE DIAGRAM DELETE CALENDAR EVENT .....	105
FIGURE 45 SEQUENCE DIAGRAM CREATE USER GROUP .....	106
FIGURE 46 SEQUENCE DIAGRAM DELETE USER GROUP .....	107
FIGURE 47 SEQUENCE DIAGRAM CREATE SHIFT .....	108
FIGURE 48 SEQUENCE DIAGRAM DELETE SHIFT .....	108
FIGURE 49 SEQUENCE DIAGRAM UPDATE SHIFT.....	109
FIGURE 50 SEQUENCE DIAGRAM CREATE SPECIAL TASK.....	109
FIGURE 51 SEQUENCE DIAGRAM UPDATE SPECIAL TASK.....	110
FIGURE 52 SEQUENCE DIAGRAMS DELETE TASK.....	110
FIGURE 53 SEQUENCE DIAGRAM ASSIGN EMPLOYEE TO SPECIAL TASK EMPLOYEE .....	111
FIGURE 54 SEQUENCE DIAGRAMS ASSIGN EMPLOYEE TO SPECIAL TASK EMPLOYEE .....	111
FIGURE 55 DASHBOARD MOCK FLOW .....	112
FIGURE 56 LOGIN PAGE.....	113
FIGURE 57 DASH BOARD.....	113
FIGURE 58 USER LIST.....	114
FIGURE 59 CREATE USER .....	115
FIGURE 60 VIEW USER GROUP .....	115
FIGURE 61 CREATE USER GROUP.....	116
FIGURE 62 USER GROUP UPDATE.....	116
FIGURE 63 VIEW ALLOWANCE .....	117
FIGURE 64 CREATE ALLOWANCE.....	117
FIGURE 65 ALLOWANCE UPDATE.....	118
FIGURE 66 ASSIGN ALLOWANCE TO EMPLOYEE.....	118
FIGURE 67 ASSIGN ALLOWANCE TO EMPLOYEE.....	119
FIGURE 68 CREATE CALENDAR EVENT.....	119
FIGURE 69 TIME SHEET LIST .....	120
FIGURE 70 TIME SHEET READING .....	120
FIGURE 71 UPDATE ATTENDANCE FILE.....	121
FIGURE 72 VIEW LEAVE LIST .....	121
FIGURE 73 CREATE LEAVE.....	122
FIGURE 74 LEAVE REQUEST BY STAFF .....	122
FIGURE 75 LEAVE REQUEST BY ADMIN .....	123

9 | Human Resource Management system  
List of Figures

FIGURE 76 REQUESTED LEAVE.....	123
FIGURE 77 AMEND LEAVE REQUEST.....	124
FIGURE 78 VIEW DEPARTMENT.....	124
FIGURE 79 UPDATE DEPARTMENT .....	125
FIGURE 80 ASSIGN EMPLOYEE TO DEPARTMENT.....	125
FIGURE 81 VIEW DESIGNATION.....	126
FIGURE 82 CREATE DESIGNATION.....	126
FIGURE 83 UPDATE DESIGNATION .....	127
FIGURE 84 EMPLOYEE LIST .....	127
FIGURE 85 CREATE EMPLOYEE .....	128
FIGURE 86 VIEW SHIFT .....	128
FIGURE 87 CREATE SHIFT .....	129
FIGURE 88 UPDATE SHIFT .....	129
FIGURE 89 ADD EMPLOYEE TO SHIFT .....	130
FIGURE 90 TASK LIST.....	130
FIGURE 91 UPDATE TASK.....	131
FIGURE 92 ASSIGN EMPLOYEE TO TASK.....	131
FIGURE 93 TIME SHEET .....	132
FIGURE 94 PARALLEL IMPLEMENTATION .....	133
FIGURE 95: PROJECT BURN-DOWN CHART .....	149
FIGURE 96 GANTT CHART .....	161
FIGURE 97 LOGIN PAGE.....	228
FIGURE 98 DASH BOARD.....	228
FIGURE 99 USER LIST.....	229
FIGURE 100 CREATE USER .....	229
FIGURE 101 VIEW USER GROUP.....	230
FIGURE 102 CREATE USER GROUP.....	230
FIGURE 103 USER GROUP UPDATE.....	231
FIGURE 104 VIEW ALLOWANCE.....	232
FIGURE 105 CREATE ALLOWANCE .....	232
FIGURE 106 ALLOWANCE UPDATE.....	233
FIGURE 107 ASSIGN ALLOWANCE TO EMPLOYEE.....	233
FIGURE 108 ASSIGN ALLOWANCE TO EMPLOYEE.....	234
FIGURE 109 CREATE CALENDAR EVENT.....	234
FIGURE 110 TIME SHEET LIST .....	235
FIGURE 111 TIME SHEET READING .....	235
FIGURE 112 TIME SHEET .....	236
FIGURE 113 UPDATE ATTENDANCE FILE.....	237
FIGURE 114 VIEW LEAVE LIST .....	237

FIGURE 115 CREATE LEAVE.....	238
FIGURE 116 LEAVE REQUEST BY STAFF .....	238
FIGURE 117 LEAVE REQUEST BY ADMIN .....	239
FIGURE 118 REQUESTED LEAVE.....	239
FIGURE 119 AMEND LEAVE REQUEST.....	240
FIGURE 120 VIEW DEPARTMENT.....	240
FIGURE 121 UPDATE DEPARTMENT.....	241
FIGURE 122 ASSIGN EMPLOYEE TO DEPARTMENT .....	241
FIGURE 123 VIEW DESIGNATION.....	242
FIGURE 124 CREATE DESIGNATION.....	242
FIGURE 125 UPDATE DESIGNATION .....	243
FIGURE 126 EMPLOYEE LIST .....	243
FIGURE 127 CREATE EMPLOYEE .....	244
FIGURE 128 VIEW SHIFT .....	244
FIGURE 129 CREATE SHIFT .....	245
FIGURE 130 UPDATE SHIFT .....	245
FIGURE 131 ADD EMPLOYEE TO SHIFT.....	246
FIGURE 132 TASK LIST.....	246
FIGURE 133 UPDATE TASK.....	247
FIGURE 134 ASSIGN EMPLOYEE TO TASK.....	247

## List of Tables

TABLE 1: PROBLEM OF MANAGING EMPLOYEES IN A MANUAL SYSTEM.....	19
TABLE 2: PROBLEM OF MANUAL PAYROLL SYSTEM .....	19
TABLE 3: PROBLEM OF TIME CONSUMPTION IN GENERATING REPORTS.....	19
TABLE 4: PROBLEM OF INACCURATE EMPLOYEE DATA .....	20
TABLE 5: PROBLEM OF REDUNDANT DATA.....	20
TABLE 6: PROBLEM OF INSECURE EMPLOYEE INFORMATION .....	20
TABLE 7: SOFTWARE AND HARDWARE REQUIREMENTS OF THE SYSTEM .....	25
TABLE 8: HIGH-LEVEL USE CASE 1 - CREATE USER.....	44
TABLE 9: HIGH-LEVEL USE CASE 2- LOG-IN .....	44
TABLE 10: HIGH-LEVEL USE CASE 3 - FORGOT PASSWORD .....	44
TABLE 11: HIGH-LEVEL USE CASE 4 - EDIT PROFILE .....	44
TABLE 12: HIGH-LEVEL USE CASE 5 - DELETE USER .....	45
TABLE 13: HIGH-LEVEL USE CASE 6 - UPDATE USER .....	45
TABLE 14: HIGH-LEVEL USE CASE 7 - ASSIGN USER ROLE TO USER GROUP .....	45
TABLE 15: HIGH-LEVEL USE CASE 8 - WITHDRAW USER GROUP ROLE .....	45
TABLE 16: HIGH-LEVEL USE CASE 9 - CREATE NEW DEPARTMENT .....	45
TABLE 17: HIGH-LEVEL USE CASE 10 - UPDATE DEPARTMENT INFORMATION .....	46
TABLE 18: HIGH-LEVEL USE CASE 11 - DELETE DEPARTMENT .....	46
TABLE 19: HIGH-LEVEL USE CASE 12 - CREATE NEW EMPLOYEE.....	46
TABLE 20: HIGH-LEVEL USE CASE 13 - UPDATE EMPLOYEE INFORMATION .....	46
TABLE 21: HIGH-LEVEL USE CASE 14 - DELETE EMPLOYEE .....	47
TABLE 22: HIGH-LEVEL USE CASE 15 - CREATE ALLOWANCE.....	47
TABLE 23: HIGH-LEVEL USE CASE 16 - UPDATE ALLOWANCE.....	47
TABLE 24: HIGH-LEVEL USE CASE 17 - DELETE ALLOWANCE .....	47
TABLE 25: HIGH-LEVEL USE CASE 18 - ASSIGN ALLOWANCE .....	48
TABLE 26: HIGH-LEVEL USE CASE 19 - CREATE NEW LEAVE .....	48
TABLE 27: HIGH-LEVEL USE CASE 20 - DELETE LEAVE .....	48
TABLE 28: HIGH-LEVEL USE CASE 21 - UPDATE LEAVE INFORMATION.....	48
TABLE 29: HIGH-LEVEL USE CASE 22 - APPROVE/DECLINE LEAVE .....	48
TABLE 30: HIGH-LEVEL USE CASE 23 - REQUEST LEAVE.....	49
TABLE 31: HIGH-LEVEL USE CASE 24 - AMEND LEAVE REQUEST.....	49
TABLE 32: HIGH-LEVEL USE CASE 25 - CHECK-IN.....	49
TABLE 33: HIGH-LEVEL USE CASE 26 - REQUEST FOR UPDATE ATTENDANCE.....	49
TABLE 34: HIGH-LEVEL USE CASE 27 - VIEW ATTENDANCE .....	50
TABLE 35: HIGH-LEVEL USE CASE 28 - CREATE DESIGNATION .....	50
TABLE 36: HIGH-LEVEL USE CASE 29 - UPDATE DESIGNATION .....	50
TABLE 37: HIGH-LEVEL USE CASE 30 - DELETE DESIGNATION.....	50

TABLE 38: HIGH-LEVEL USE CASE 31- PAYROLL PROCESS.....	50
TABLE 39: HIGH-LEVEL USE CASE 32 - UPDATE ATTENDANCE .....	51
TABLE 40: HIGH-LEVEL USE CASE 33 - ASSIGN EMPLOYEE TO THE DEPARTMENT .....	51
TABLE 41: HIGH-LEVEL USE CASE 34 – WITHDRAW LEAVE .....	51
TABLE 42: HIGH-LEVEL USE CASE 35 - ASSIGN LEAVE .....	51
TABLE 43: HIGH-LEVEL USE CASE 36 - UPDATE DEPARTMENT INFORMATION .....	52
TABLE 44: HIGH-LEVEL USE CASE 37 - UPDATE CALENDAR EVENT.....	52
TABLE 45: HIGH-LEVEL USE CASE 38 - DELETE CALENDAR EVENT .....	52
TABLE 46: HIGH-LEVEL USE CASE 39 - CREATE USER GROUP .....	52
TABLE 47: HIGH-LEVEL USE CASE 40 - DELETE USER GROUP .....	52
TABLE 48: HIGH-LEVEL USE CASE 41 – CREATE SHIFT .....	53
TABLE 49: HIGH-LEVEL USE CASE 42 - DELETE SHIFT .....	53
TABLE 50: HIGH-LEVEL USE CASE 43 - UPDATE SHIFT .....	53
TABLE 51: HIGH-LEVEL USE CASE 44 - CREATE SPECIAL TASK.....	53
TABLE 52: HIGH-LEVEL USE CASE 45 - UPDATE SPECIAL TASK.....	53
TABLE 53: HIGH-LEVEL USE CASE 46 - DELETE TASK.....	54
TABLE 54: HIGH-LEVEL USE CASE 47 - ASSIGN EMPLOYEE TO SPECIAL TASK EMPLOYEE .....	54
TABLE 55: HIGH-LEVEL USE CASE 48 - WITHDRAW EMPLOYEE .....	54
TABLE 56: HIGH-LEVEL USE CASE 49 - CREATE SPECIAL TASK .....	54
TABLE 57: EXPANDED USE CASE 1- CREATE USER.....	55
TABLE 58: EXPANDED USE CASE 2- LOG IN .....	55
TABLE 59: EXPANDED USE CASES 4 - EDIT PROFILE.....	56
TABLE 60: EXPANDED USE CASES 5 - DELETE USER.....	57
TABLE 61: EXPANDED USE CASES 7 - ASSIGN ROLE TO THE USER .....	58
TABLE 62: EXPANDED USE CASES 8 - WITHDRAW USER ROLE.....	58
TABLE 63: EXPANDED USE CASES 9 - CREATE NEW DEPARTMENT .....	59
TABLE 64: EXPANDED USE CASES 10 - UPDATE DEPARTMENT INFORMATION .....	59
TABLE 65: EXPANDED USE CASES 11 - DELETE DEPARTMENT .....	60
TABLE 66: EXPANDED USE CASES 12 - CREATE NEW EMPLOYEE.....	60
TABLE 67: EXPANDED USE CASE 13 - UPDATE EMPLOYEE INFORMATION.....	61
TABLE 68 : EXPANDED USE CASE 14 - DELETE EMPLOYEE.....	61
TABLE 69: EXPANDED USE CASE 15 - CREATE ALLOWANCE.....	62
TABLE 70: EXPANDED USE CASE 16 - UPDATE ALLOWANCE.....	62
TABLE 71: EXPANDED USE CASE 17 - DELETE ALLOWANCE .....	63
TABLE 72: EXPANDED USE CASE 18 - ASSIGN ALLOWANCE.....	64
TABLE 73: EXPANDED USE CASE 19 - CREATE NEW LEAVE .....	64
TABLE 74: EXPANDED USE CASE 20 - DELETE LEAVE .....	65
TABLE 75: EXPANDED USE CASE 21 - UPDATE LEAVE INFORMATION .....	65
TABLE 76: EXPANDED USE CASE 22 - APPROVE/DECLINE LEAVE .....	66

TABLE 77 : EXPANDED USE CASES 23 - REQUESTS FOR LEAVE.....	66
TABLE 78: EXPANDED USE CASE 24 - AMEND LEAVE REQUEST.....	67
TABLE 79: EXPANDED USE CASE 25 - CHECK-IN .....	67
TABLE 80: EXPANDED USE CASE 26 - UPDATE ATTENDANCE.....	68
TABLE 81: EXPANDED USE CASE 27 - VIEW ATTENDANCE.....	68
TABLE 82: EXPANDED USE CASE 28 - CREATE DESIGNATION .....	68
TABLE 83: EXPANDED USE CASE 29 - UPDATE DESIGNATION .....	69
TABLE 84: EXPANDED USE CASE 30 - DELETE DESIGNATION .....	69
TABLE 85: EXPANDED USE CASE 31 - PAYROLL PROCESS .....	70
TABLE 86: EXPANDED USE CASE 32 - UPDATE ATTENDANCE.....	70
TABLE 87: EXPANDED USE CASE 33 - ASSIGN EMPLOYEE TO THE DEPARTMENT .....	71
TABLE 88: EXPANDED USE CASE 34 - WITHDRAW LEAVE .....	71
TABLE 89: EXPANDED USE CASE 35 - ASSIGN LEAVE.....	72
TABLE 90: EXPANDED USE CASE 36 - UPDATE DEPARTMENT INFORMATION .....	72
TABLE 91: EXPANDED USE CASE 37 - UPDATE CALENDAR EVENT .....	73
TABLE 92: EXPANDED USE CASE 38 - DELETE CALENDAR EVENT .....	73
TABLE 93: EXPANDED USE CASE 39 - CREATE USER GROUP .....	74
TABLE 94: EXPANDED USE CASE 40 - DELETE USER GROUP.....	74
TABLE 95: EXPANDED USE CASE 41 - CREATE SHIFT .....	75
TABLE 96: EXPANDED USE CASE 42 - DELETE SHIFT.....	75
TABLE 97: EXPANDED USE CASE 43 - UPDATE SHIFT .....	76
TABLE 98: EXPANDED USE CASE 44 - UPDATE SPECIAL TASK .....	76
TABLE 99: EXPANDED USE CASE 45 - DELETE TASK.....	77
TABLE 100: EXPANDED USE CASE 46 - ASSIGN EMPLOYEE TO SPECIAL TASK EMPLOYEE.....	77
TABLE 101: EXPANDED USE CASE 47 - WITHDRAW EMPLOYEE .....	78
TABLE 102: EXPANDED USE CASE 48 - CREATE SPECIAL TASK .....	78
TABLE 103: TABLE STRUCTURE FOR TABLE ACCESS GROUPS.....	79
TABLE 104TABLE STRUCTURE FOR TABLE ALLOWANCE .....	80
TABLE 105TABLE STRUCTURE FOR TABLE CALENDAR .....	80
TABLE 106TABLE STRUCTURE FOR TABLE DEPARTMENT .....	80
TABLE 107TABLE STRUCTURE FOR TABLE DESIGNATION .....	80
TABLE 108TABLE STRUCTURE FOR TABLE EMPLOYEE.....	80
TABLE 109TABLE STRUCTURE FOR TABLE EMPLOYEEALLOWANCE .....	81
TABLE 110TABLE STRUCTURE FOR TABLE EMPLOYEE LEAVE .....	81
TABLE 111TABLE STRUCTURE FOR TABLE LEAVLIST .....	81
TABLE 112TABLE STRUCTURE FOR TABLE RESOURCES .....	81
TABLE 113TABLE STRUCTURE FOR TABLE SHIFT.....	81
TABLE 114TABLE STRUCTURE FOR TABLE SPECIAL TASK.....	82
TABLE 115TABLE STRUCTURE FOR TABLE SPECIAL TASK EMP .....	82

TABLE 116 TABLE STRUCTURE FOR TABLE TIMESHEET .....	82
TABLE 117 TABLE STRUCTURE FOR TABLE USERGROUP.....	82
TABLE 118 TABLE STRUCTURE FOR TABLE USERS.....	82
TABLE 119 TABLE STRUCTURE FOR TABLE WAGES .....	83
TABLE 120 TEST CASE - REQUEST LEAVE.....	163
TABLE 121 TEST CASE - VIEW LEAVE .....	164
TABLE 122 TEST CASE - UPDATE LEAVE.....	165
TABLE 123 TEST CASE - DELETE LEAVE .....	166
TABLE 124 TEST CASE - CREATE ALLOWANCE.....	167
TABLE 125 TEST CASE - VIEW ALLOWANCE .....	168
TABLE 126 TEST CASE - UPDATE ALLOWANCE.....	169
TABLE 127 TEST CASE - DELETE ALLOWANCE .....	170
TABLE 128 TEST CASE - SEARCH ALLOWANCE.....	171
TABLE 129 TEST CASE - ASSIGN ALLOWANCE.....	172
TABLE 130 TEST CASE - WITHDRAW ALLOWANCE FROM EMPLOYEE.....	173
TABLE 131 TEST CASE - CREATE CALENDAR .....	174
TABLE 132 TEST CASE - VIEW CALENDAR .....	175
TABLE 133 TEST CASE - UPDATE CALENDAR .....	176
TABLE 134 TEST CASE - DELETE CALENDAR.....	177
TABLE 135 TEST CASE - SEARCH CALENDAR .....	178
TABLE 136 TEST CASE - CREATE DESIGNATION .....	179
TABLE 137 TEST CASE - VIEW DESIGNATION .....	180
TABLE 138 TEST CASE - UPDATE DESIGNATION.....	181
TABLE 139 TEST CASE - DELETE DESIGNATION.....	182
TABLE 140 TEST CASE - SEARCH DESIGNATION .....	183
TABLE 141 TEST CASE - ASSIGN EMPLOYEE TO DESIGNATION .....	184
TABLE 142 TEST CASE - WITHDRAW EMPLOYEE FROM DESIGNATION .....	185
TABLE 143 TEST CASE - CREATE DEPARTMENT .....	186
TABLE 144 TEST CASE - VIEW DEPARTMENT .....	187
TABLE 145 TEST CASE - UPDATE DEPARTMENT .....	188
TABLE 146 TEST CASE - DELETE DEPARTMENT .....	189
TABLE 147 TEST CASE - SEARCH DEPARTMENT .....	190
TABLE 148 TEST CASE - ASSIGN EMPLOYEE TO DEPARTMENT .....	191
TABLE 149 TEST CASE - WITHDRAW EMPLOYEE FROM DEPARTMENT .....	192
TABLE 150 TEST CASE - CREATE USER.....	193
TABLE 151 TEST CASE - VIEW USER.....	194
TABLE 152 TEST CASE - UPDATE USER.....	195
TABLE 153 TEST CASE - DELETE USER .....	196
TABLE 154 TEST CASE - SEARCH USER.....	197

TABLE 155 TEST CASE - LOGIN USER.....	198
TABLE 156 TEST CASE - LOGOUT USER.....	199
TABLE 157 TEST CASE - CREATE USER GROUP.....	200
TABLE 158 TEST CASE - VIEW USER GROUP.....	201
TABLE 159 TEST CASE - UPDATE USER GROUP .....	202
TABLE 160 TEST CASE - DELETE USER GROUP.....	203
TABLE 161 TEST CASE - SEARCH USER GROUP .....	204
TABLE 162 TEST CASE - ASSIGN USER ROLE.....	205
TABLE 163 TEST CASE - WITHDRAW USER ROLE.....	206
TABLE 164 TEST CASE - CREATE LEAVE TYPE .....	207
TABLE 165 TEST CASE - VIEW LEAVE TYPE .....	208
TABLE 166 TEST CASE - UPDATE LEAVE TYPE .....	209
TABLE 167 TEST CASE - DELETE LEAVE TYPE.....	210
TABLE 168 TEST CASE - SEARCH LEAVE TYPE .....	211
TABLE 169 TEST CASE - APPROVE LEAVE .....	212
TABLE 170 TEST CASE - ACCEPTANCE TEST.....	214

## List of Symbols and Abbreviations

**TinyHRMS:** Name of the product

**HRMS:** Human Resource Management System

**HR Staff:** Human Resource Department Staff

**MVC:** Model View Control

**MD5:** Message-Digest Algorithm that produces a 128-bit (16-byte) hash value.

**MSE:** Maldives Stock Exchange (client)

**CMDA:** Capital Market Development Authority

**MSA:** Maldives Securities Act

**SEO:** Search Engine Optimization

# CHAPTER 1: PROPOSAL

## 1.1 Project Summary

Human Resource Management System (HRMS) designed and developed for Maldives Stock Exchange, named “TinyHRMS” is a simple web-based system that manages payroll, attendance, leave management and staff personal information.

The aim of this project is to design a web-based application that allows the client, Maldives Stock Exchange (MSE), to streamline their human resource tasks and manage their employees in a more effective and efficient way.

The system will ensure effective utilization and maximum development of human resource, generate and maintain human resource records and allow proper interaction and timely access to accurate information to those who require the information.

## 1.2 Company Background

The direct customer of this project is Maldives Stock Exchange (MSE). The main function of the MSE is to facilitate companies raising capital through the issue of new securities and provide a regulated market for the trading of existing stocks between investors.

Maldives Stock Exchange established on 14th of April, 2002 was operated by the Capital Market Development Authority (CMDA) as part of the regulator. However, to separate the Exchange operation, Maldives Stock Exchange (MSE) was licensed as a private sector exchange by Capital Market Development Authority (CMDA) on 23<sup>rd</sup> of January 2008, under the Maldives Securities Act (MSA : 02/2006).MSE is operated by the Maldives Stock Exchange Company Pvt Ltd, effective from 24th January 2008.

The primary function of MSE is to facilitate companies raising capital through the issue of new securities. The secondary function of the MSE is to provide a regulated market for the trading of existing stocks between investors. The MSE is also the centre for trading, reporting and pricing of the stocks. The trading information is

released to the public by the MSE ensuring transparency in market dealings (Maldives Stock Exchange, 2012).

### 1.2.1 Board of Directors

The board of directors of the Exchange are as follows:

- Mr. Hassan Manik - Chief Executive Officer
- Ms. Sana Mansoor - Director
- Ms. Asiyath Nazeeha Ahmed - Director
- Mr. Riluwan Shareef - Director
- Mr. Alau Ali - Director
- Mr. Mohamed Manih Ahmed - Director
- Mr. Ibrahim Waheed - Company Secretary

### 1.2.2 Organization Chart of Maldives Stock Exchange

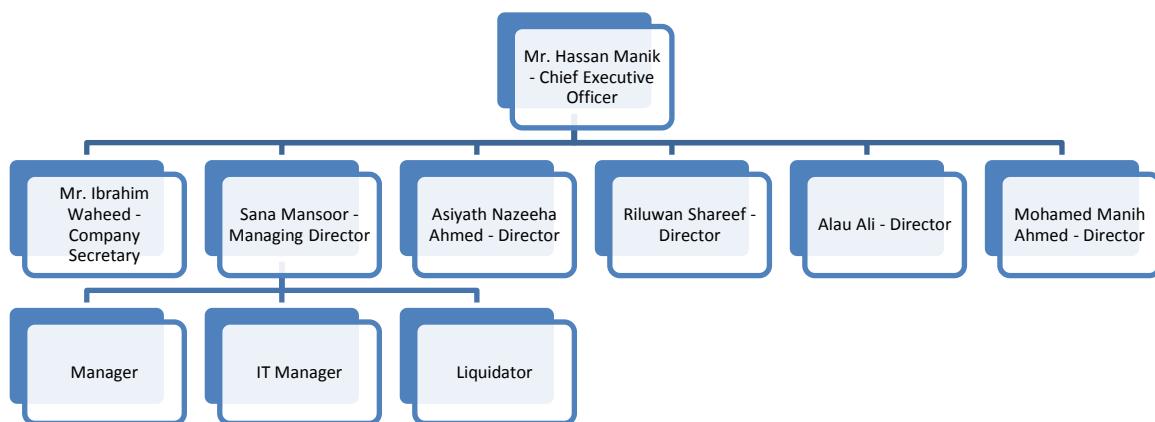


Figure 1: Low-level Organization Chart of MSE

### 1.3.Issues with Current System

**Table 1: Problem of Managing Employees in a Manual System**

The problem of	Managing employees in a manual system
effects	Errors are created during the process of maintaining attendance and doing payroll. Less secure data. Data redundancy.
the impact of which is	Unreliable data sheets. Employee dissatisfaction.
a successful solution would be	Automate the attendance and payroll systems.

**Table 2: Problem of Manual Payroll System**

The problem of	Manual Payroll System
effects	Human errors cause great loss to employee and the organization. Time consuming.
the impact of which is	Problems in payroll. Employee dissatisfaction. Loss to organization.
a successful solution would be	Automate payroll system.

**Table 3: Problem of Time Consumption in Generating Reports**

The problem of	Time Consumption in Generating Reports
effects	More time is needed to finish otherwise simple job if the system is automated.
the impact of which is	Less productive.
a successful solution would be	System needs to generate employee data automatically.

**Table 4: Problem of Inaccurate Employee Data**

The problem of effects	Inaccurate Employee Data Unreliable data.
the impact of which is	Distrust to employee data handled by the organization.
a successful solution would be	Increase data security and easy of data maintenance through reliable system.

**Table 5: Problem of Redundant Data**

The problem of Effects	Redundant Data Unrealistic data.
the impact of which is	Distrust to employee data handled by the organization Loss of storage space.
a successful solution would be	Implement data maintainable system.

**Table 6: Problem of Insecure Employee Information**

The problem of Effects	Insecure Employee Information Unwillingness to provide information to organization. Decrease respect to organization.
the impact of which is	Loss of organization professionalism. Employee dissatisfaction.
a successful solution would be	Build high security system.

#### **1.4.Benefits and Constraints of Proposed System**

After the completion of this project, the product will be used by Maldives Stock Exchange. The current manual system will be discarded and all basic HR functions will be automated. Our product contains comprehensive benefits in administration and addresses all of Maldives Stock Exchange compliance reporting needs; time-off and absence tracking and powerful import and export functions.

The user interface of the system will provide quick access to information, and multi-level security options will control who can access or view information. Searching

employee information will be very easy and it will save a lot of time. All data are stored in a centralized database and redundancy of data will not occur in this system.

## 1.5. Project Description

“TinyHRMS” is web-based application software that can manage employee information by automating the core human resource functions which is basically based on employee information, benefits and payroll processes. It has functions of employee management that increase efficiency and productivity of human resource department and will reduce the time consumption taken between processes by timely generating the necessary reports and statistics. In addition to this, the system will reduce redundant data and error scope by easily creating accurate reporting.

“TinyHRMS” contains an intuitive user interface that provides quick access to information. Sophisticated, multi-level security options controls the accessing or viewing of information stored in the system. A centralized database contains all the present and historic information about the active and non-active employees of the organization.

Furthermore, “TinyHRMS” will increase easy retention of data by providing a high level of services to the employees. Altogether, the system will reduce routine administration and promote a paperless environment.

### 1.5.1. Overview Structure of TinyHRMS

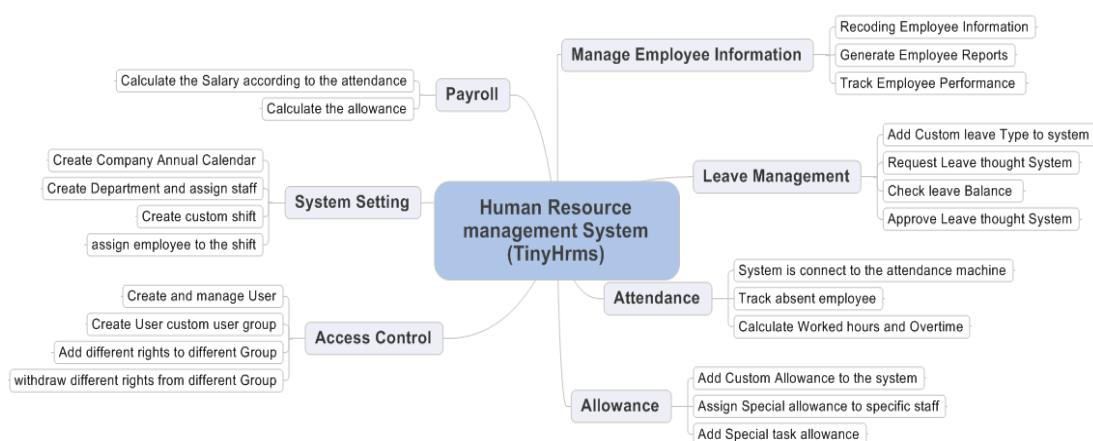


Figure 2 Overview of TinyHRMS

### 1.5.2. Important Modules of the System

- ✓ Attendance Management
- ✓ Payroll Management
- ✓ Leave Management
- ✓ Access Control
- ✓ Employee Information Management

### 1.5.3. Main Users of the System

- ✓ Staff
- ✓ HR Staff
- ✓ HR admin
- ✓ Head of Organization
- ✓ System Administrator

### 1.5.4. High level Use case Diagram

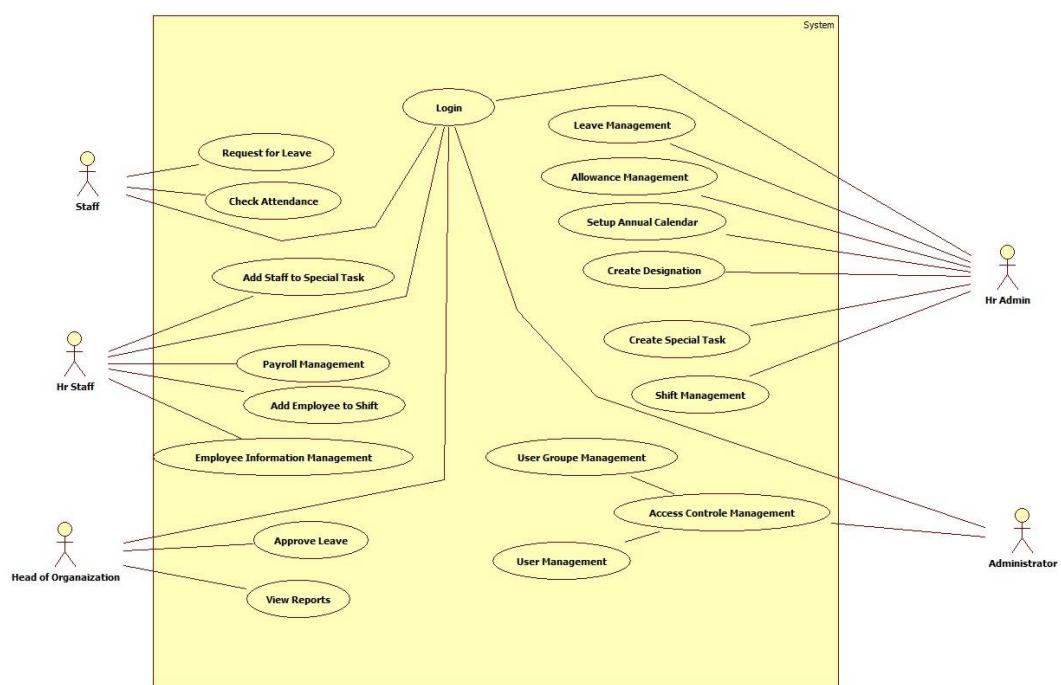


Figure 3: High-level Use case Diagram

## **1.6.Project Aims**

The aim of this project is to provide and design a web-based application that allows the client to streamline their human resource tasks and manage their employees in a more effective and efficient way.

The system will ensure effective utilization and maximum development of human resources, generate and maintain human resources records, and allow proper interaction and timely access to accurate information to those who require the information.

## **1.7.Project Objectives**

1. To manage employee information by automating core HR, benefits, and payroll processes for increased efficiency and productivity
2. To reduce the time consumption taken between processes by timely generating the necessary reports and statistics
3. To increase retention by providing a high level of service to employees.
4. To reduce redundant data, and error scope by easily creating accurate reporting and analysis.
5. To secure employee information.
6. To reduce routine administration, and promote a paperless environment.

## 1.8.Project Scope

“TinyHRMS” is a human resource management system that automates payroll, attendance management; employees leave management, employee information management and access control. The system can be managed by different user levels such as HR staff, HR admin, Admin and head of organization.

A centralized database contains all the present and historic information about the active and non-active employees of the organization. This HRMS is a single source for flawless employee administration, registration, analysis, employee life-cycle management, and all other crucial employee related information. This HRMS does not include modules such as appraisal form management and online recruitment.

## 1.9.Software and Hardware Requirements

“TinyHRMS” can run on cloud or on a local server. The system is developed using open-source technology such as Hypertext Pre-processor (PHP) and MySQL. To run this HRMS, the host needs PHP version 5.2.4 or greater, and MySQL version 5.0 or greater. Although, Apache or Nginx is the most robust and best server for running “TinyHRMS”, the software can run on any server that supports PHP and MySQL.

As “TinyHRMS” is operating system independent, and will run perfectly on Windows, Macintosh, Linux or Sun Solaris. As the software runs similar to a website, user will be able to interact with HTML interface or with the help of any web browser such as Firefox, Google Chrome or Safari. The system can also be interacted from computers which maybe tablet PC or a mobile phone that supports networking and has a web browser.

If a user wants to log on to the system, user has to open the web browser and type the URL of the application. User will be able to interact with through HTML interface, input data and retrieve data from system through the web browser. The speed of the system depends on the hardware or hosted type. If the system is hosted on local server, then it will be very fast and multiples of users can use the system concurrently without any difficulties.

However in Microsoft Internet Explorer, the system will not work perfectly and user may experience some difficulties if using mobile phone web browsers

**Table 7: Software and Hardware Requirements of the System**

<b>Hardware Requirements</b>	
Processor	Pentium IV
Min RAM	512 MB
Min Disk Space	160GB
<b>Software Requirements</b>	
Operating System	Windows, Macintosh, Linux, Sun Solaris, Android, IOS, Symbian
Web Browser	Safari, Firefox, Google Chrome, Opera, Android, Donfin, NetScape

## **1.10. Development Methodology**

System development methodology is a framework that is used to structure, plan and control the process of developing an information system. A wide variety of such frameworks have evolved over the years, each with its own recognized strengths and weaknesses, and one system development methodology is not necessarily suitable for use by all projects (Department of Health and Human Services, 2008).

Before, developing software, it is advisable to choose the methodology which suits best for the selected system. Hence, before the start of this project, we have considered some methodologies of system developed that helped us choose Waterfall model which was presumed best for the development of “TinyHRMS”.

### **1.10.1. Waterfall Model**

The waterfall model is a sequential design process. This model is frequently used in software development processes, in which improvement is seen as flowing steadily downwards (like a waterfall) through the stages of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance (Murugaiyan, 2012).

Testing of the software is carried out only when the code has been fully developed. Each work-product or activity is completed before moving on to next. Each phase of development proceeds in order without any overlapping and the tasks are scheduled to be completed in a specific period of time (Murugaiyan, 2012).

In waterfall model, requirements should be made clear and distinct before moving on to the next phase of design.

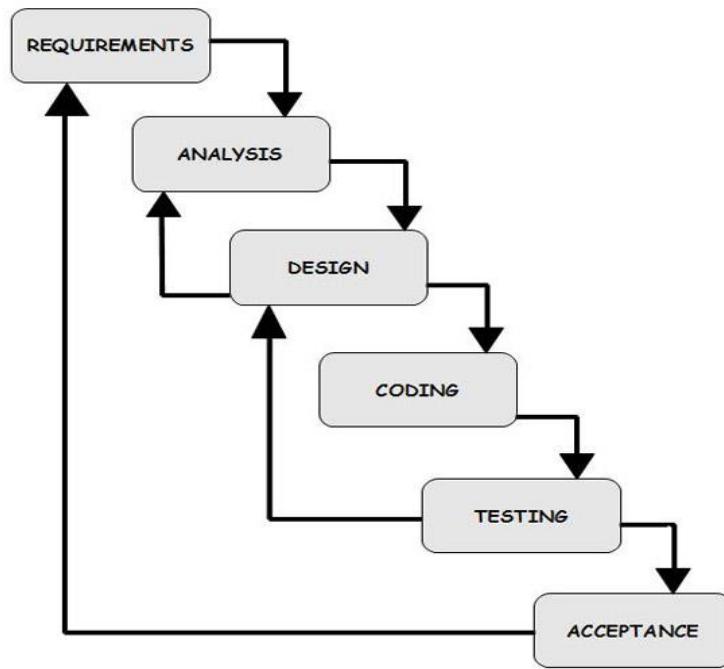


Figure 4 Waterfall Model

### Pros

Requirement is clear before development starts. Each phase is completed in specified period of time after that it moves to next phase. As its linear model, it's easy to implement. The amount of resources required to implement this model are minimal. For each phase, proper documentation is followed for the quality of the development.

### Cons

The problems with one phase are never solved completely during that phase and in fact many problems regarding a particular phase arise after the phase is signed off, this result in badly structured system.

If client want the requirement to be changed, it will not implemented in the current development process. In spite of the cons, the many pros of this model ensure that it remains one of the most popular models used in the field of software development.

### **Why Waterfall Method was Chosen for this Project**

Being a linear model, Waterfall methodology is easy to implement as amount of resources required to implement this model are minimal. This method is chosen when the requirements are clear. Each phase of Waterfall model is completed in specified period of time, before moving to a new phase.

Based on these reviews, for the development of “TinyHRMS”, the development team chose Waterfall method, as this method found out to be the most preferable method when time is a constraint of the project. This methodology was also chosen, because the requirements of the system are well-known and very clear to the development team.

#### **1.11. Project Plan**

This project consist the 7 major tasks and the duration of the project is 146 days. These 7 tasks are spited in to 40 task, which is assigned to different members of the team.

<b>Tasks</b>	<b>Duration</b>
Initiation	4 days
Project Planning	18 days
Requirement Gathering and Analysis	29 days
Design system and software	41 days
Coding	51 days
Testing	8 days
Development	4 days

##### **1.11.1 Gantt Chart**

**Refer to Appendix B**

## CHAPTER 2: LITERATURE REVIEW

### 2.1.Introduction

This document is a review of some of the relevant and recent scholarly work on the importance of HRMS to organizations, types of methodologies for developing of software, and on programming methods.

This literature review shows that for the development of “TinyHRMS”, Waterfall is the most appropriate development methodology, and Model View Controller is the preferable software pattern for programming the system. It also shows that PHP is the ideal tool for developing this web-based HRMS.

### 2.2.Importance of HRMS

With the increasing effect of globalization and technology, various functions of human resource management systems have become central in managing organizations effectively. (Obeidat, 2012) Hence, organizations must treat information as any other resource or asset. For information to exhibit quality, it must be organized, managed and disseminated in an effective manner. (Obeidat, 2012)

Human Resource management systems support activities such as identifying potential employees, maintaining records of existing employees. HR systems help senior management to identify the manpower requirements in order to meet the organization’s long term business plans and strategic goals. (Obeidat, 2012) Middle management uses human resources systems to monitor and analyse the recruitment, allocation and compensation of employees. Operational management uses HR systems to track the recruitment and placement of the employees. HRMS can also support various HR practices such as workforce planning, staffing, compensation programs, salary forecasts, pay budgets and labour or employee relations (Yasemin Bal).

The client, Maldives Stock Exchange, is an organization small in size. MSE employs eleven staffs and manages its HR functions manually. The organization seeks an automated HR system that will cater its exact HRM needs.

### **2.3.Similar System that Exist**

#### **2.3.1. Sage Abra**

Sage HRMS, formally known as Sage Abra, is built for mid-size organizations which has core Human Resource functions such as talent management, work force management and learning management system in order to optimize employee lifecycle (HRMS, Sage HRMS, 2011).

This software is developed by HR professionals who have face challenges in Human resource working environment and had the experience of real time problem and solutions. Sage HRMS is built for organization having employee between 50 - 3000 employees (HRMS, 2012).

#### **2.3.2. Fi-es HRMS**

Fi-es HRMS is developed locally in Maldives, and is in use in Maldives and other countries. This system has attendance management, calculating leave, absent days, late coming etc. It also has payroll calculation features and recruitment features. System has reporting capabilities to generate monthly reports. Fi-es help the organization speeding up the decision making of the organization (fi-es, 2012).

### **2.4.System Development Methodology**

A Software Development Life Cycle (SDLC) binds to the important stages that are necessary for developers like planning, analysis, design and implementation. Several SDLC models in use include Waterfall, Agile, V-Model, Rapid prototyping, Spiral, etc.

Developers of “TinyHRMS” considered most commonly known and used models; Waterfall, Agile and V-Model. A comparative study is conducted on these models before choosing the most appropriate model for the development of this HRMS.

### 2.4.1. Comparative Study on Development Methodologies

#### Waterfall Model

Waterfall model is the sequential development model where its requirements should be made clear and distinct before moving on to the next phase of design. (Boehm, May 1988) Testing is carried out once the code has been fully developed and each work-product or activity is completed before moving on to next. Each phase of development proceeds in order without any overlapping and the tasks are scheduled to be completed in a specific period of time (Murugaiyan, 2012).

The documentation and testing happens at the end of each phase, which helps in maintaining the quality of the project.

#### *Advantages of Waterfall Model*

Clear requirements are gathered before the development of the system. Each phase is completed in specified period of time, before moving to a new phase. (Murugaiyan, 2012). As its linear model, it's easy to implement and the amount of resources required to implement this model are minimal. Proper documentation is followed for the quality of the development of the system (Murugaiyan, 2012).

#### *Disadvantages of Waterfall Model*

Problems with one phase are never solved completely during that phase and problems regarding a particular phase arise after the phase is signed off. If the client wants the requirement to be changed, it will not be implemented in the current development process (Murugaiyan, 2012).

#### V- Model

V-model or Verification model is an improved version of the Waterfall method. This developmental process is stable and depends on the confirmation from the preceding steps. In contrast to the Waterfall method, V-model was

designed so that the stages would return back upwards when the coding phase is completed (Murugaiyan, 2012).

In each and every phase, the product is checked and verified before moving on to the next stage. Both the developer and tester in V-model works parallel to each other. In this model, test cases are prepared based on the requirements the system. After this, the coding will be done. In V-model, it shows a relationship between each development stages and testing stages (Murugaiyan, 2012).

### ***Advantages of V-Model***

In this model, the main disadvantage is that tester role is present in the beginning of the requirement phase.

### ***Disadvantages of V-Model***

It is very rigid and less flexible. If any changes occur half way through, not only the requirements documents but also the test documentation will have to be updated. Not recommended for short term projects as it needs to be reviewed at each stage of the process. V-model is mostly used and is most suitable for large organizations that have more resources.

### **Agile Model**

The term agile stands for 'moving quickly' and therefore this methodology requires a team of skilful members who can challenge the fluctuating requirements. In this method, the software is released frequently with more updates and this makes customers more satisfied (Murugaiyan, 2012).

### ***Advantages of Agile Model***

Has ability to respond to the changing requirements of the project. No guesswork between the development team and the customer, as there is face to face communication and continuous inputs from the client (Boehm, May 1988).

### ***Disadvantages of Agile Model***

Recommended to smaller projects but for larger projects it is difficult to judge the time and effort required in the project. Only senior developers are in a better position to take the decisions necessary for the agile type of development, which leaves hardly any place for newbie programmers.

## **2.5.Pattern-Oriented Software Architecture**

Creating of interactive software systems with multiple types of user interfaces is expensive and with high error scope, when the user interface is closely interwoven with the functional core (Deacon, 2009). This problem can be avoided by the separating user interface tasks from the functional core. To solve this problem, two software patterns are used and they are related to the way of building the applications user interface. These are the Model-View- Controller (MVC) and Presentation-Abstraction-Control (PAC).

### **2.5.1. Presentation-Abstraction-Control (PAC)**

PAC design pattern is based on the concept of cooperative agents, which are organized in a hierarchical structure. Each agent is a unitary aspect of the system, which works as a node in the hierarchy of agents and consists of the components of presentation, abstraction and control according to this pattern, interactive systems consist of cooperative agents. One type of agents specialized in human computer interaction accepts user input and displays data. Other agents maintain the system data model and provide functionality that is based on such data. Additional agents are responsible for different tasks such as error management or communication with other software systems

#### **Advantages of PAC**

Different semantic concepts in the application domain are presented by separate agents, maintaining its own state and data, coordinated with, yet independent of other PAC agents. Individual PAC agents offer their own human-computer interaction allowing the development of dedicated data.

Provides support for change and expansion allowing modification of the data model in the base of the PAC agent or modification of its user interface. All PAC agents communicate with each other through predefined interfaces. Existing agents can dynamically register the new PAC agents to provide communication and cooperation. Mode of coordination of agents in the PAC pattern provides a good basis for multitasking. PAC agents can easily be distributed to different threads, processes or machines.

#### **Disadvantages of PAC**

Increases complexity of the system as implementation of each semantic concept as its own PAC agent can result in a complex system structure. Agents must be coordinated and controlled, which requires additional coordinating agents (Plakalović D., 2010 JANUARY). The quality of implementation of the control component is critical both for effective cooperation among agents and the entire system architecture quality. Individual roles of the control component should be strictly separated from each other. Implementation of these roles should not be dependent on the specific details of other agents. Additional tasks related to communication between PAC agents can affect the efficiency of the system. If the agents are distributed, the data transfer also requires inter-process cooperation, which includes a series of actions that further burden the communication system (Plakalović D., 2010 JANUARY).

#### **2.5.2. Model View Controller (MVC)**

Model View Controller (MVC) is architecture of programming that separates database, business logic and user interface. The model consists of application data and business rules, and the controller mediates input, converting it to commands for the model or view. A view can be any output representation of data, such as a chart or a diagram. Multiple views of the same data are possible, such as a pie chart for management and a tabular view for accountants (Deacon, 2009).

### **Advantages of MVC**

In MVC, pages can be downloaded faster and Search Engine Optimization (SEO) will be better. MVC break the system into three layers; database, business logic and view. Multiple programmers can read the code and develop it which aids in faster development. Also, the URL of the application is user friendly (Plakalović D., 2010 JANUARY).

### **Disadvantages of MVC**

The whole controller approach needs to define a class for every controller and in a separate file. Every action in a controller is a function which does not differ much from some simpler approaches. There are templates and view directory which has separate HTML codes for each and this means that there will be more files and codes (Plakalović D., 2010 JANUARY).

## **2.6. Tools for Dynamic Websites**

Creation of database-driven websites used to be complex and time-consuming before server-scripting tools were invented to make it easier for publishers to generate web content automatically from their databases instead of manually coding it in HTML. The two most popular and competing technologies, for creating database-driven websites are open-source PHP and Microsoft's ASP. These two technologies were considered before choosing the best technology for the development of “TinyHRMS”.

### **2.6.1. Hypertext Pre-processor (PHP)**

PHP is widely used and popular in Web development community, following the proliferation of Apache on Linux and UNIX servers. (Hartman, 2001)

#### **Advantages of PHP (Hartman, 2001)**

PHP is open-source code, and works with a multitude of servers on many different operating systems including Microsoft's. PHP is designed specifically as a Web-scripting language, hence it is very efficient.

### **Disadvantages of PHP (Hartman, 2001)**

PHP is not suitable for making desktop applications, and error handling is traditionally considered poor when compared to other programming languages.

#### **2.6.2. Active Server Pages (ASP)**

ASP is Microsoft's solution for generating dynamic web pages. ASP runs exclusively on Microsoft IIS servers. Even though there are third-party products, such as Chili! Soft, for running ASP on other servers and other platforms, they have not been widely adopted (Hartman, 2001).

#### **Advantages (Hartman, 2001)**

ASP is easy to learn. With the exception of some Web-specific tasks, ASP is easy to use.

#### **Disadvantages (Hartman, 2001)**

In ASP, certain tasks require more steps than PHP. ASP runs exclusively on Microsoft IIS servers, hence cost is required.

## **2.7. Conclusion**

Maldives Stock Exchange seeks HR software that is tailored to the needs of the organization. Existing software such as Sage Abra are too complex for the organization, and Fi-es HR System is expensive for the client. Hence, the client chose “TinyHRMS” which is to be developed for the requirements of MSE, free of charge.

V- Model is chosen for large project, and is not recommended for short-term projects as frequent changes in requirements and proper validation needs to take place in each stage.

Even though, agile method is recommended to smaller projects, it is difficult to judge the time and effort required in the project. And highly skilled senior developers are in

a better position to take the decisions necessary for the agile type of development, which leaves hardly any place for new programmers.

Being a linear model, Waterfall methodology is easy to implement as amount of resources required to implement this model are minimal. This method is chosen when the requirements are clear. Each phase of Waterfall model is completed in specified period of time, before moving to a new phase.

Based on these reviews, for the development of “TinyHRMS”, the development team chose Waterfall method, as this method found out to be the most preferable method when time is a constraint of the project. This methodology was also chosen, because the requirements of the system are well-known and very clear to the development team.

As a programming method, MVC was selected over PAC by the development team as multiple programmers can read the code and develop it which aids in faster development. Furthermore, this application is very fast and its URL is user friendly.

As a tool for development, PHP was preferred as it is open-source, and works with a multitude of servers, on many different operating systems, including Microsoft's.

## CHAPTER 3: REQUIREMENTS ANALYSIS

### 3.1. Introduction

This document is intended to provide the software requirements for Human Resource Management System, TinyHRMS. Based on information gathered from the end-users and the client, the end-product will meet the needs of the client, Maldives Stock Exchange, in managing its HR functions.

### 3.2. Requirements Gathering

Gathering requirements is a critical step for every project. By gathering up front enables better planning, accurate cost estimates, and mostly improved customer satisfaction and adoption of the final product.

Hence, a great importance was given on gathering correct requirements. Requirement gathering techniques used for this project are interviewing, use cases, and observation.

#### 3.2.1 Interviewing

Client was interviewed using a set of open-ended questions, and then more probing questions were asked that helped to uncover requirements. These interviews were planned ahead of time. Some of the interviews were conducted face-to-face, while some interviews were conducted over the phone. Even though this technique proved to be time consuming, it was good for exploring many important issues of the system.

#### 3.2.2 Use cases

Use cases helped to uncover behavioural and functional requirements of the systems. Use cases improved the overall development process by enhancing understanding of requirements early in the software development life cycle.

### 3.2.3 Observation

This technique involved observing users by watching their behaviours in the clients/users natural settings. Active observing method was used as it is more effective at getting an understanding of the existing business process. This technique helped to identify process flows, opportunities for improvement and uncover implicit requirements.

## 3.3 Requirements Summary

### 3.3.1. Modules of HRMS

“TinyHRMS” consist of five different users; Staff, HR Staff, HR Admin, Admin and Head of Organization. The software includes major 5 modules; Payroll Management, Attendance Management, Leave Management, Employee Information and Access Control.

#### Attendance Management

Attendance management module is one of the important modules in this system. Based on attendance information the system will generate the employee salary. When users check-in through the fingerprint machine then the check in time, date and employee id will be stored in MySQL database. Each employee can view his/her attendance information. When an employee needs to change his/her attendance information then he/she has to create a request in the system. HR staff attendance information will only be amended after the request from an employee.

#### Payroll Management

HR head can add different type of allowance to the system and assign it to a specific staff. When calculating salary, the system will get the total allowance and work hour of the employee from the attendance module. This will generate the monthly salary. Head of HR can add and delete the allowance from system.

### Leave Management

In this HRMS, Head of HR can add leave to the system. The system will maintain all the leave in separate data table. When an employee wants a leave, he/she can request for the leave through the system. When employee requests for leave, the leave is confirmed with an approval of the Head of organization. Instead of approving the leave, Head of HR or Head of organization can amend or reject the leave. Employee can see the status of the leave through employee interface. When a leave is approved then the employee can take the leave chit.

### Access Control

Access Control enables to create different access area and different access right. System administrator can assign any role to the user and withdraw the right from user at any time.

### Employee Information management

The main objective of this function is managing employee information. In this function the organization will be split in to departments. HR head can create departments and assign staff and department head can identify the performance of the departments with the help on attendance information.

#### **3.3.2. Main Users of the System**

##### Staff

The STAFF as an actor of this system can access twelve functions of the system which are:

- |                                     |                                  |
|-------------------------------------|----------------------------------|
| ✓ View department leave Schedule    | ✓ Check Request Status           |
| ✓ View Attendance                   | ✓ Check In (Fingerprint Machine) |
| ✓ View Employee Contact information | ✓ Amend Leave Request            |
| ✓ Check Leave Availability          | ✓ Request for Attendance Change  |
| ✓ Edit profile                      | ✓ Request For leave              |
|                                     | ✓ View Department Information    |

Above functions are general functions that can be accessed by all the staff of the organization. These functions are called service functions.

#### HR Staff

HR Staff is the most leading actor of this origination who will be doing all the operational tasks of the system. The admin and HR head will facilitate the data and option to complete the task of HR staff. When HR staff does his/her work then all the staff can view the relevant data of his/her. The functions of HR staff include:

- |                                     |                                 |
|-------------------------------------|---------------------------------|
| ✓ Payroll process                   | ✓ Withdraw leave                |
| ✓ Create employee                   | ✓ Assign leave                  |
| ✓ Update employee information       | ✓ Assign allowance              |
| ✓ Update attendance                 | ✓ Update department information |
| ✓ Assign employee to the department | ✓ Create Special Task           |
|                                     | ✓ Assign Task to Employee       |

#### HR Admin

HR Admin is one of the powerful users in this system. HR admin can edit and update all the data except system setting. Hr admin will have access to all the work that Hr Staff and other employees do. Following functions are special functions for Head of HR.

- |                      |                      |
|----------------------|----------------------|
| ✓ Create Leave       | ✓ Delete Designation |
| ✓ Delete Leave       | ✓ Create Allowance   |
| ✓ Update Leave       | ✓ Update Designation |
| ✓ Delete Employee    |                      |
| ✓ Create Departments |                      |
| ✓ Delete Department  |                      |
| ✓ Update allowance   |                      |
| ✓ Delete allowance   |                      |
| ✓ Create Designation |                      |

### **System Administrator**

The duties of a system administrator are wide-ranging which include:

- ✓ Create new User
- ✓ Withdraw user role form user
- ✓ Delete user
- ✓ Update user

### **Head of Organization**

These staffs include organization such as managing director, manager and other director who involve in making decisions according the report generates from system. Following functions are specialized for Head of Organization

- ✓ Approving Leave
- ✓ Check generated reports

### 3.3.3. Use Case Diagram

A use-case diagram is typically used to communicate the high-level functions of the system and the system's scope. Use case diagrams illustrate a unit of functionality provided by the system and help development teams visualize the functional requirements of a system (Bell, 2003).

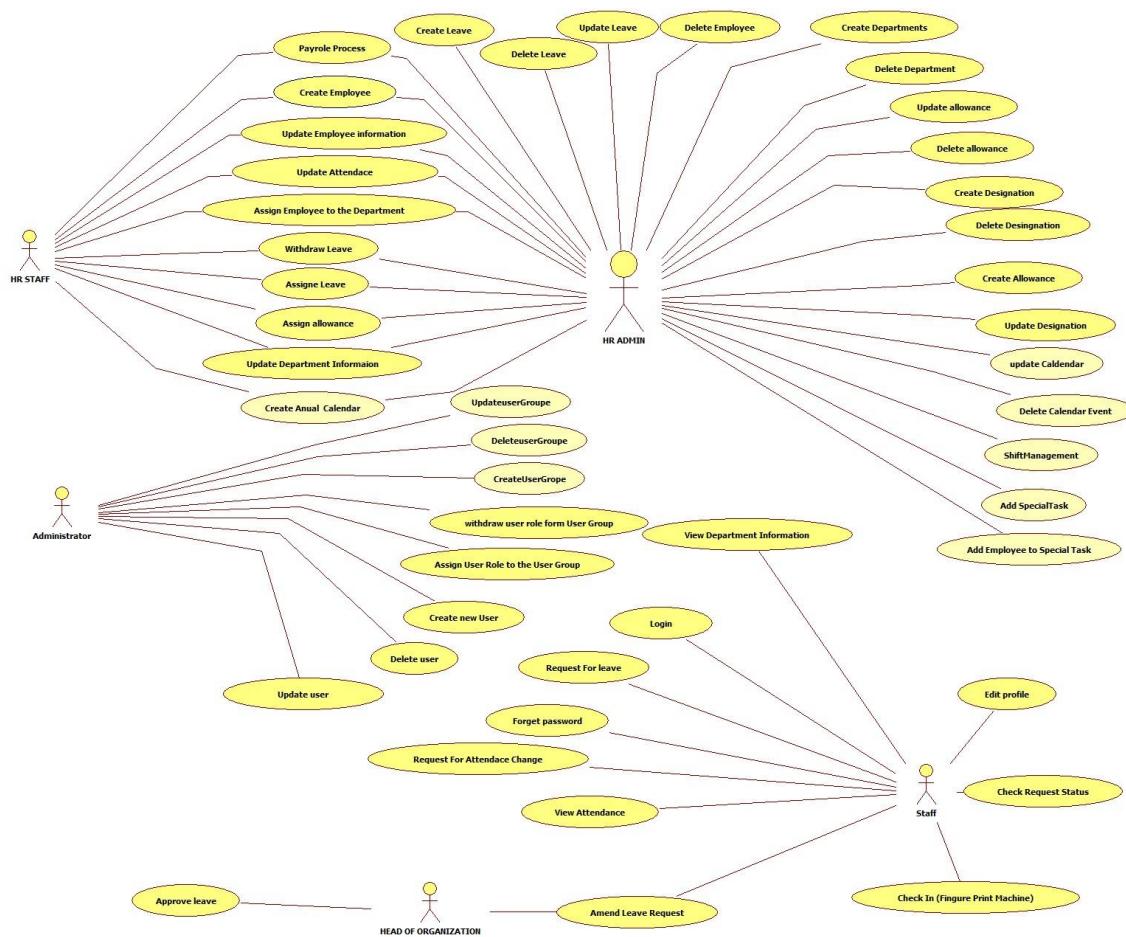


Figure 5 Use case Diagram of TinyHRMS

### 3.3.4. High-level Use Cases

**Table 8:** High-level Use Case 1 - Create User

<b>Use Case 1</b>	<b>Create User</b>
<b>Actor</b>	Admin
<b>Type</b>	Primary
<b>Description</b>	In this use case, admin creates a new user for every employee.

**Table 9:** High-level Use Case 2- Log-in

<b>Use Case 2</b>	<b>Log-in</b>
<b>Actor</b>	Employee
<b>Type</b>	Primary
<b>Description</b>	Log-in is the credentials required to obtain access to the system When user enters his/her username and password, login process starts.

**Table 10:** High-level Use Case 3 - Forgot Password

<b>Use Case 3</b>	<b>Forgot Password</b>
Actor	Employee
Type	Primary
Description	If employee forgets the password, employee uses this function to reset his/her password.

**Table 11:** High-level Use Case 4 - Edit Profile

<b>Use Case 4</b>	<b>Edit Profile</b>
<b>Actor</b>	Staff
<b>Type</b>	Primary
<b>Description</b>	Staff sends request with required documents to HR. Request is seen by the HR Staff and HR Admin and HR Admin and HR Staff checks for the validity, and edits the profile.

**Table 12: High-level Use Case 5 - Delete User**

<b>Use Case 5</b>	<b>Delete User</b>
<b>Actor</b>	System Administrator
<b>Type</b>	Primary
<b>Description</b>	System administrator selects user and presses “DELETE” button.

**Table 13: High-level Use Case 6 - Update User**

<b>Use Case 6</b>	<b>Update User</b>
<b>Actor</b>	System Administrator
<b>Type</b>	Primary
<b>Description</b>	System administrator selects user and opens in editable mode, then system administrator sets the new values and saves it.

**Table 14: High-level Use Case 7 - Assign User role to user group**

<b>Use Case 7</b>	<b>Assign Role to the User Group</b>
<b>Actor</b>	System Administrator
<b>Type</b>	Primary
<b>Description</b>	System administrator selects user group from user group list, and selects available right from list and presses “ASSIGN” button.

**Table 15: High-level Use Case 8 - Withdraw User Group Role**

<b>Use Case 8</b>	<b>Withdraw User Group Role</b>
<b>Actor</b>	System Administrator
<b>Type</b>	Primary
<b>Description</b>	System Administrator selects user group from user group list, and selects current rights from list and presses “REMOVE” button.

**Table 16: High-level Use Case 9 - Create New Department**

<b>Use Case 9</b>	<b>Create New Department</b>
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin receives new department information. The HR Admin enters new department information to the

	system.
--	---------

**Table 17: High-level Use Case 10 - Update Department Information**

Use Case 10	Update Department Information
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin receives amendment for existing department's information. The HR Admin updates department information as per the amendment.

**Table 18: High-level Use Case 11 - Delete Department**

Use Case 11	Delete Department
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin desires to delete a department. The HR Admin deletes the department.

**Table 19: High-level Use Case 12 - Create New Employee**

Use Case 12	Create New Employee
<b>Actor(s)</b>	HR Staff HR Admin
<b>Description</b>	This use case begins when HR Staff receives new employee information. The HR Staff enters new employee information to the system.

**Table 20: High-level Use Case 13 - Update Employee Information**

Use Case 13	Update Employee Information
<b>Actor(s)</b>	HR Staff HR Admin
<b>Description</b>	This use case begins when HR Staff receives amendment for existing employee's information. The HR Staff updates employee information as per the amendment.

**Table 21: High-level Use Case 14 - Delete Employee**

<b>Use Case 14</b>	<b>Delete Employee</b>
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin desires to delete an employee. The HR Admin deletes the employee.

**Table 22: High-level Use Case 15 - Create Allowance**

<b>Use Case 15</b>	<b>Create Allowance</b>
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin receives new allowance information. The HR Admin enters new allowance information to the system.

**Table 23: High-level Use Case 16 - Update Allowance**

<b>Use Case 16</b>	<b>Update Allowance</b>
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin receives new allowance information. The HR Admin selects the allowance and updates new allowance information to the system.

**Table 24: High-level Use Case 17 - Delete Allowance**

<b>Use Case 17</b>	<b>Delete Allowance</b>
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin desires to delete an allowance. The HR Admin deletes the allowance.

**Table 25: High-level Use Case 18 - Assign Allowance**

<b>Use Case 18</b>	<b>Assign Allowance</b>
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin desires to assign allowance to the staff. HR admin selects staff and selects allowance from list and HR admin updates the employee allowance.

**Table 26: High-level Use Case 19 - Create New Leave**

<b>Use Case 19</b>	<b>Create New Leave</b>
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin receives new leave information. The HR Admin enters new leave information to the system.

**Table 27: High-level Use Case 20 - Delete Leave**

<b>Use Case 20</b>	<b>Delete Leave</b>
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin wishes to delete an existing leave. The HR Admin deletes an existing leave.

**Table 28: High-level Use Case 21 - Update Leave Information**

<b>Use Case 21</b>	<b>Update Leave Information</b>
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin receives amendment for existing leave information. The HR Admin updates leave information as per the amendment.

**Table 29: High-level Use Case 22 - Approve/Decline Leave**

<b>Use Case 22</b>	<b>Approve/Decline Leave</b>
--------------------	------------------------------

<b>Actor(s)</b>	Head of Organization
<b>Description</b>	This use case begins when Head of Organization desires to approve/decline requested leave application. The Head of Organization approves/declines the leave request.

**Table 30: High-level Use Case 23 - Request Leave**

<b>Use Case 23</b>	<b>Request Leave</b>
<b>Actor(s)</b>	Employee
<b>Description</b>	When employee needs to take leave then he/she have to request through the system. User selects the leave type ,dates of the leave and reason of the leave and presses “REQUEST” button.

**Table 31: High-level Use Case 24 - Amend Leave Request**

<b>Use Case 24</b>	<b>Amend Leave Request</b>
<b>Actor(s)</b>	Head of department ,Employee
<b>Description</b>	This use case begins when the Head of Organization opens employee leave management interface. User selects the request and updates the leave request.

**Table 32: High-level Use Case 25 - Check-In**

<b>Use Case 25</b>	<b>Check-In</b>
<b>Actor</b>	Staff, HR Staff, HR Admin
<b>Type</b>	Primary
<b>Description</b>	When staff wants to check in, he has to enter the finger print to the fingerprint machine.

**Table 33: High-level Use Case 26 - Request for Update Attendance**

<b>Use Case 26</b>	<b>Request for Update Attendance</b>
<b>Actor</b>	Staff
<b>Type</b>	Primary
<b>Description</b>	This use case begins when Staff requests for amend. When request is sent

	request is seen by the HR Staff and HR Admin Staff so he can amend.
--	---

**Table 34: High-level Use Case 27 - View Attendance**

<b>Use Case 27</b>	<b>View Attendance</b>
<b>Actor</b>	Staff, HR Staff, HR Admin
<b>Type</b>	Primary
<b>Description</b>	When staff enters the page attendance is seen.

**Table 35: High-level Use Case 28 - Create Designation**

<b>Use Case 28</b>	<b>Create Designation</b>
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Staff receives new designation information. The HR Staff enters new designation information to the system.

**Table 36: High-level Use Case 29 - Update Designation**

<b>Use Case 29</b>	<b>Update Designation</b>
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin receives amendment for existing Designation information. The HR Admin updates Designation information as per the amendment.

**Table 37: High-level Use Case 30 - Delete Designation**

<b>Use Case 30</b>	<b>Delete Designation</b>
<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin wishes to delete an existing designation. The HR Admin deletes an existing designation.

**Table 38: High-level Use Case 31- Payroll Process**

<b>Use Case 31</b>	<b>Payroll Process</b>
--------------------	------------------------

<b>Actor(s)</b>	HR Admin
<b>Description</b>	This use case begins when HR Admin wishes to process payroll. HR admin selects Staff and presses the “PROCESS PAYROLL” button

**Table 39: High-level Use Case 32 - Update Attendance**

<b>Use Case 32</b>	<b>Update Attendance</b>
<b>Actor</b>	HR Admin
<b>Type</b>	Primary
<b>Description</b>	System HR Staff selects user and opens in editable mode, then system HR Staff sets the new values and saves it.

**Table 40: High-level Use Case 33 - Assign Employee to the Department**

<b>Use Case 33</b>	<b>Assign Employee to the Department</b>
<b>Actor</b>	HR Staff, HR Admin
<b>Type</b>	Primary
<b>Description</b>	HR Staff selects user from user list, and selects available department from list and presses “ASSIGN” button.

**Table 41: High-level Use Case 34 – Withdraw Leave**

<b>Use Case 34</b>	<b>Withdraw Leave</b>
<b>Actor</b>	HR Staff
<b>Type</b>	Primary
<b>Description</b>	System HR Staff selects user from user list, and selects current leave from list and presses “REMOVE” button.

**Table 42: High-level Use Case 35 - Assign Leave**

<b>Use Case 35</b>	<b>Assign Leave</b>
<b>Actor</b>	HR Staff, HR Admin
<b>Type</b>	Primary
<b>Description</b>	HR Staff selects user from user list, and selects available special leave from list and presses “ASSIGN” button.

**Table 43: High-level Use Case 36 - Update Department Information**

<b>Use Case 36</b>	<b>Update Department Information</b>
<b>Actor</b>	HR Staff
<b>Type</b>	Primary
<b>Description</b>	System HR Staff selects Department information and opens in editable mode, then system HR Staff sets the new values to name, detail saves it.

**Table 44: High-level Use Case 37 - Update Calendar Event**

<b>Use Case 37</b>	<b>Update Calendar Event</b>
<b>Actor</b>	HR Staff
<b>Type</b>	Primary
<b>Description</b>	System HR Staff selects Calendar Event and opens in editable mode, then system HR Staff sets the new values to name, detail or date and saves it.

**Table 45: High-level Use Case 38 - Delete Calendar Event**

<b>Use Case 38</b>	<b>Delete Calendar Event</b>
<b>Actor(s)</b>	HR Staff
<b>Description</b>	This use case begins when HR Staff desires to delete a Calendar Event. The HR Staff deletes the department.

**Table 46: High-level Use Case 39 - Create User group**

<b>Use Case 39</b>	<b>Create User group</b>
<b>Actor(s)</b>	HR Staff
<b>Description</b>	This use case begins when HR Staff receives new User group information. The HR Staff enters new User group information to the system.

**Table 47: High-level Use Case 40 - Delete User group**

<b>Use Case 40</b>	<b>Delete User group</b>
<b>Actor(s)</b>	HR Staff
<b>Description</b>	This use case begins when HR Staff desires to delete user group. The HR Staff deletes the deposer group.

**Table 48: High-level Use Case 41 – Create Shift**

<b>Use Case 41</b>	<b>Create Shift</b>
<b>Actor(s)</b>	HR Staff
<b>Description</b>	This use case begins when HR Staff receives new shift information. The HR Staff enters new shift information to the system.

**Table 49: High-level Use Case 42 - Delete Shift**

<b>Use Case 42</b>	<b>Delete Shift</b>
<b>Actor(s)</b>	HR Staff
<b>Description</b>	This use case begins when HR Staff desires to delete a Shift. The HR Staff deletes the shift.

**Table 50: High-level Use Case 43 - Update Shift**

<b>Use Case 43</b>	<b>Update Shift</b>
<b>Actor</b>	HR Staff
<b>Type</b>	Primary
<b>Description</b>	System HR Staff selects Shift and opens in editable mode, then system HR Staff sets the new values to name, in time or out time and saves it.

**Table 51: High-level Use Case 44 - Create Special task**

<b>Use Case 44</b>	<b>Create Special task</b>
<b>Actor(s)</b>	HR Staff
<b>Description</b>	This use case begins when HR Staff receives new task information. The HR Staff enters new task information to the system.

**Table 52: High-level Use Case 45 - Update Special task**

<b>Use Case 45</b>	<b>Update Special task</b>
<b>Actor</b>	HR Admin
<b>Type</b>	Primary
<b>Description</b>	System HR Staff selects Special task and opens in editable mode, then system HR Staff sets the new values to name, detail, rate or date and saves

	it.
--	-----

**Table 53: High-level Use Case 46 - Delete task**

<b>Use Case 46</b>	<b>Delete task</b>
<b>Actor(s)</b>	HR Staff
<b>Description</b>	This use case begins when HR Staff desires to delete a task. The HR Staff deletes the task.

**Table 54: High-level Use Case 47 - Assign employee to special task Employee**

<b>Use Case 47</b>	Assign employee to special task Employee
<b>Actor</b>	HR Staff, HR Admin
<b>Type</b>	Primary
<b>Description</b>	HR Staff selects user from user list, and selects available special task from list and presses “ASSIGN” button.

**Table 55: High-level Use Case 48 - Withdraw Employee**

<b>Use Case 48</b>	<b>Withdraw Employee</b>
<b>Actor</b>	HR Staff
<b>Type</b>	Primary
<b>Description</b>	System HR Staff selects user from user list, and selects current task from list and presses “REMOVE” button.

**Table 56: High-level Use Case 49 - Create Special task**

<b>Use Case 49</b>	<b>Create Special task</b>
<b>Actor(s)</b>	HR Staff, HR Admin
<b>Description</b>	This use case begins when HR Staff receives new special task information. The HR Staff enters new special task information to the system.

### 3.3.5. Expanded Use Cases

**Table 57: Expanded Use Case 1- Create User**

<b>Use Case 1</b>	<b>Create User</b>
<b>Goal in Context</b>	To allow admin add new user to the system.
<b>Primary Actor</b>	admin
<b>Secondary Actor</b>	
<b>Trigger</b>	Admin presses “Create New User” button.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when admin wishes to create a new user.	
2. User presses the “Create New User” button.	3. System shows the input form that includes employee id, username and password.
4. User presses the “SUBMIT” button.	5. System creates a new user.
<b>Alternative course of events</b>	
Line 4: a- If username already exists in the system, an error message will be displayed. b- If employee id already exists in the system, an error message will be displayed.	

**Table 58: Expanded Use Case 2- Log in**

<b>Use Case 2</b>	<b>Log in</b>
<b>Goal in Context</b>	To allow employee to login to the system.
<b>Primary Actor</b>	Employee
<b>Secondary Actor</b>	
<b>Trigger</b>	Employee enters login information.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when employee desire to login in to the system.	
2. Employee enters login Information.	3. System validates employee input.
4. Employee presses login button.	5. User will login.
<b>Alternative course of events</b>	
Line 4: a- If username and password are wrong then system stops the login process.	

**Table: Expanded Use Case 3- Forgot Password**

<b>Use Case 3</b>	<b>Forgot Password</b>
<b>Goal in Context</b>	To allow employee to reset their password when they forget password.
<b>Primary Actor</b>	Employee
<b>Secondary Actor</b>	
<b>Trigger</b>	When employee presses “FORGOT PASSWORD” button.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>

1. This use case begins when employee wishes to reset password.	
2. Employee “FORGOT PASSWORD” button.	3. System shows input box for entering secret answer.
4. Employee enters the secret answer.	5. System sends password to mailbox.
<b>Alternative course of events</b>	
Line 4: a- If secret answer is wrong then the process stops.	

**Table 59: Expanded Use Cases 4 - Edit Profile**

Use Case 4	Edit Profile			
<b>Goal in Context</b>	Edit profile of the employee.			
<b>Primary Actor</b>	Staff			
<b>Secondary Actor</b>	HR Staff, HR Admin			
<b>Trigger</b>	When Staff request to edit his/her information.			
<b>Typical Course of Events</b>				
Actor Action	<b>System Response</b>			
1. Staff sends request and required documents.	Request is seen by the HR Staff and HR Admin.			
2. HR Admin and HR Staff checks for the validity and edits the profile.	Profile is updated.			
<b>Alternative Course of Events</b>				
Line 1a: If data is not valid HR Admin or HR Staff can reject the request or send back for amendment.				

**Table 60: Expanded Use Cases 5 - Delete User**

<b>Use Case 5</b>	<b>Delete User</b>
<b>Goal in Context</b>	To allow admin head to remove user from the system when needed.
<b>Primary Actor</b>	Administrator
<b>Secondary Actor</b>	
<b>Trigger</b>	Administrator searches the user.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when Administrator search the user.	2. The system will show the user.
3. Press delete button.	4. Confirmation message will appear with two option (OK, Cancel).
4. User presses OK.	5. System removes user from the system.
<b>Alternative course of events</b>	
Line 3: a-If the selected designation is assign to one or more staff, the system shows error.	

**Table: Expanded Use Cases 6 - Update Users**

<b>Use Case 6</b>	<b>Update User</b>
<b>Goal in Context</b>	To allow HR head to Update User information.
<b>Primary Actor</b>	System admin
<b>Secondary Actor</b>	
<b>Trigger</b>	System admin selects user from user list.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when system admin select user form user list.	2. Edit button enables.
3. Press “EDIT” button.	4. System shows user information is editable format.
4. User change current information (username, password, status).	
Press “UPDATE” button.	System shows the success message.
<b>Alternative course of events</b>	
Line 4: a-if required field are not fill, system shows the error message.	

**Table 61: Expanded Use Cases 7 - Assign Role to the User**

<b>Use Case 7</b>	<b>Assign Role to the User</b>
<b>Goal in Context</b>	To allow Admin to add Role to the user.
<b>Primary Actor</b> <b>Secondary Actor</b>	admin
<b>Trigger</b>	Admin presses the access control management menu.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when admin wishes to add new role to the user.	
2. User selects USER from list.	3. System shows the current access right of the user and available access right that not assign to the user.
4. Selects access right form list.	
5. Presses Assign button.	System shows the success message.
<b>Alternative course of events</b>	

**Table 62: Expanded Use Cases 8 - Withdraw User Role**

<b>Use Case 8</b>	<b>Withdraw User Role</b>
<b>Goal in Context</b>	To allow admin to withdraw Role from the user.
<b>Primary Actor</b> <b>Secondary Actor</b>	admin
<b>Trigger</b>	Admin presses the access control management menu.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when admin wishes to withdraw role from the user.	
2. User selects user from list	3. System shows the current access right of the user and available access right that not assign to the user.
4. Selects access right given.	
5. Presses "WITHDRAW" button.	System shows success message.
<b>Alternative course of events</b>	

**Table 63: Expanded Use Cases 9 - Create New Department**

<b>Use Case 9</b>	<b>Create New Department</b>	
<b>Goal in Context</b>	New department information is correctly entered into the system.	
<b>Primary Actor</b> <b>Secondary Actor</b>	HR Admin	
<b>Main Course Description</b>	<b>Step</b>	<b>Action</b>
	1	This use case begins when an HR Admin receives information about a new department.
	2	The HR Admin enters the new department name.
<b>Alternative Course Description</b>	3	A unique department ID number is generated by the system.
	<b>Step</b>	<b>Branching Action</b>
Notes		

**Table 64: Expanded Use Cases 10 - Update Department Information**

<b>Use Case 10</b>	<b>Update Department Information</b>	
<b>Goal in Context</b>	To update an existing department information.	
<b>Primary Actor</b> <b>Secondary Actor</b>	HR Admin	
<b>Main Course Description</b>	<b>Step</b>	<b>Action</b>
	1	This use case begins when an HR Admin receives amendment to information of existing department.
	2	The HR Admin alters the department's existing information as per the amendment.
<b>Alternative Course Description</b>	<b>Step</b>	<b>Branching Action</b>

**Table 65: Expanded Use Cases 11 - Delete Department**

<b>Use Case 11</b>	<b>Delete Department</b>	
<b>Goal in Context</b>	To delete an existing department.	
<b>Primary Actor</b> <b>Secondary Actor</b>	HR Admin	
<b>Main Course Description</b>	<b>Step</b>	<b>Action</b>
	1	This use case begins when an HR Admin desires to delete an existing department.
	2	The HR Admin locates the department by department ID number.
<b>Alternative Course Description</b>	3	The HR Admin deletes the department.
		<b>Branching Action</b>

**Table 66: Expanded Use Cases 12 - Create New Employee**

<b>Use Case 12</b>	<b>Create New Employee</b>	
<b>Goal in Context</b>	New employee information is correctly entered into the system.	
<b>Primary Actor</b> <b>Secondary Actor</b>	HR Staff HR Admin	
<b>Main Course Description</b>	<b>Step</b>	<b>Action</b>
	1	This use case begins when an HR Staff receives information about a new employee.
	2	The HR Staff enters the new employee's full name, permanent address, present address, national ID card number, contact number, date of birth, gender)
<b>Alternative Course Description</b>	3	A unique employee ID number is generated by the system.
		<b>Branching Action</b>

**Table 67: Expanded Use Case 13 - Update Employee Information**

<b>Use Case 13</b>	<b>Update Employee Information</b>	
<b>Goal in Context</b>	To update an existing employee's information.	
<b>Primary Actor</b>	HR Staff	
<b>Secondary Actor</b>	HR Admin	
	<b>Step</b>	<b>Action</b>
	1	This use case begins when an HR Staff receives amendment to information of existing employee.
<b>Main Course Description</b>	2	The HR Staff alters the employee's existing information as per the amendment.
	<b>Step</b>	<b>Branching Action</b>
<b>Notes</b>		

**Table 68 : Expanded Use Case 14 - Delete Employee**

<b>Use Case 14</b>	<b>Delete Employee</b>	
<b>Goal in Context</b>	To delete an existing employee.	
<b>Primary Actor</b>	HR Admin	
<b>Secondary Actor</b>	<b>Step</b>	<b>Action</b>
	1	This use case begins when an HR Admin desires to delete an employee.
	2	The HR Admin locates the employee by employee ID number.
<b>Main Course Description</b>	3	The HR Admin deletes the employee.
	<b>Step</b>	<b>Branching Action</b>
<b>Alternative Course Description</b>		

**Table 69: Expanded Use Case 15 - Create allowance**

Use Case 15	Create allowance
<b>Goal in Context</b>	To allow HR head to add new allowance when its needed
<b>Primary Actor</b>	HR Head
<b>Secondary Actor</b>	
<b>Trigger</b>	HR staff selects allowance management interface.
<b>Typical Course of Events</b>	
Actor Action	System Response
1. This use case begins when HR head need to add new allowance to the system.	
2. HR Head opens allowance management interface	3. System shows list of allowances.
4. HR Head presses Add button.	5. System shows form with text field, name , calculation type.
6. HR head enters name of the allowance and selects Calculation type.	7. System shows amount box, dropdown menu that include type of allowance.
8-HR Head presses “CREATE” button.	9- New allowance appears in allowance list.
<b>Alternative course of events</b>	
Line 7: a- If the calculation type of allowance is in percentage, the amount field will not visible. b- If the calculation type of allowance is in amount, the percentage field will not visible.	

**Table 70: Expanded Use Case 16 - Update Allowance**

Use Case 16	Update Allowance
<b>Goal in Context</b>	To allow HR head to update allowance when it need
<b>Primary Actor</b>	HR head
<b>Secondary Actor</b>	
<b>Trigger</b>	HR staff selects allowance management interface.
<b>Typical Course of Events</b>	
Actor Action	System Response
1. This use case begins when HR head need to Update allowance information.	
2. HR Head opens allowance management interface.	3. System shows list of allowances.
4. HR Head selects allowance from list.	5. “EDIT” button enables.
6. HR Head presses edit button.	7- System shows selected allowance information in editable format.
8-HR Head changes the allowance information in editable text field, allowance name. , percentage, amount, calculated type and type.	

9. HR Head presses the update button.	10. System shows success message.
<b>Alternative course of events</b>	
Line 7: a- If the calculation type of allowance is in percentage, the amount field is not visible. b- If the calculation type of allowance is in amount, the percentage field is not visible.	
Line 9- a-if the required field is not filled, the system indicates error.	

**Table 71: Expanded Use Case 17 - Delete Allowance**

Use Case 17	Delete Allowance
<b>Goal in Context</b>	To allow HR head to Delete the allowance from the system.
<b>Primary Actor</b>	HR head
<b>Secondary Actor</b>	
<b>Trigger</b>	HR staff select allowance management interface.
<b>Typical Course of Events</b>	
Actor Action	System Response
1. This use case begins when HR head need to delete allowance information.	
2. HR Head opens allowance management interface.	3. System will show list of allowances.
4. HR Head selects allowance from list.	5. The delete button enables.
6. HR Head press delete button.	7- System shows confirm message with two buttons, OK, CANCEL.
8- HR Head presses “OK” button.	9- System deletes selected allowance.
10-HR Head presses “CANCEL” button.	10. System stops the process.
<b>Alternative course of events</b>	
Line 9: a-if the selected allowance is assign to the employee, system indicates error message.	

**Table 72: Expanded Use Case 18 - Assign Allowance**

Use Case 18	Assign Allowance
<b>Goal in Context</b>	To allow HR staff to assign special leave to employees.
<b>Primary Actor</b>	HR staff
<b>Secondary Actor</b>	HR head
<b>Trigger</b>	HR staff selects staff form allowance management interface.
<b>Typical Course of Events</b>	
Actor Action	System Response
1. This use case begins when HR staff needs to assign special allowance to the staff.	
2. HR Staff selects employee allowance management interface.	3. System shows list of active employees.
4. HR Staff selects desire employee.	
5- HR staff press assign allowance button.	6- System shows a pop-up box with a list of allowance that has not been assigned to the selected employee.
7- HR Staff presses Proceed button.	8-System assigns selected allowance to the selected employee and closes the pop-up box.
<b>Alternative course of events</b>	
Line 6: If there is no special allowance for the selected employee, then system indicates error.	

**Table 73: Expanded Use Case 19 - Create New Leave**

Use Case 19	Create New Leave	
Goal in Context	New leave information is correctly entered into the system.	
Primary Actor	HR Admin	
Secondary Actor		
Main Course Description	Step	Action
	1	This use case begins when an HR Admin receives new leave information.
	2	The HR Admin categorizes the leave type (general/special).
	3	The HR Admin enters the new leave name, no. of days.
	4	A unique leave ID number is generated by the system.
Alternative Course Description	Step	Branching Action

**Table 74: Expanded Use Case 20 - Delete Leave**

<b>Use Case 20</b>	<b>Delete Leave</b>	
<b>Goal in Context</b>	To delete an existing leave.	
<b>Primary Actor</b>	HR Admin	
<b>Main Course Description</b>	<b>Step</b>	<b>Action</b>
	1	This use case begins when an HR Admin wishes to delete an existing leave.
	2	The HR Admin alters the existing leave information as per the amendment.
<b>Alternative Course Description</b>	<b>Step</b>	<b>Branching Action</b>
<b>Notes</b>		

**Table 75: Expanded Use Case 21 - Update Leave Information**

<b>Use Case 21</b>	<b>Update Leave Information</b>	
<b>Goal in Context</b>	To update an existing leave information.	
<b>Primary Actor</b>	HR Admin	
<b>Main Course Description</b>	<b>Step</b>	<b>Action</b>
	1	This use case begins when an HR Admin receives amendment to information of existing leave.
	2	The HR Admin alters the existing leave information as per the amendment.
<b>Alternative Course Description</b>	<b>Step</b>	<b>Branching Action</b>
<b>Notes</b>		

**Table 76: Expanded Use Case 22 - Approve/Decline Leave**

<b>Use Case 22</b>	<b>Approve/Decline Leave</b>	
<b>Goal in Context</b>	To approve/decline a requested leave.	
<b>Primary Actor</b> <b>Secondary Actor</b>	Head of Organization	
<b>Main Course Description</b>	<b>Step</b>	<b>Action</b>
	1	This use case begins when Head of Organization desires to approve/decline a requested leave.
	2	The Head of Organization approves/declines the requested leave.
<b>Alternative Course Description</b>	<b>Step</b>	<b>Branching Action</b>

**Table 77 : Expanded Use Cases 23 - Requests for Leave**

<b>Use Case 23</b>	<b>Request for Leave</b>
<b>Goal in Context</b>	To allow an Employee to request leave from the HRMS.
<b>Primary Actor</b> <b>Secondary Actor</b>	Employee
<b>Trigger</b>	Employee login to the HRMS portal and opens leave request interface.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when the employee open leave request interface.	2. Requests for filling employee leave request form.
3. The Employee provides leave type, start date, end date and reason for leave.	4. Verifies employee leave balance.
5. Employee presses Request button.	6. The request is saved to the database.
<b>Alternative course of events</b>	
Line 3: If the request duration is too long the then indicates error.	

**Table 78: Expanded Use Case 24 - Amend Leave Request**

Use Case 24	Amend Leave Request
<b>Goal in Context</b>	To allow Head of Department to amend Leave request of the employee.
<b>Primary Actor</b>	Head of organization
<b>Secondary Actor</b>	Head of Department
<b>Trigger</b>	Opens leave management interface.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when the Head of organization opens employee leave management interface.	2. System shows list of all the requested leave.
3. Head of department selects leave request from list.	4. System shows select leave request in edit mode.
5. Head of organization changes its information, start date, end date, the reason for the amendment.	6. Verifies input data.
7- Head of organization Press update button.	8- Selected leave request is updated in the database.
<b>Alternative course of events</b>	
Line 5: If the duration is too long the then indicates error.	

**Table 79: Expanded Use Case 25 - Check-in**

Use Case 25	Check-in
<b>Goal in Context</b>	Add finger print data when check in
<b>Primary Actor</b>	Staff, HR Staff, HR Admin
<b>Secondary Actor</b>	
<b>Trigger</b>	When staff wants to check-in.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. Staff enters finger print.	Employee check-in time and date is recorded in the system log.
<b>Alternative Course of Events</b>	
Line 1a: If staff is not in the database, reject message appears.	

**Table 80: Expanded Use Case 26 - Update Attendance**

<b>Use Case 26</b>	<b>Update Attendance</b>
<b>Goal in Context</b>	Update attendance when Staff requests for amend.
<b>Primary Actor</b>	Staff
<b>Secondary Actor</b>	HR Staff, HR Admin
<b>Trigger</b>	When Staff requests for amendment.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. Staff requests for amendment.	Request is seen by the HR Staff and HR Admin Staff.
2. HR staff or HR Admin Staff verifies the amendment.	Attendance log is updated.
<b>Alternative Course of Events</b>	
Line 2a: If amend request is wrong and the data is allow for edit then edit and correct the request and amend.	
Line 2b: If amend request has incorrect data send back for correct request.	
Line 2c: If request is wrong HR staff or HR Admin can reject the request.	

**Table 81: Expanded Use Case 27 - View Attendance**

<b>Use Case 27</b>	<b>View Attendance</b>
<b>Goal in Context</b>	View the attendance
<b>Primary Actor</b>	Staff, HR Staff, HR Admin
<b>Secondary Actor</b>	-
<b>Trigger</b>	When Staff wants to view the attendance.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. Staff view the attendance	Attendance is viewed.
<b>Alternative Course of Events</b>	
Line 2a: HR Staff and HR Admin has authority to view all the employees attendance.	

**Table 82: Expanded Use Case 28 - Create designation**

<b>Use Case 28</b>	<b>Create designation</b>
<b>Goal in Context</b>	To allow HR head to add new job type or designation to the system.
<b>Primary Actor</b>	HR head
<b>Secondary Actor</b>	
<b>Trigger</b>	HR head add new designation button.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Head presses the button "Add New Designation".	2. System will show popup window that include three input field (name, rank and basic salary).
3. User fills the input form.	

4. User presses the submit button.	5. System shows the success message.
<b>Alternative course of events</b>	
Line 3: a- If the user does not fill the required field then system indicate error. c- If the basic salary is too big then system indicates error.	

**Table 83: Expanded Use Case 29 - Update Designation**

Use Case 29	Update Designation
<b>Goal in Context</b>	To allow HR Head to update designation information.
<b>Primary Actor</b>	HR head
<b>Secondary Actor</b>	
<b>Trigger</b>	HR head selects designation from list.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Head Select designation form list	2. The “EDIT” button enables.
3. Presses “EDIT” button.	4. System shows designation information is editable format.
4. User change current information (name, rank and basic salary).	
Presses “UPDATE” button.	System shows success message.
<b>Alternative course of events</b>	
Line 3: a-If required field are not filled, system shows the error message	

**Table 84: Expanded Use Case 30 - Delete Designation**

Use Case 30	Delete Designation
<b>Goal in Context</b>	To allow HR head to delete job type or designation
<b>Primary Actor</b>	HR head
<b>Secondary Actor</b>	
<b>Trigger</b>	HR head selects designation from list.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Head Select designation form list.	2. “DELETE” button enables.
3. Presses delete button.	4. Confirmation message appears with two options (OK, Cancel).
4. User presses OK.	5. System shows the success message
<b>Alternative course of events</b>	
Line 3: a- If the selected designation is assign to one or more staff, the system shows error.	

**Table 85: Expanded Use Case 31 - Payroll Process**

<b>Use Case 31</b>	<b>Payroll Process</b>
<b>Goal in Context</b>	To generate pay slip.
<b>Primary Actor</b>	HR Staff, HR admin
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff selects payroll process interface.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff wish to generate pay slip in end of the month.	
2. User selects employee form list.	3. System shows the timesheet and allowance of the employee.
4. Press payroll process button.	System generates the pays slip.
<b>Alternative course of events</b>	
Line -4. if the employee timesheet is not filled, system indicate error	

**Table 86: Expanded Use Case 32 - Update Attendance**

<b>Use Case 32</b>	<b>Update Attendance</b>
<b>Goal in Context</b>	To allow HR Staff to Update Attendance information.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff selects Attendance from Attendance list by using user id.
<b>Typical Course of Events</b>	
<b>Action</b>	<b>System Response</b>
1. This use case begins when HR Staff select Attendance form Attendance list by using user id.	2. Edit button enables.
3. Press “EDIT” button.	4. System shows user information is editable format.
4. User change current information (date, time, status).	
Press “UPDATE” button.	System shows the success message.
<b>Alternative course of events</b>	
Line 4: a-if required field are not fill, system shows the error message.	

**Table 87: Expanded Use Case 33 - Assign Employee to the Department**

<b>Use Case 33</b>	<b>Assign Employee to the Department</b>
<b>Goal in Context</b>	To allow HR Staff to add Employee to the Department.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff presses the Department from the menu.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff wishes to add new Employee to the Department.	
2. User selects Employee from user list.	3. System shows the current Department of the Employee and available Department that not assign to the Employee.
4. Selects Department form list.	
5. Presses Assign button.	System shows the success message.
<b>Alternative course of events</b>	

**Table 88: Expanded Use Case 34 - Withdraw Leave**

<b>Use Case 34</b>	<b>Withdraw Leave</b>
<b>Goal in Context</b>	To allow HR Staff to withdraw leaves from the user.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff presses the Leave in menu.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff wishes to withdraw Leave from the user.	
2. User selects user from list	3. System shows the current Leave of the user and available Leave that not assign to the user.
4. Selects Leave given.	
5. Presses "WITHDRAW" button.	System shows success message.
<b>Alternative course of events</b>	

**Table 89: Expanded Use Case 35 - Assign Leave**

Use Case 35	Assign Leave
<b>Goal in Context</b>	To allow HR Staff to add leaves to the Employee.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff presses the Add Leave from the menu.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff wishes to add Leave to the Employee.	
2. User selects Employee from user list.	3. System shows the current Leave and available Leave that not assign to the Employee.
4. Selects Leave form list.	
5. Presses Assign button.	System shows the success message.
<b>Alternative course of events</b>	

**Table 90: Expanded Use Case 36 - Update Department Information**

Use Case 36	Update Department Information
<b>Goal in Context</b>	To allow HR Admin to update Department information.
<b>Primary Actor</b>	HR Admin
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Admin selects Department from Department list.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Admin select Department form Department list.	2. Edit button enables.
3. Press “EDIT” button.	4. System shows user information is editable format.
4. User change current information (name, detail).	
Press “UPDATE” button.	System shows the success message.
<b>Alternative course of events</b>	
Line 4: a-if required field are not fill, system shows the error message.	

**Table 91: Expanded Use Case 37 - Update Calendar Event**

Use Case 37	Update Calendar Event
<b>Goal in Context</b>	To allow HR Admin to update Calendar Event information.
<b>Primary Actor</b>	HR Admin
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Admin selects Calendar Event from Calendar Event list.
<b>Typical Course of Events</b>	
Actor Action	System Response
1. This use case begins when HR Admin select Calendar Event form Calendar Event list.	2. Edit button enables.
3. Press “EDIT” button.	4. System shows user information is editable format.
4. User change current information (name, date, detail).	
Press “UPDATE” button.	System shows the success message.
<b>Alternative course of events</b>	
Line 4: a-if required field are not fill, system shows the error message.	

**Table 92: Expanded Use Case 38 - Delete Calendar Event**

Use Case 38	Delete Calendar Event
<b>Goal in Context</b>	To allow HR Staff to remove Calendar Event from the system when needed.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff searches the Calendar Event.
<b>Typical Course of Events</b>	
Actor Action	System Response
1. This use case begins when HR Staff search the Calendar Event.	2. The system will show the Calendar Event.
3. Press delete button.	4. Confirmation message will appear with two options (OK, Cancel).
4. User presses OK.	5. System removes Calendar Event from the system.
<b>Alternative course of events</b>	

**Table 93: Expanded Use Case 39 - Create User group**

<b>Use Case 39</b>	<b>Create User group</b>
<b>Goal in Context</b>	To allow HR Staff add new User group to the system.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	Admin presses “Create New User group” button.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff wishes to create a new user group.	
2. User presses the “Create New User group” button.	3. System shows the input form that includes name and detail.
4. User presses the “SUBMIT” button.	5. System creates a new user group.
<b>Alternative course of events</b>	
Line 4: a- If group already exists in the system, an error message will be displayed. b- If group id already exists in the system, an error message will be displayed.	

**Table 94: Expanded Use Case 40 - Delete User group**

<b>Use Case 40</b>	<b>Delete User group</b>
<b>Goal in Context</b>	To allow HR Staff to remove User group from the system when needed.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff searches the User group.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff search the User group.	2. The system will show the User group.
3. Press delete button.	4. Confirmation message will appear with two option (OK,Cancel).
4. User presses OK.	5. System removes User group from the system.
<b>Alternative course of events</b>	

**Table 95: Expanded Use Case 41 - Create Shift**

<b>Use Case 41</b>	<b>Create Shift</b>
<b>Goal in Context</b>	To allow HR Staff adds new Shift to the system.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff presses “Create New Shift” button.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff wishes to create a new Shift.	
2. User presses the “Create New Shift” button.	3. System shows the input form that includes name, in and out.
4. User presses the “SUBMIT” button.	5. System creates a new Shift.
<b>Alternative course of events</b>	
Line 4: a- If Shift already exists in the system, an error message will be displayed. b- If Shift id already exists in the system, an error message will be displayed.	

**Table 96: Expanded Use Case 42 - Delete Shift**

<b>Use Case 42</b>	<b>Delete Shift</b>
<b>Goal in Context</b>	To allow HR Staff to remove Shift from the system when needed.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff searches the Shift.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff search the Shift.	2. The system will show the Shift.
3. Press delete button.	4. Confirmation message will appear with two option (OK,Cancel).
4. User presses OK.	5. System removes Shift from the system.
<b>Alternative course of events</b>	

**Table 97: Expanded Use Case 43 - Update Shift**

<b>Use Case 43</b>	<b>Update Shift</b>
<b>Goal in Context</b>	To allow HR Staff to update Shift information.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff selects Shift from Shift list.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff select Shift form Shift list.	2. Edit button enables.
3. Press “EDIT” button.	4. System shows user information is editable format.
4. User change current information (name, in, out).	
Press “UPDATE” button.	System shows the success message.
<b>Alternative course of events</b>	
Line 4: a-if required field are not fill, system shows the error message.	

**Table 98: Expanded Use Case 44 - Update Special task**

<b>Use Case 44</b>	<b>Update Special task</b>
<b>Goal in Context</b>	To allow HR Staff to update Special task information.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff selects Special task from Special task list.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff select Special task form Special task list.	2. Edit button enables.
3. Press “EDIT” button.	4. System shows user information is editable format.
4. User change current information (name, date, detail, rate).	
Press “UPDATE” button.	System shows the success message.
<b>Alternative course of events</b>	
Line 4: a-if required field are not fill, system shows the error message.	

**Table 99: Expanded Use Case 45 - Delete task**

<b>Use Case 45</b>	<b>Delete task</b>
<b>Goal in Context</b>	To allow HR Staff to remove task from the system when needed.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff searches the task.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff search the task.	2. The system will show the task.
3. Press delete button.	4. Confirmation message will appear with two option (OK, Cancel).
4. User presses OK.	5. System removes task from the system.
<b>Alternative course of events</b>	

**Table 100: Expanded Use Case 46 - Assign employee to special task Employee**

<b>Use Case 46</b>	<b>Assign employee to special task Employee</b>
<b>Goal in Context</b>	To allow HR Staff to add Special task to the Employee.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff presses the Special task from the menu.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff wishes to add Special task to the Employee.	
2. User selects Employee from user list.	3. System shows the Special task assign to the Employee.
4. Selects Special task form list.	
5. Presses Assign button.	System shows the success message.
<b>Alternative course of events</b>	

**Table 101: Expanded Use Case 47 - Withdraw Employee**

<b>Use Case 47</b>	<b>Withdraw Employee</b>
<b>Goal in Context</b>	To allow HR Staff to withdraw user from the task.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff presses the task in menu.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff wishes to withdraw task from the user.	
2. User selects user from list	3. System shows the current task of the user and available task that not assign to the user.
4. Selects task given.	
5. Presses “WITHDRAW” button.	System shows success message.
<b>Alternative course of events</b>	

**Table 102: Expanded Use Case 48 - Create Special task**

<b>Use Case 48</b>	<b>Create Special task</b>
<b>Goal in Context</b>	To allow HR Staff add new special task to the system.
<b>Primary Actor</b>	HR Staff
<b>Secondary Actor</b>	
<b>Trigger</b>	HR Staff presses “Create special task” button.
<b>Typical Course of Events</b>	
<b>Actor Action</b>	<b>System Response</b>
1. This use case begins when HR Staff wishes to create a new special task.	
2. User presses the “Create special task” button.	3. System shows the input form that includes employee id, username, date, detail and rate.
4. User presses the “SUBMIT” button.	5. System creates a new special task.
<b>Alternative course of events</b>	
Line 4: a- If special task already exists in the system, an error message will be displayed. b- If special task id already exists in the system, an error message will be displayed.	

# CHAPTER 4: SYSTEM DESIGN

## 4.1. Introduction

This document proposes the detailed system design for the Human Resource Management System created for Maldives Stock Exchange. In this document, several standard notations are used to illustrate the entire system in great depth.

Unified Modelling Language (UML) notations are primarily used, since it is a widespread standard among object-oriented developers. Storyboards are used for User Interface demonstration.

This design addresses all the requirements stated in the “Requirements Definition and Specification” document. Implementing this design will result a system that can meet the needs of the client, MSE.

## 4.2. Database Design

The major elements of a system are data flow, data stores, processes, and entities. The Data Dictionary describes all these elements of a system. It is an electronic glossary of items. It defines each element encountered during the analysis and design of a new system (Jibitesh Mishra, Ashok Mohanty, 2011).

### 4.2.1. System Data Dictionary

Table 103: Table Structure for Table Access groups

Column	Type	Null	Default
<i>id</i>	int(11)	No	
accessGroupeId	int(11)	No	
ResourcesId	int(11)	No	
rights	varchar(20)	No	

**Table 104**Table structure for table allowance

Column	Type	Null	Default
<i>id</i>	int(11)	No	
name	text	No	
amount	double	No	
calculationType	text	No	
types	text	No	

**Table 105**Table structure for table calendar

Column	Type	Null	Default
<i>id</i>	int(11)	No	
date	date	No	
Details	text	No	
type	varchar(255)	No	

**Table 106**Table structure for table department

Column	Type	Null	Default
<i>id</i>	int(11)	No	
name	text	No	

**Table 107**Table structure for table designation

Column	Type	Null	Default
<i>id</i>	int(11)	No	
name	text	No	
rank	text	No	
basicSalary	double	No	

**Table 108**Table structure for table employee

Column	Type	Null	Default
<i>id</i>	int(11)	No	
fullName	varchar(400)	No	
permanentAddress	varchar(400)	No	
joinDate	date	No	
CurrentAddress	text	No	
idCardNumber	varchar(10)	No	
gender	varchar(7)	No	
mobileNumber	varchar(10)	No	
dateOfBirth	date	No	
emailAddress	varchar(30)	No	
designationId	int(11)	No	

shift_id	int(11)	No	
----------	---------	----	--

**Table 109**Table structure for table employeeallowance

Column	Type	Null	Default
<i>id</i>	int(11)	No	
emp_id	int(11)	No	
allowanace_id	int(11)	No	

**Table 110**Table structure for table employee leave

Column	Type	Null	Default
<i>id</i>	int(11)	No	
employee_id	int(50)	No	
leave_id	int(50)	No	
StartDate	date	No	
user_id	int(50)	No	
endDate	date	No	
reason	text	No	
requestDate	varchar(50)	No	
status	varchar(255)	No	

**Table 111**Table structure for table leavlist

Column	Type	Null	Default
<i>id</i>	int(11)	No	
nameOfLeave	text	No	
NoOfDays	int(11)	No	
types	varchar(255)	No	

**Table 112**Table structure for table resources

Column	Type	Null	Default
<i>id</i>	int(11)	No	
filename	varchar(255)	No	
description	text	No	

**Table 113**Table structure for table shift

Column	Type	Null	Default
<i>id</i>	int(11)	No	
name	text	No	
in	time	No	
out	time	No	

**Table 114**Table structure for table special task

Column	Type	Null	Default
<i>id</i>	int(11)	No	
name	text	No	
date	date	No	
details	text	No	
rate	int(11)	No	

**Table 115**Table structure for table special task emp

Column	Type	Null	Default
<i>id</i>	int(11)	No	
task_id	int(11)	No	
emp_id	int(11)	No	

**Table 116**Table structure for table timesheet

Column	Type	Null	Default
<i>id</i>	int(11)	No	
emp_id	int(11)	No	
date	date	No	
time	datetime	No	
state	varchar(20)	No	

**Table 117**Table structure for table usergroup

Column	Type	Null	Default
<i>id</i>	int(11)	No	
name	varchar(255)	No	
details	text	No	

**Table 118**Table structure for table users

Column	Type	Null	Default
<i>id</i>	int(11)	No	
username	varchar(50)	No	
password	varchar(40)	No	
emp_id	varchar(11)	No	
email	varchar(255)	No	
UserLevel	int(11)	No	

**Table 119** Table structure for table wages

Column	Type	Null	Default
<i>id</i>	int(11)	No	
emp_id	int(11)	No	
allowanceId	int(11)	No	
type	varchar(30)	No	
date	datetime	No	
amout	bigint(20)	No	
status	varchar(40)	No	

#### 4.2.2. Entity Relational (ER) Diagram

The Entity-Relationship diagram is widely used in structured analysis and conceptual modeling, as this approach is easy to understand, powerful to model real-world problems and readily translated into a database schema. The ER views that the real world consists of a collection of business entities, the relationships between them and the attributes used to describe them (Il-Yeol Song, Mary Evans, E.K. Park, 1995).

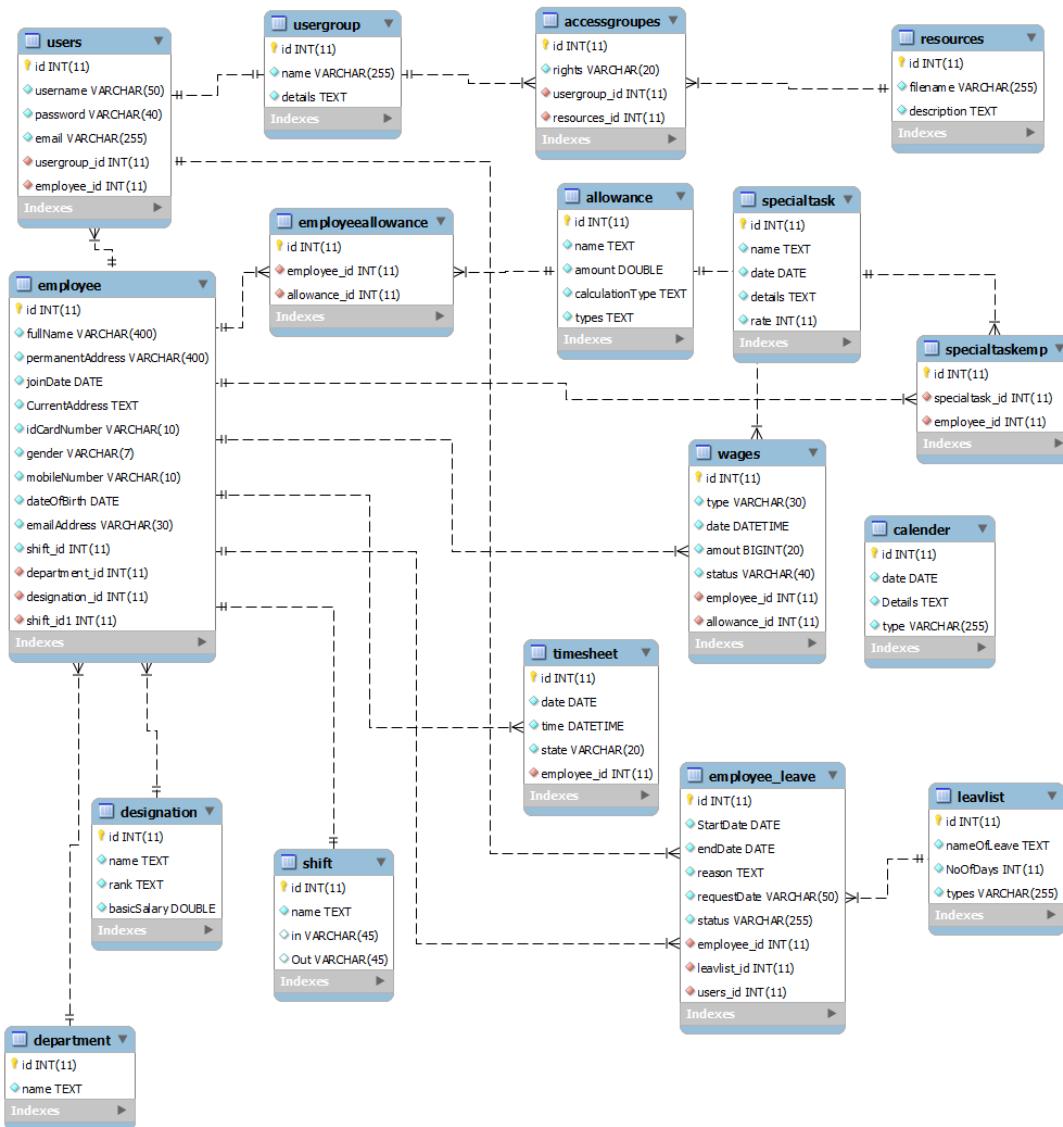


Figure 6 Entity Relational (ER) Diagram of TinyHRMS

## 4.3.Logical Design

### 4.3.1. Class Design Diagram

Class diagram shows how different entities (people, things, and data) relate to each other. In other words, it shows the static structures of the system (Bell, 2003).

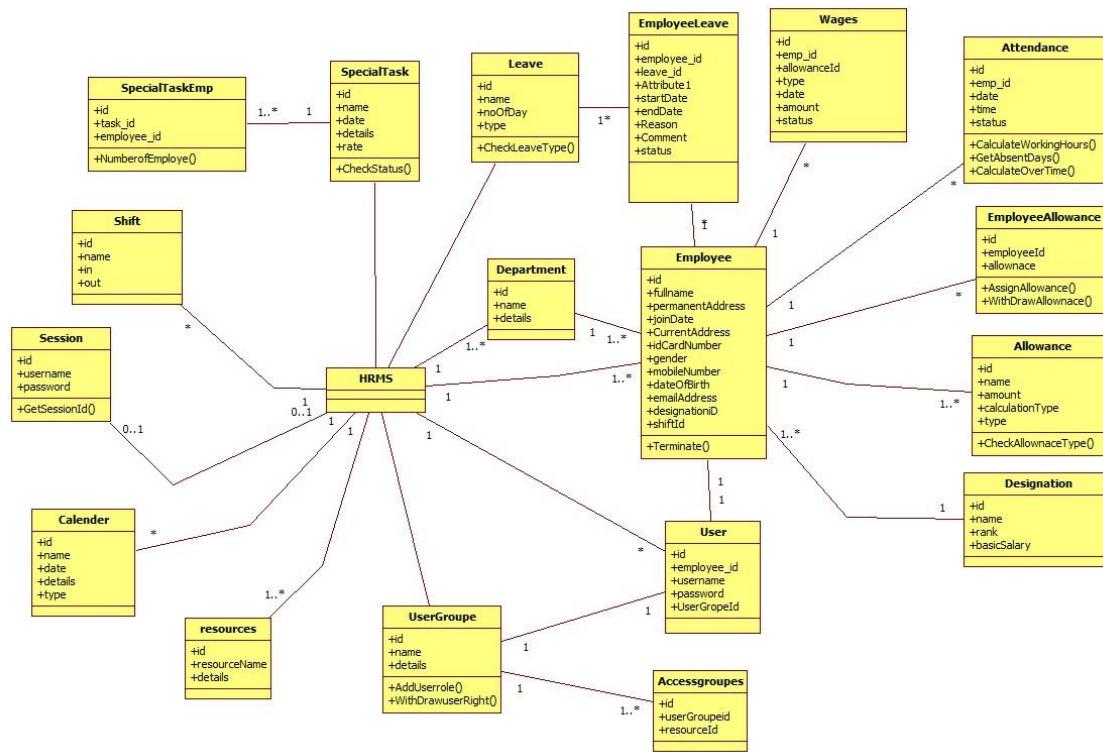


Figure 7 Class Design Diagram

### 4.3.2. Web-based Structural Design

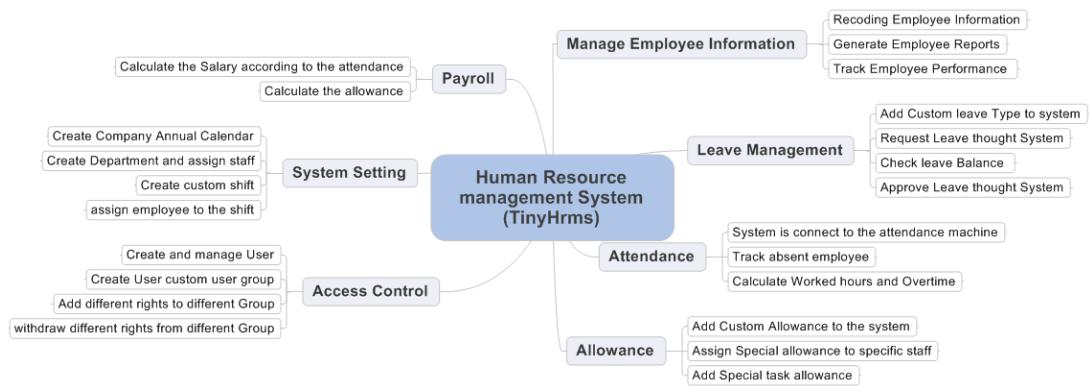


Figure 8 Web-based Structural Design

## 4.4. Dynamic Design

### 4.4.1. Sequence Diagrams

Sequence diagrams show a detailed flow for a specific use case or a part of a specific use case. They show the calls between the different objects in their sequence and different calls to different objects.

A sequence diagram has two dimensions: The vertical dimension shows the sequence of messages/calls in the time order that they occur; the horizontal dimension shows the object instances to which the messages are sent (Bell, 2003).

The following sequence diagrams show detailed flows for use cases of “TinyHRMS”.

#### Use case 1: Create User

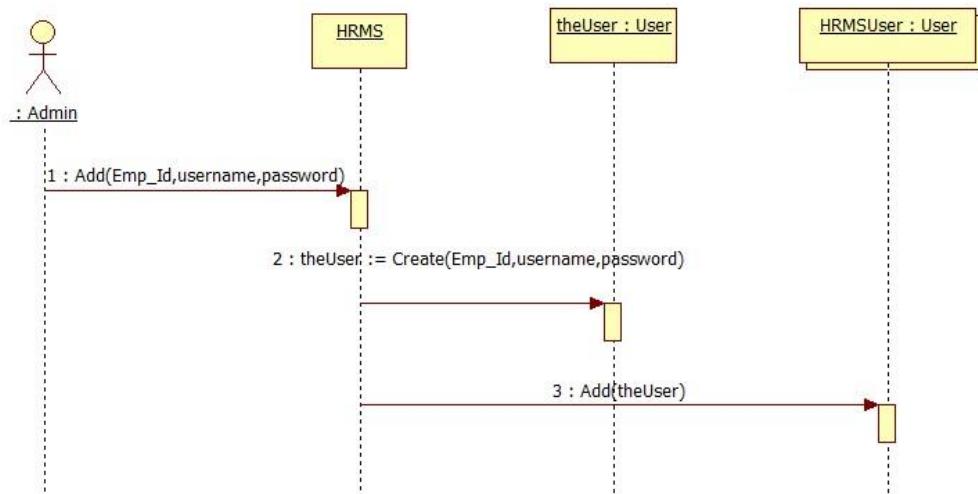


Figure 9 Sequence Diagram Create User

### Use case 2: Log-in

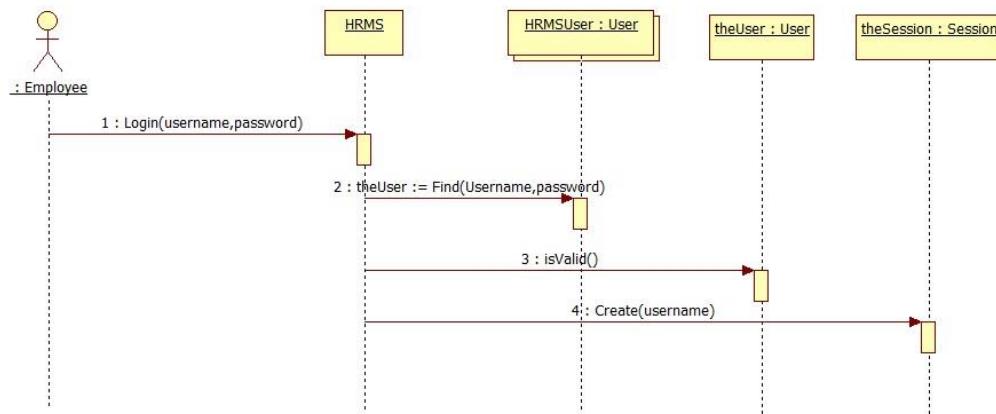


Figure 10 Sequence Diagram Log-in

### Use case 3: Forgot Password

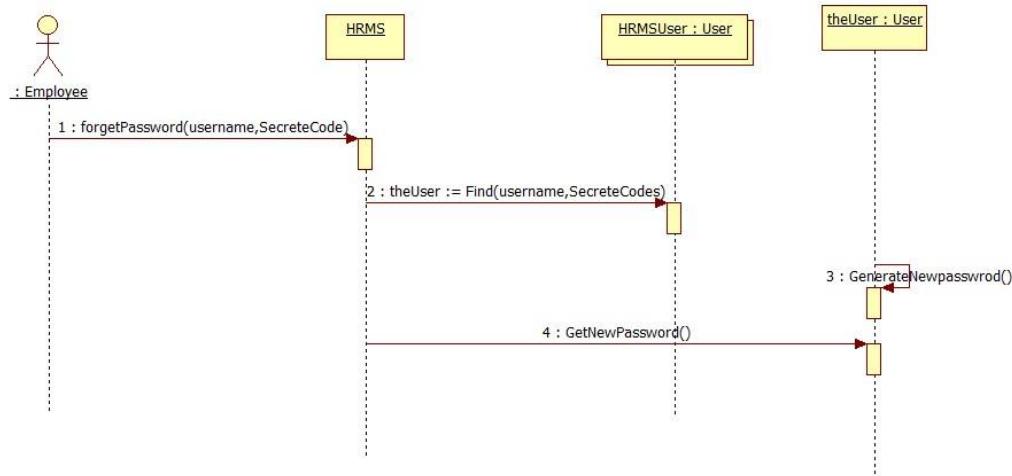


Figure 11 Sequence Diagram Forgot Password

### Use case 4: Edit Profile

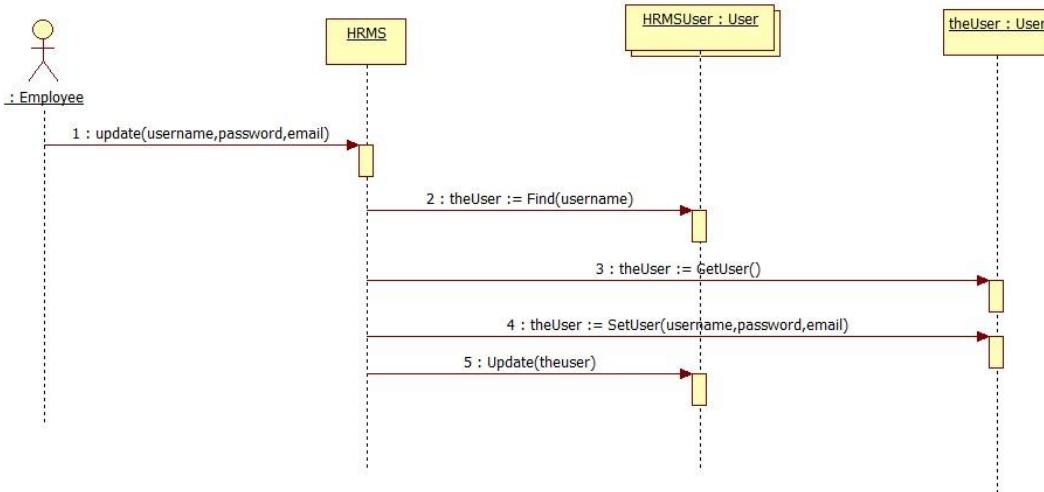


Figure 12 Sequence Diagram Edit Profile

### Use case 5: Delete User

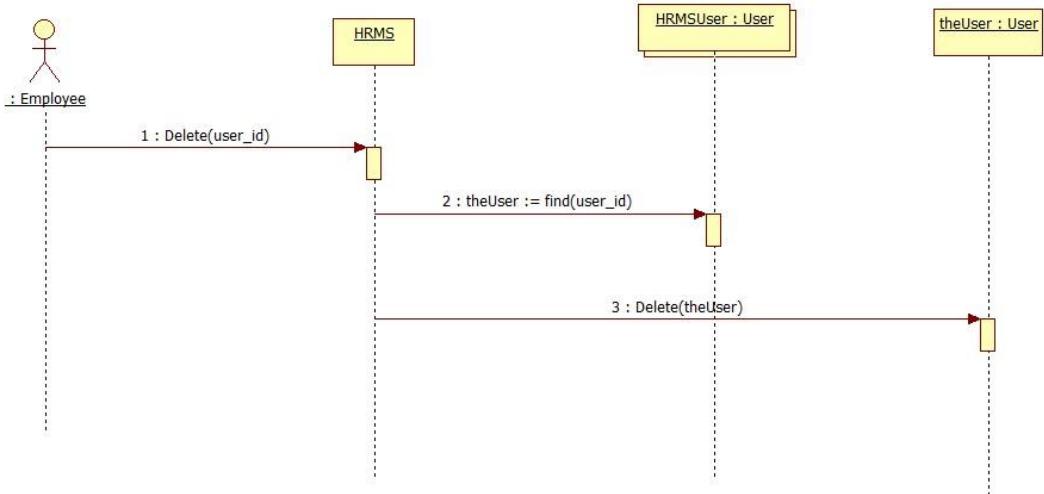


Figure 13 Sequence Diagram Delete User

### Use case 6: Update User

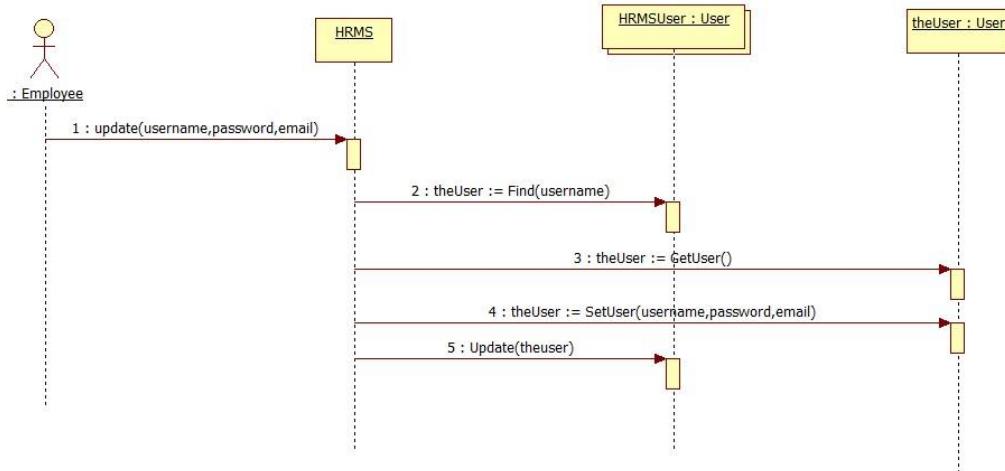


Figure 14 Sequence Diagram Update User

### Use case 7: Assign Role to the User

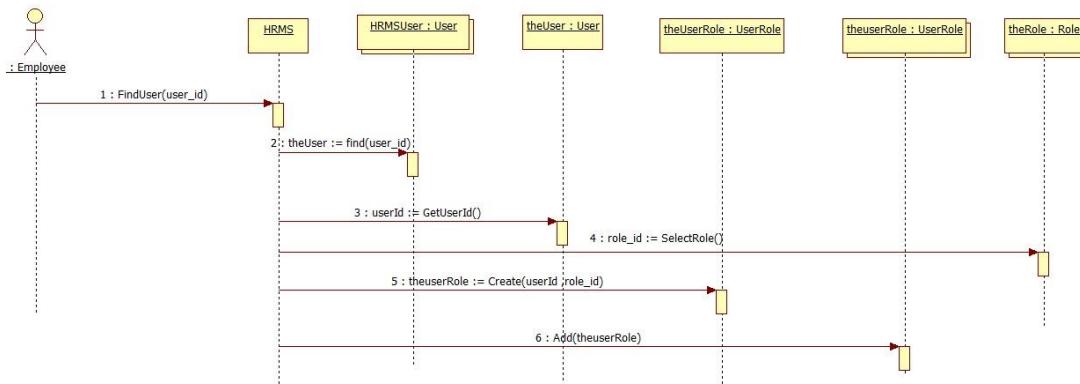


Figure 15 Sequence Diagram Assign Role to the User

### Use case 8: Withdraw User Role

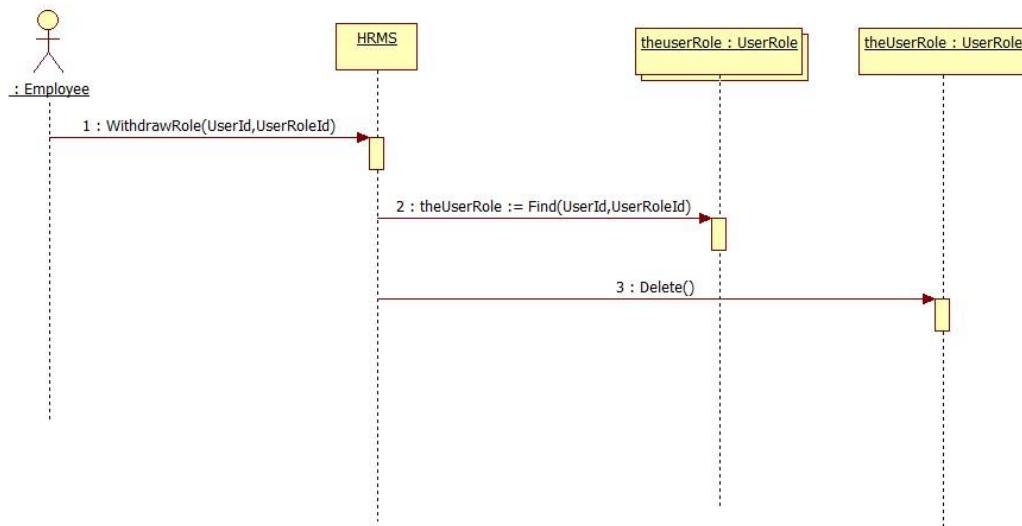


Figure 16 Sequence Diagram Withdraw User Role

### Use case 9: Create New Department

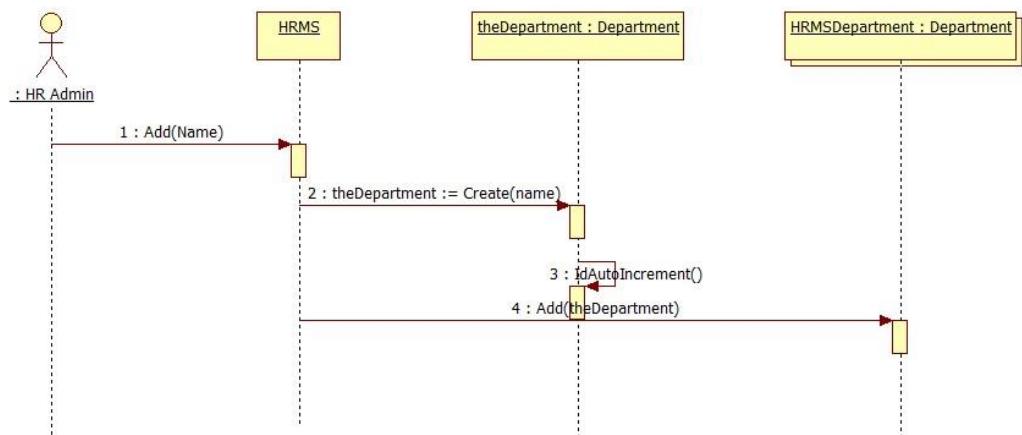


Figure 17 Sequence Diagram Create New Department

### Use case 10: Update Department Information

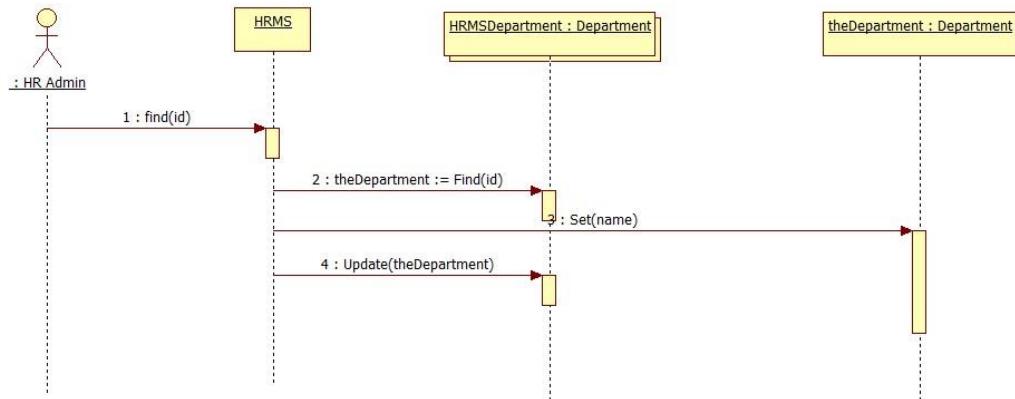


Figure 18 Sequence Diagram Update Department Information

### Use case 11: Delete Department

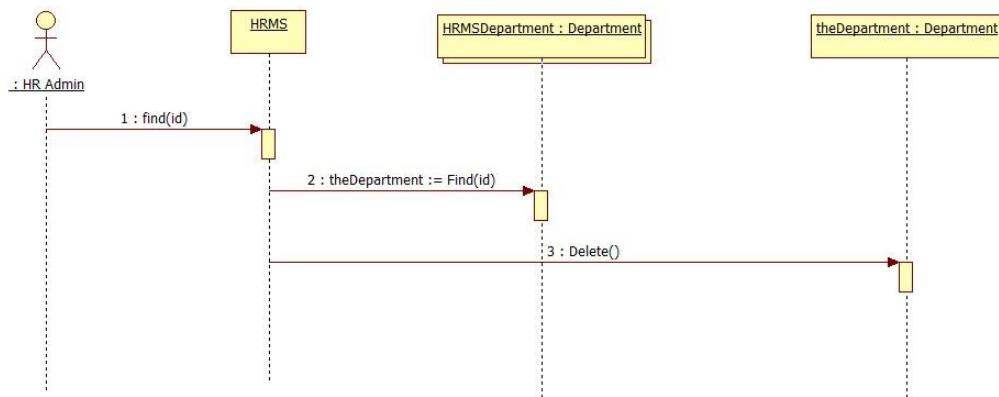


Figure 19 Sequence Diagram Delete Department

### Use case 12: Create New Employee

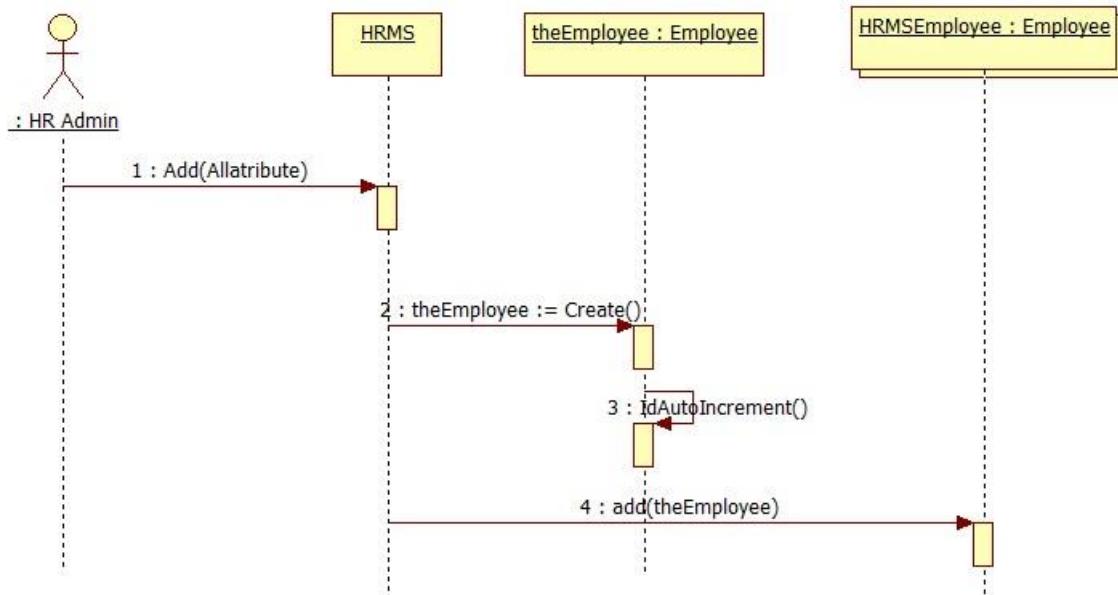


Figure 20 Sequence Diagram Create New Employee

### Use case 13: Update Employee Information

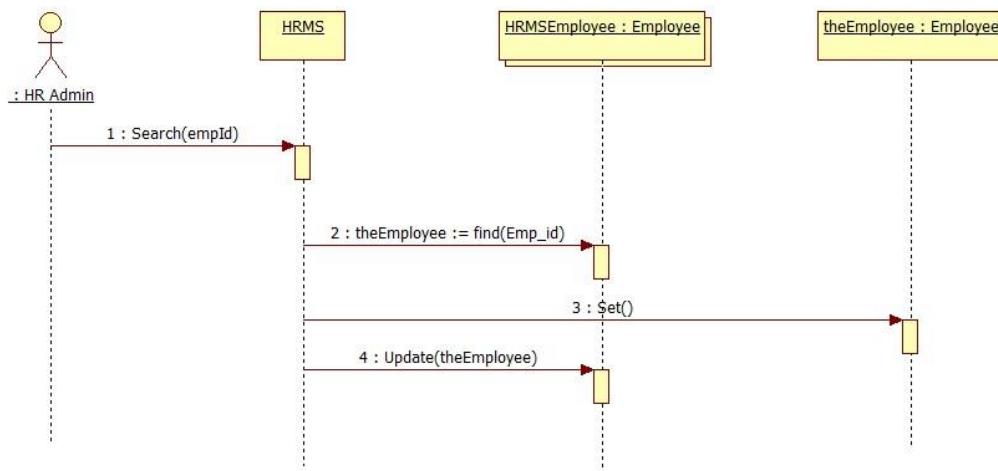


Figure 21 Sequence Diagram Update Employee Information

### Use case 14: Delete Employee

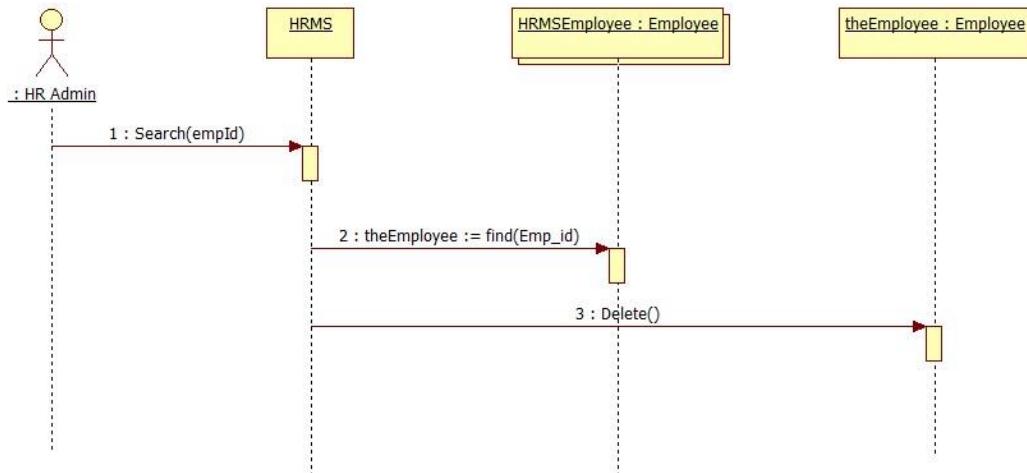


Figure 22 Sequence Diagram Delete Employee

### Use case 15: Create Allowance

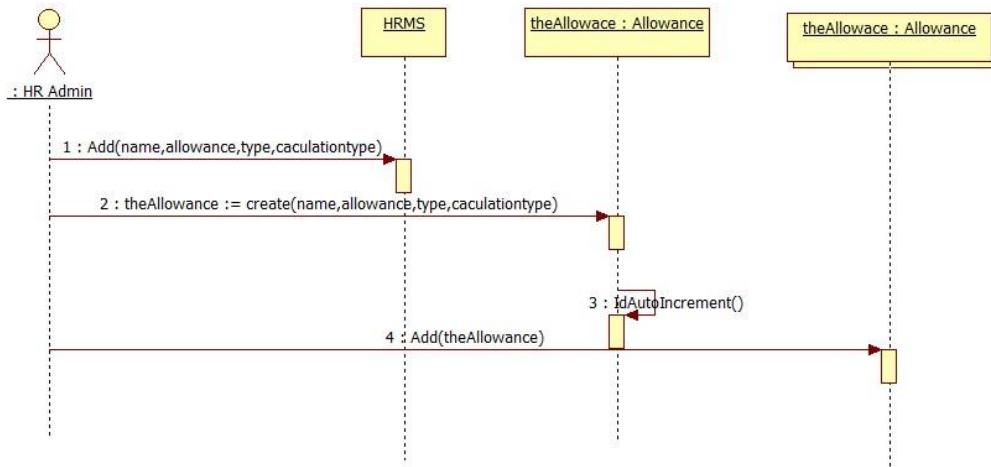


Figure 23 Sequence Diagram Create Allowance

### Use case 16: Update Allowance

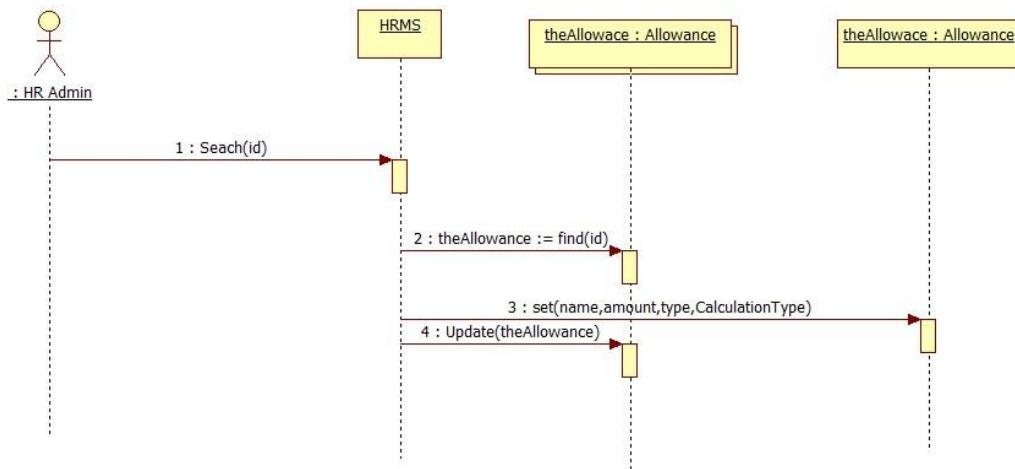


Figure 24 Sequence Diagram Update Allowance

### Use case 17: Delete Allowance

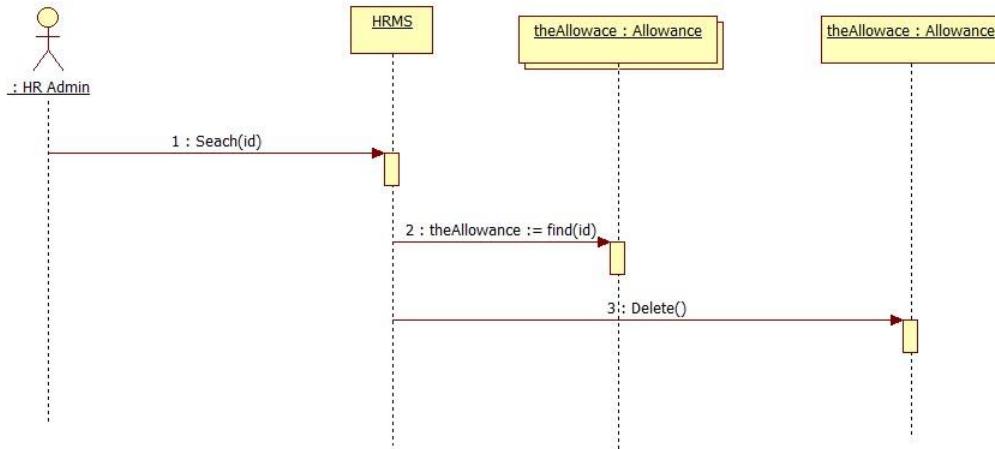


Figure 25 Sequence Diagram Delete Allowance

### Use case 18: Assign Allowance

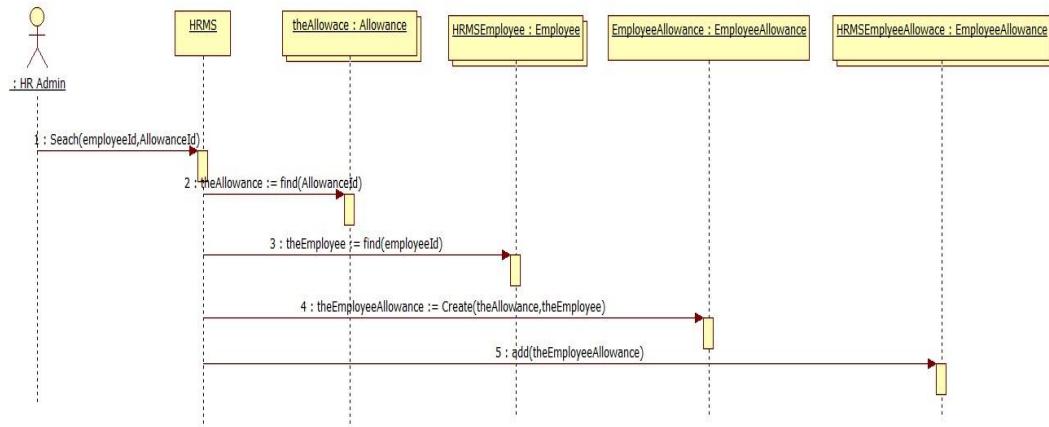


Figure 26 Sequence Diagram Assign Allowance

### Use case 19: Create New Leave

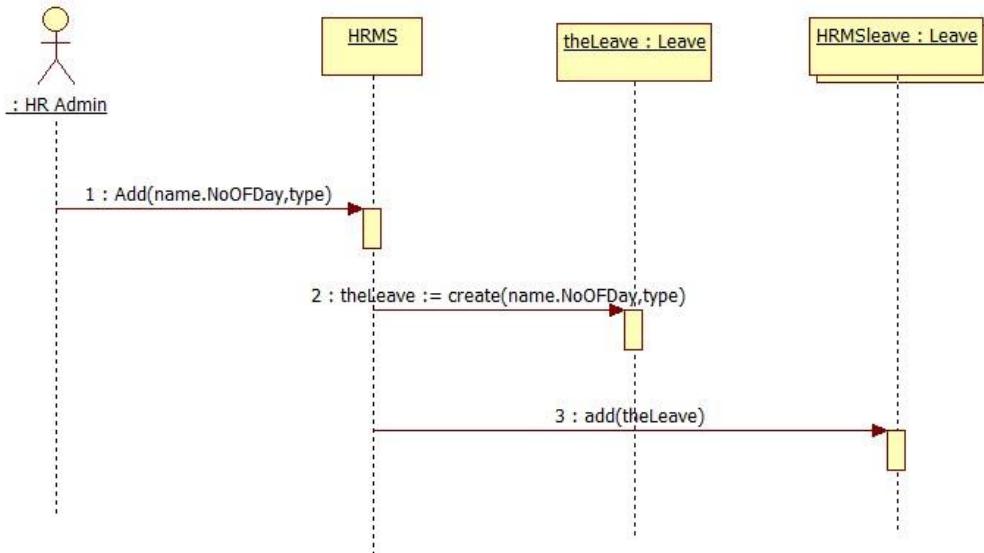


Figure 27 Sequence Diagram Create New Leave

### Use case 20: Delete Leave

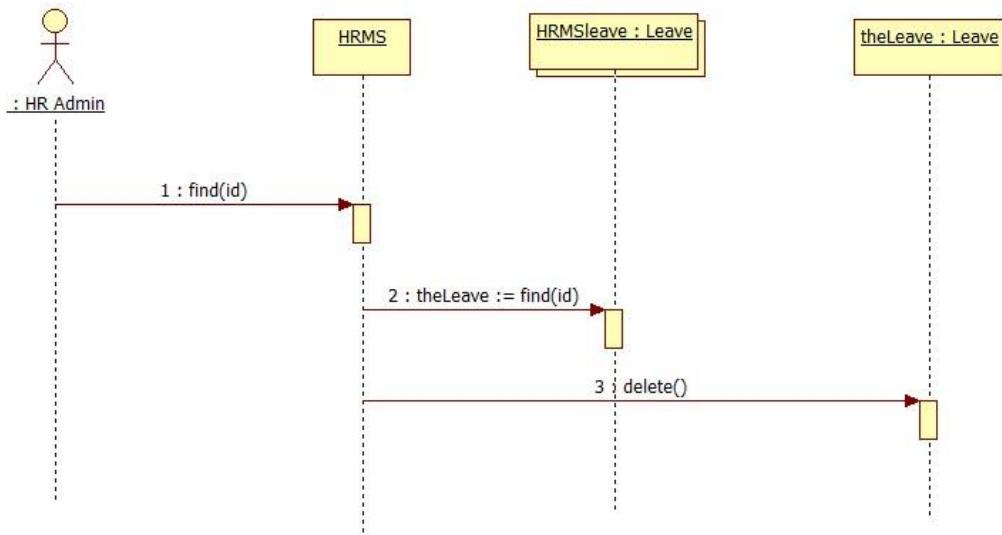


Figure 28 Sequence Diagram Delete Leave

### Use case 21: Update Leave Information

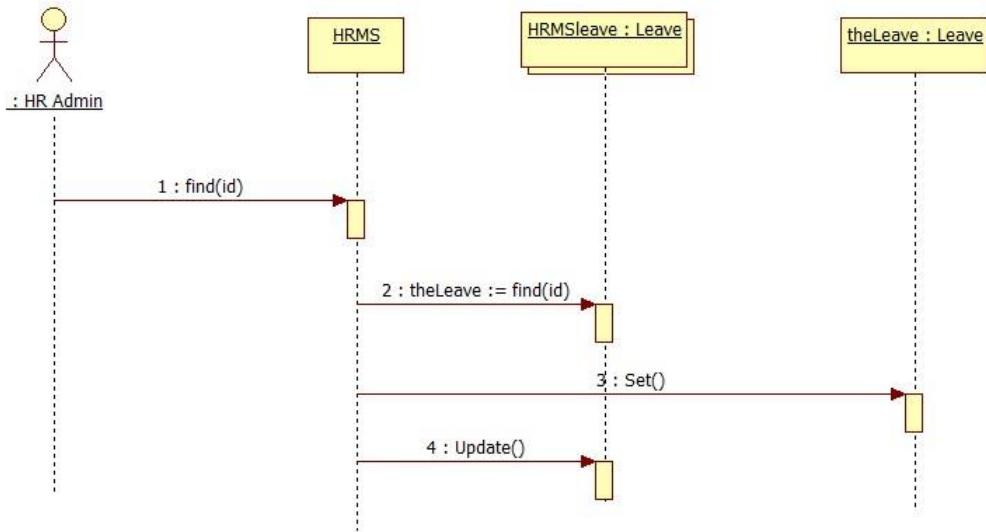


Figure 29 Sequence Diagram Update Leave Information

### Use case 22: Approve Leave

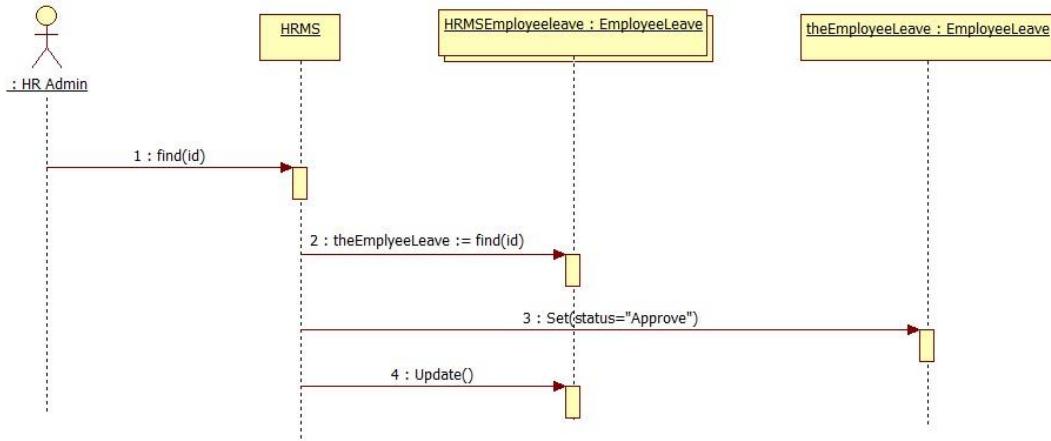


Figure 30 Sequence Diagram Approve Leave

### Use case 23: Leave Request

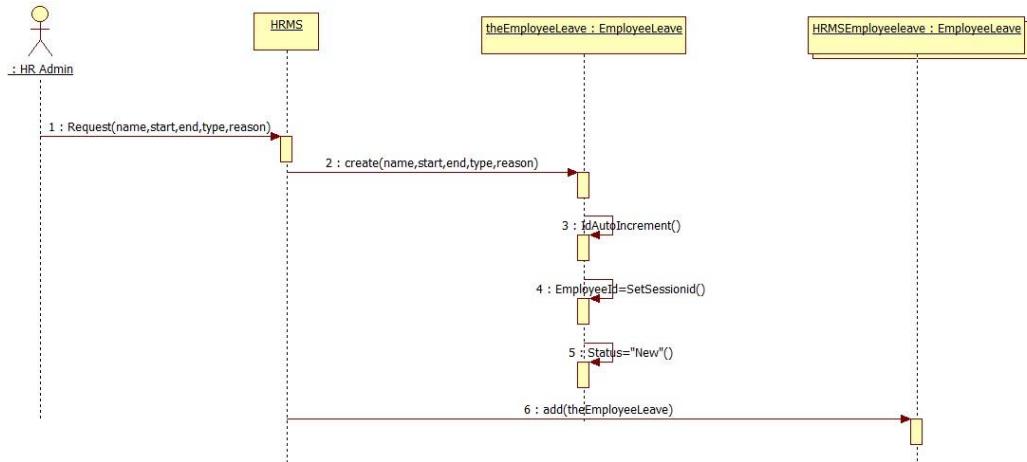


Figure 31 Sequence Diagram Leave Request

### Use case 24: Amend Leave Request

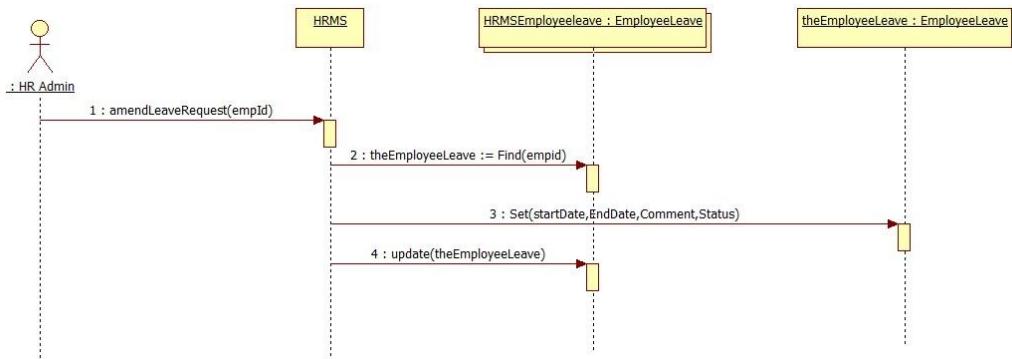


Figure 32 Sequence Diagram Amend Leave Request

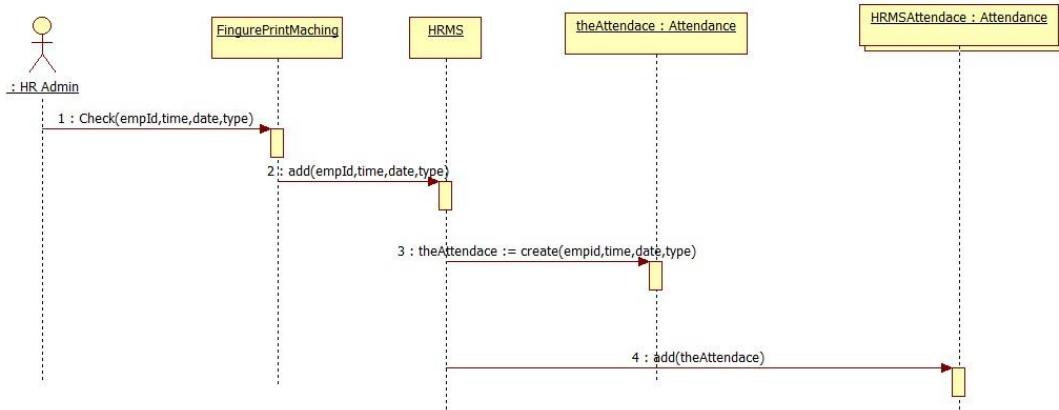
**Use case 25: Check in**

Figure 33 Sequence Diagram Check in

### Use case 26: View Attendance

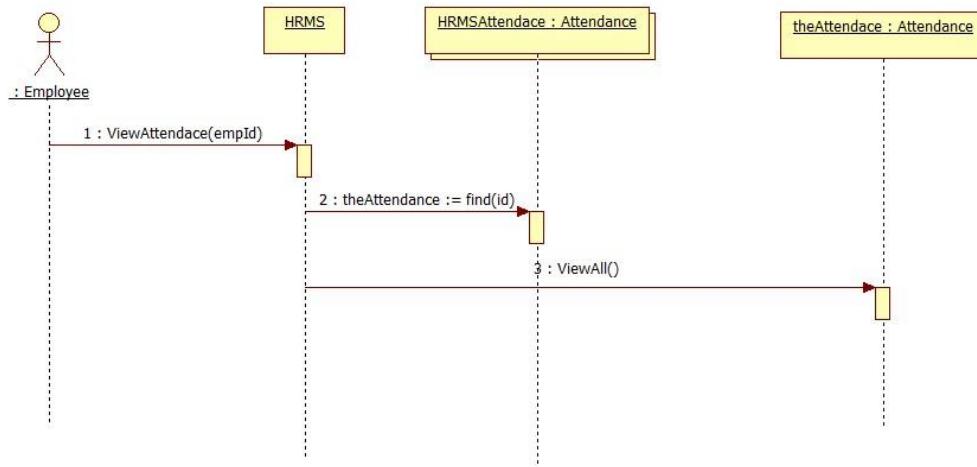


Figure 34 Sequence Diagram View Attendance

### Use case 27: Create Designation

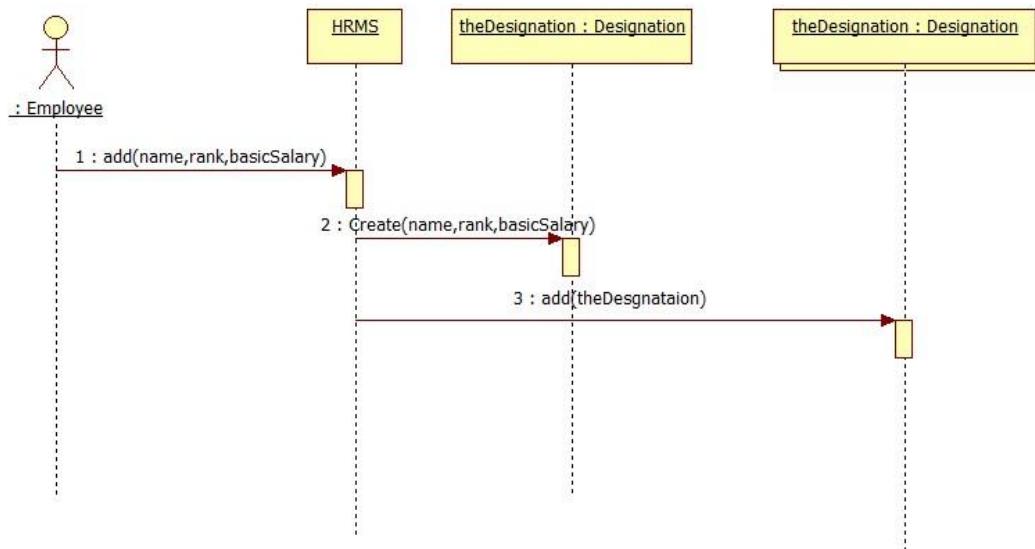


Figure 35 Sequence Diagram Create Designation

### Use case 28: Update Designation

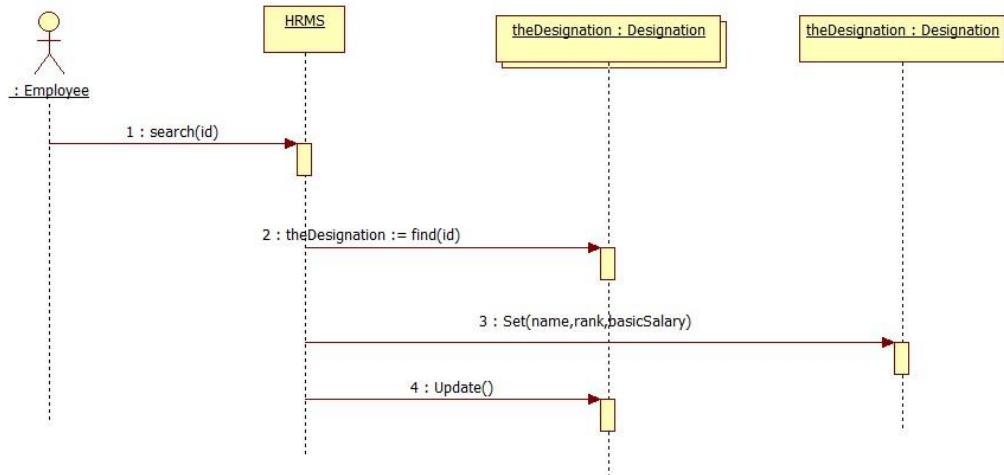


Figure 36 Sequence Diagram Update Designation

### Use case 29: Delete Designation

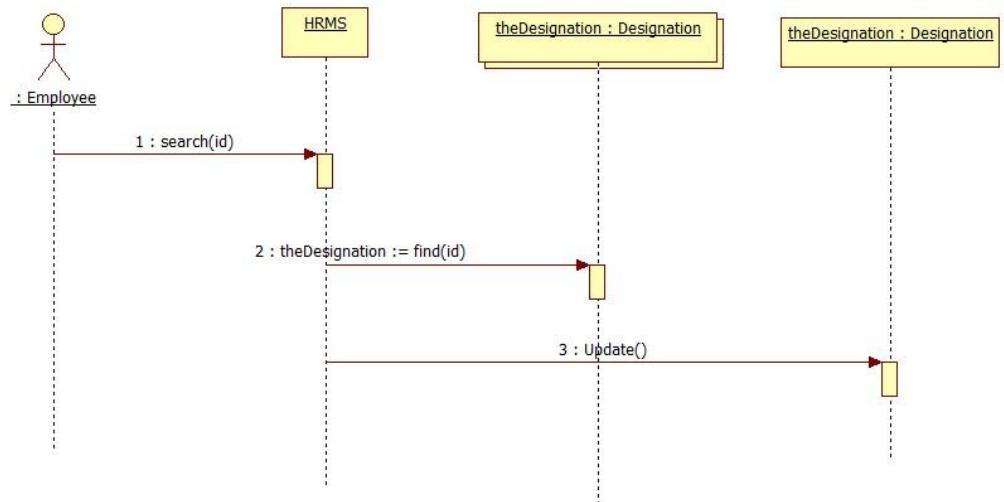


Figure 37 Sequence Diagram Delete Designation

### Use case 31: Payroll Process

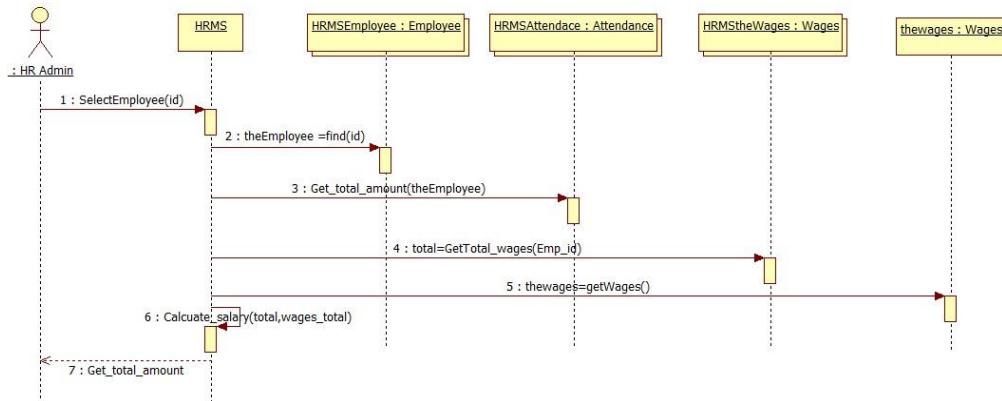


Figure 38 Sequence Diagram Payroll Process

### Use case 32: Update Attendance

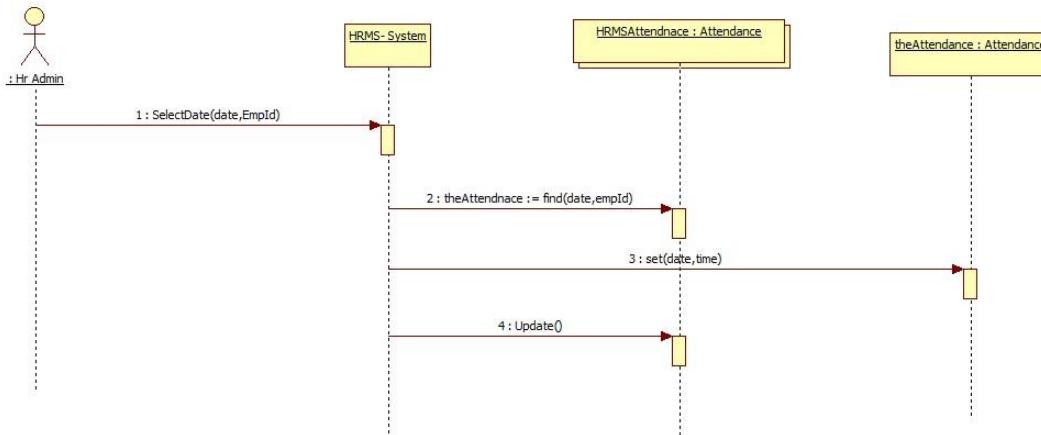


Figure 39 Sequence Diagram Update Attendance

### Use case 33 : Assign Employee to the Department

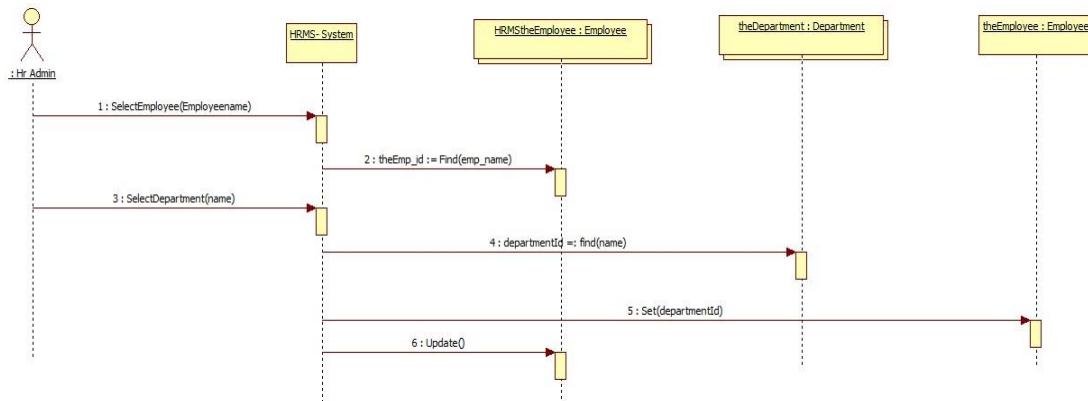


Figure 40 Sequence Diagram Assign Employee to the Department

### Use case 34: Withdraw Leave

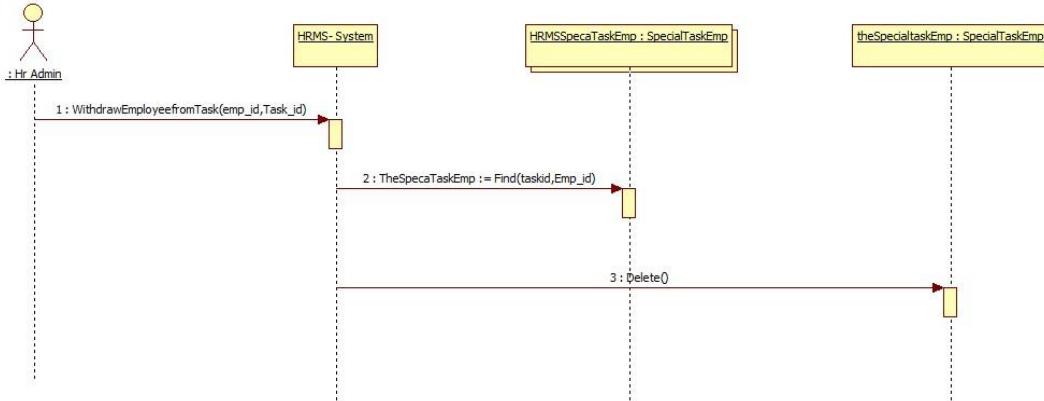


Figure 41 Sequence Diagram Withdraw Leave

### Use case 36 : Update Department Information

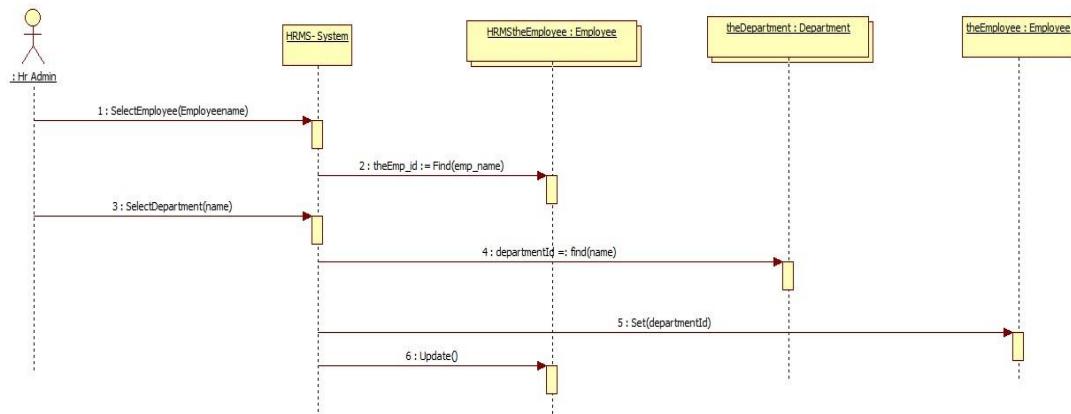


Figure 42 Sequence Diagram Update Department Information

### Use case 37:Update Calendar Event

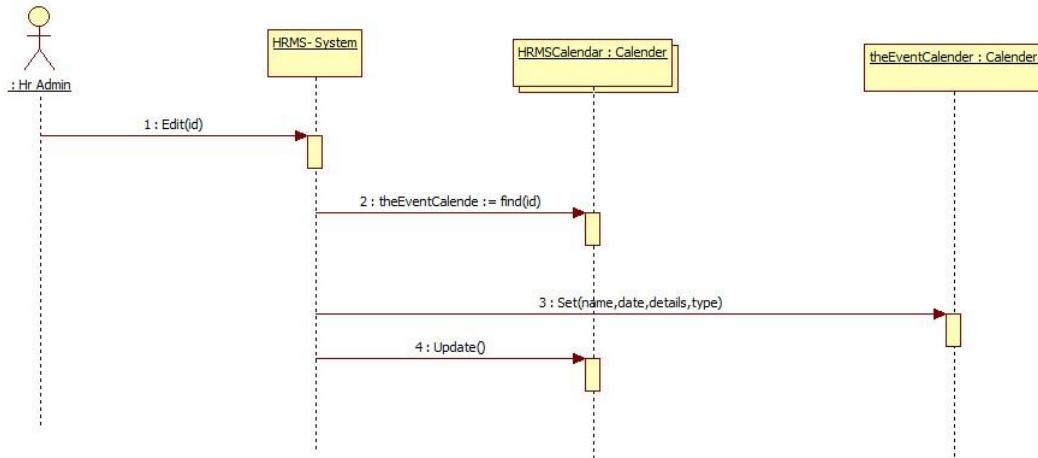


Figure 43 Sequence Diagram Update Calendar Event

### Use case 38: Delete Calendar Event

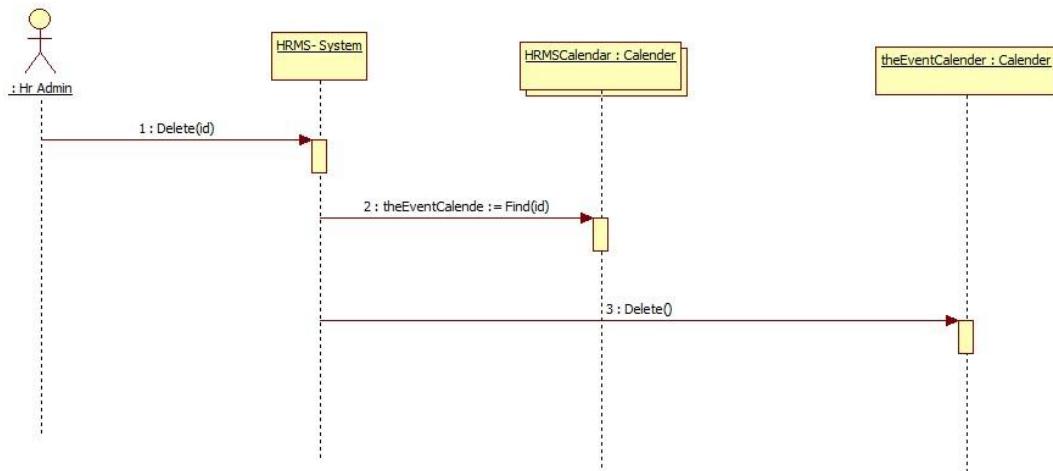


Figure 44 Sequence Diagram Delete Calendar Event

### Use case 39: Create User group

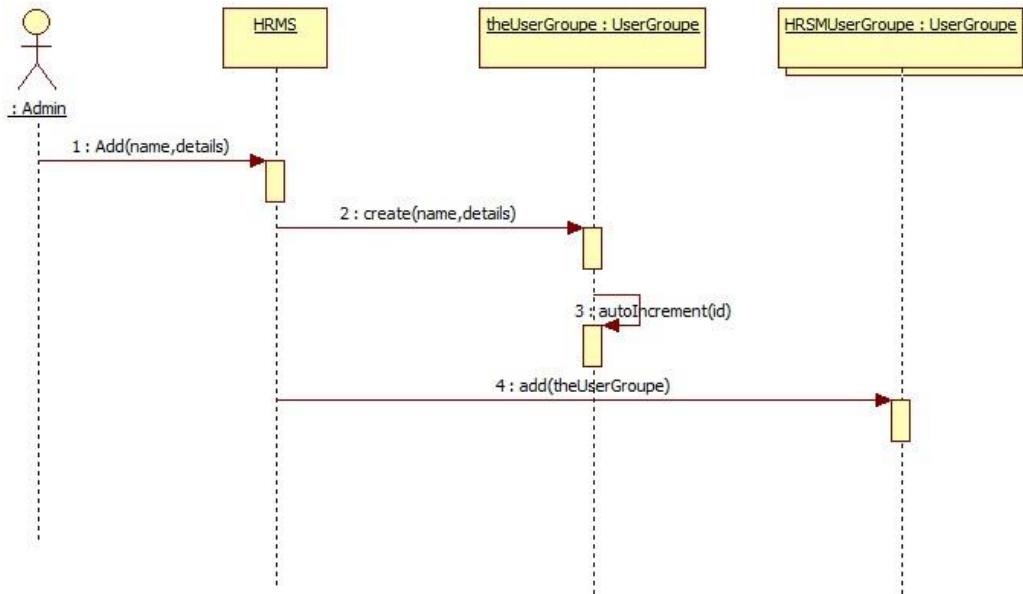


Figure 45 Sequence Diagram Create User group

### Use case 40: Delete User group

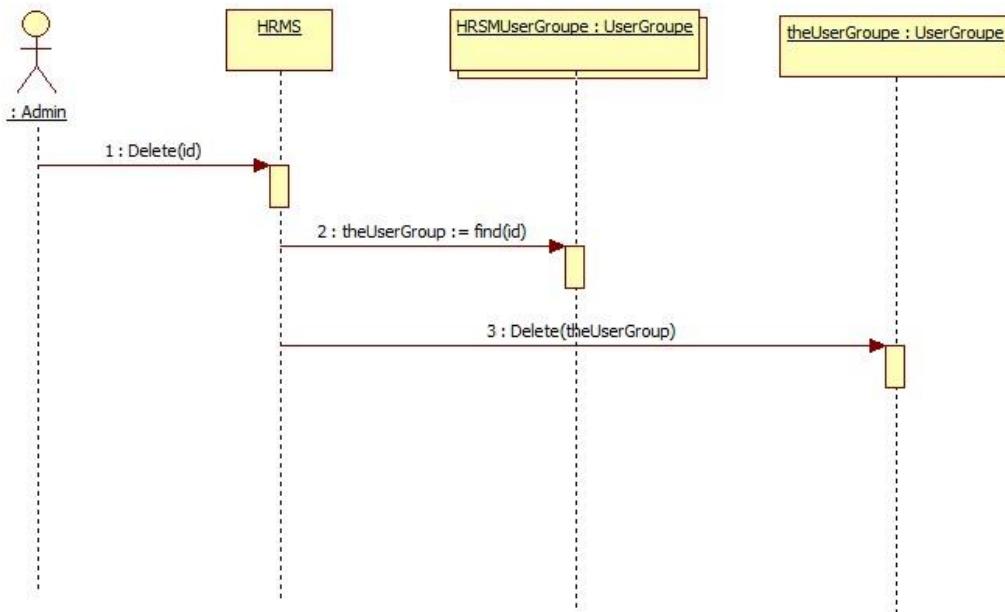


Figure 46 Sequence Diagram Delete User group

### Use case 41: Create Shift

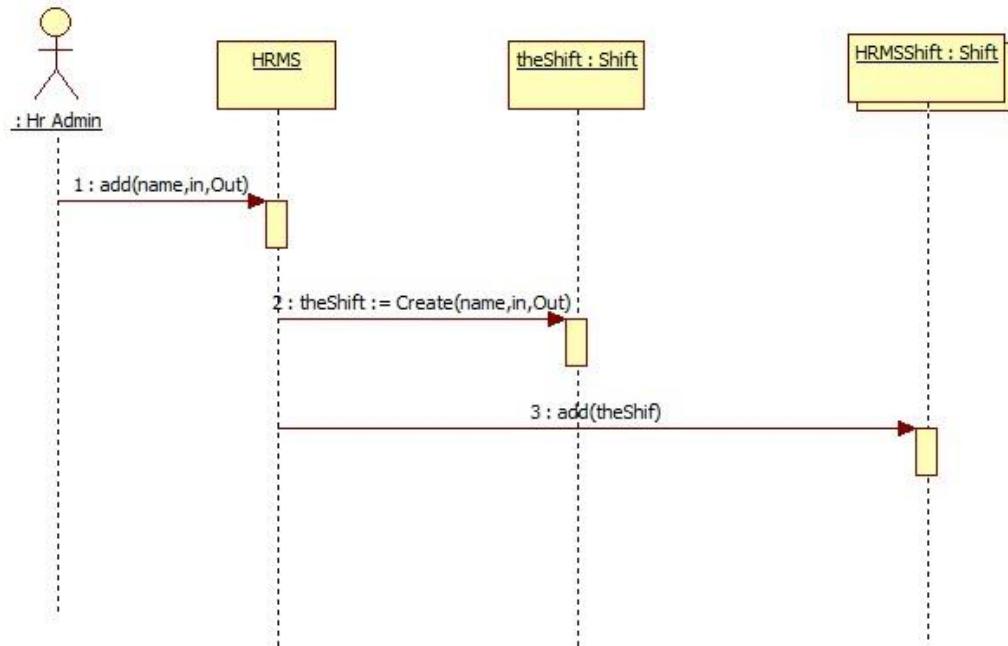


Figure 47 Sequence Diagram Create Shift

### Use case 42: Delete Shift

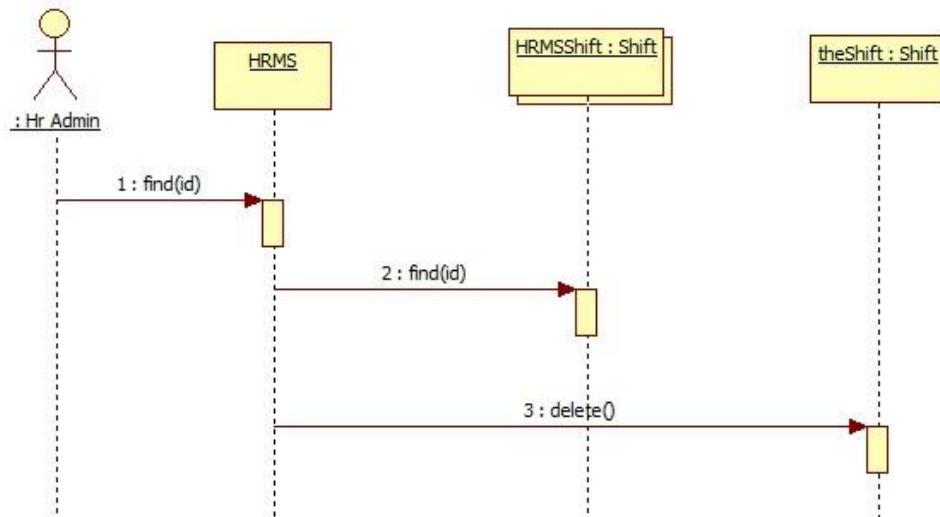


Figure 48 Sequence Diagram Delete Shift

### Use case 43: Update Shift

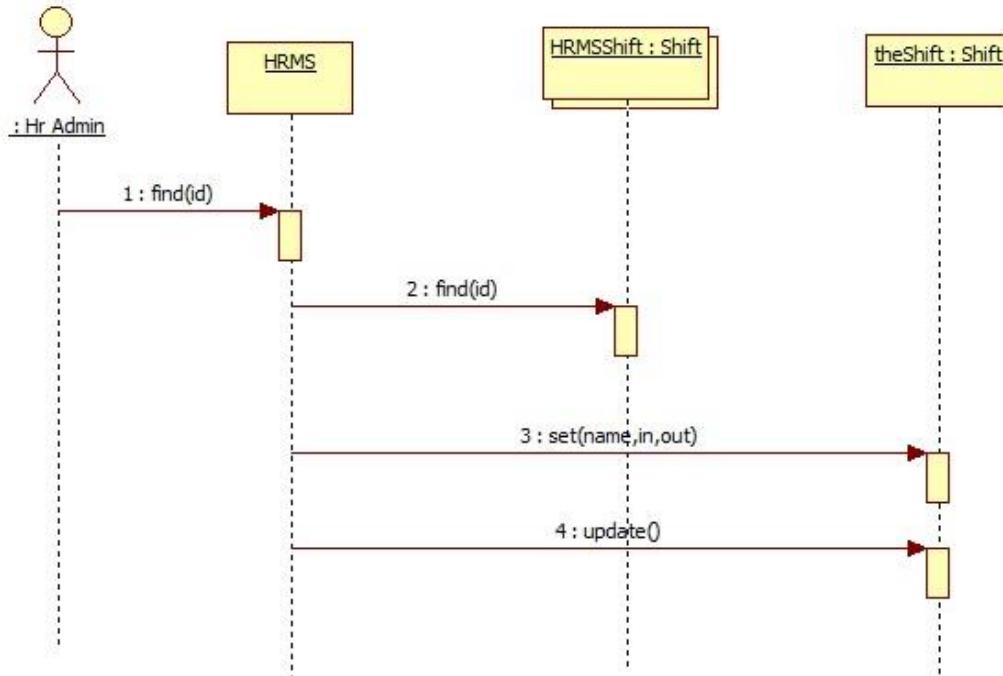


Figure 49 Sequence Diagram Update Shift

### Use case 44: Create Special task

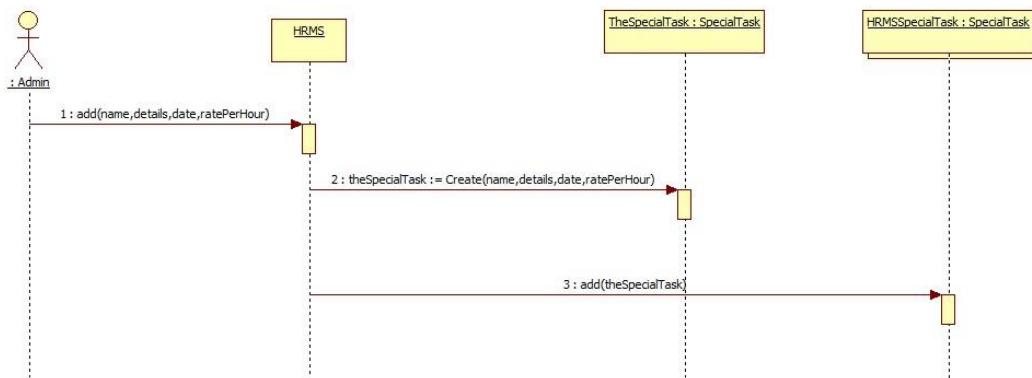


Figure 50 Sequence Diagram Create Special task

### Use case 45: Update Special task

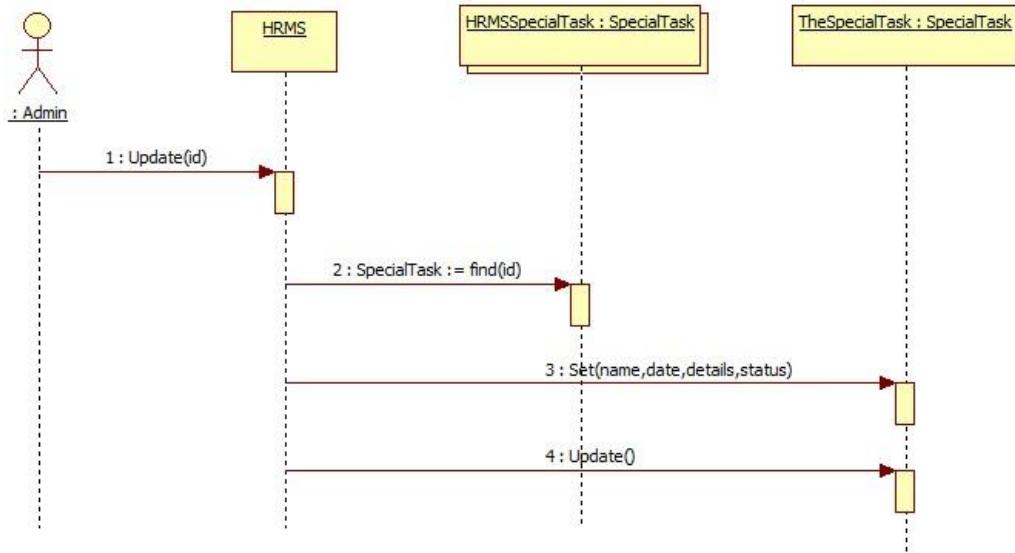


Figure 51 Sequence Diagram Update Special task

### Use case 46: Delete task

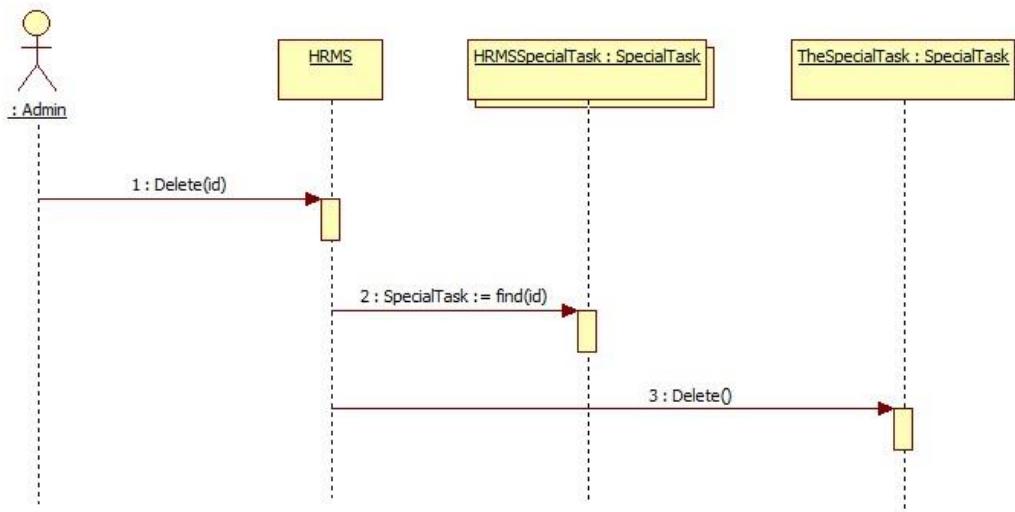


Figure 52 Sequence Diagrams Delete task

### Use case 47: Assign employee to special task Employee

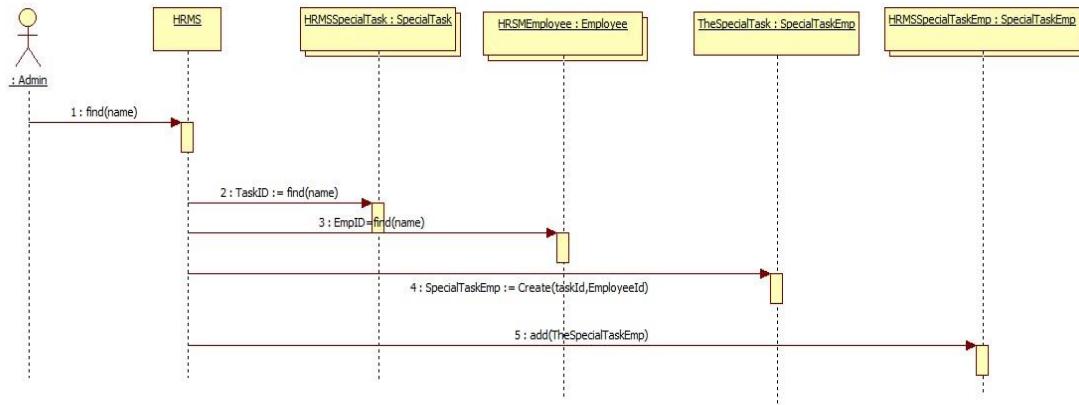


Figure 53 Sequence Diagram Assign employee to special task Employee

### Use case 48: Withdraw Employee

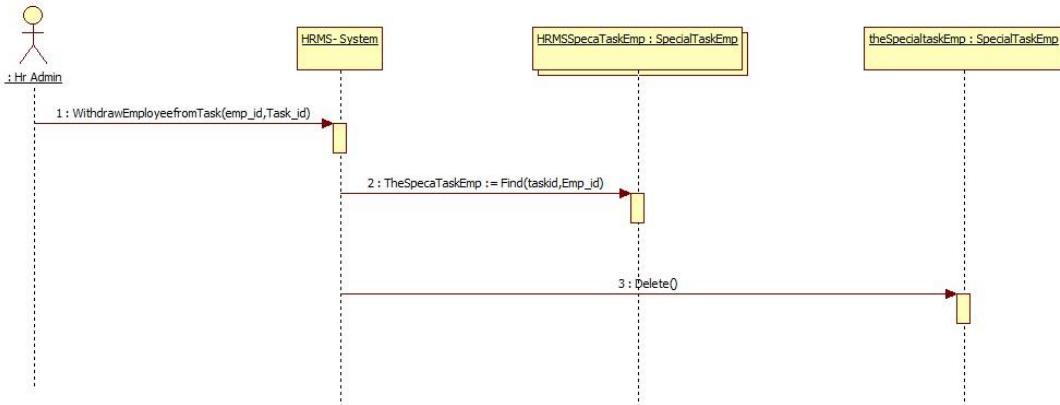


Figure 54 Sequence Diagrams Assign employee to special task Employee

## 4.5. Interface Design

### 4.5.1. HR staff dashboard

Dash board is seen when Staff login the page. User with different roles will see different widget according to their user role. Staff can edit, delete, add from the widgets.

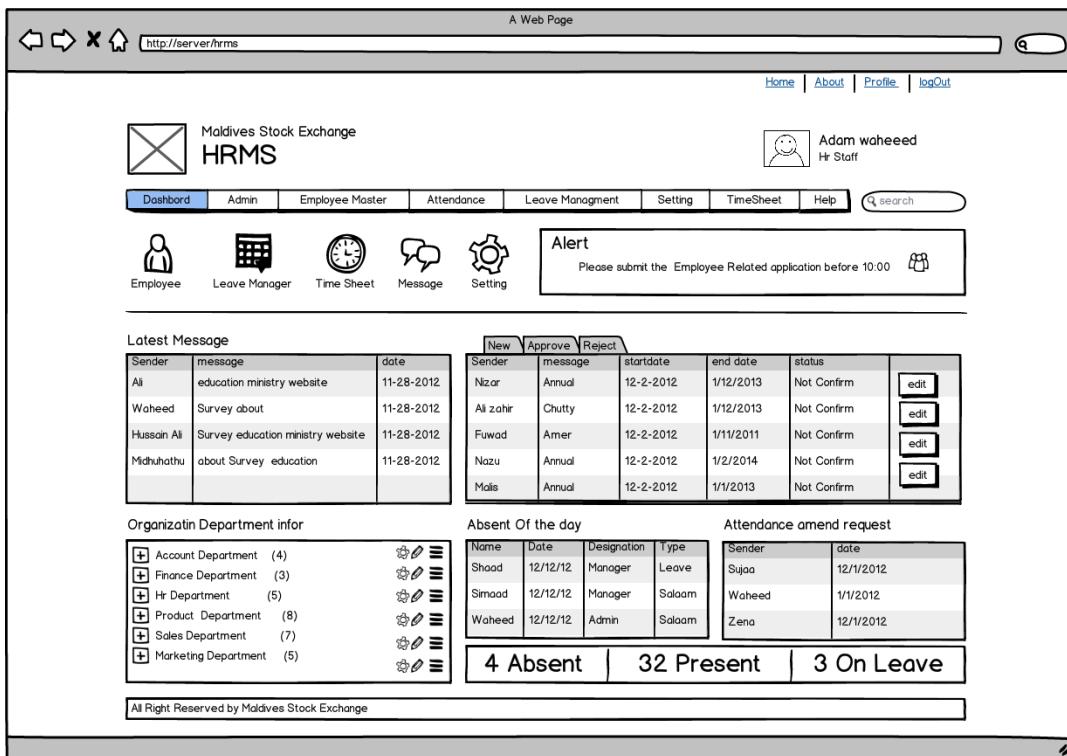


Figure 55 Dashboard Mock flow

#### 4.5.2. Screen Shots

##### Login page

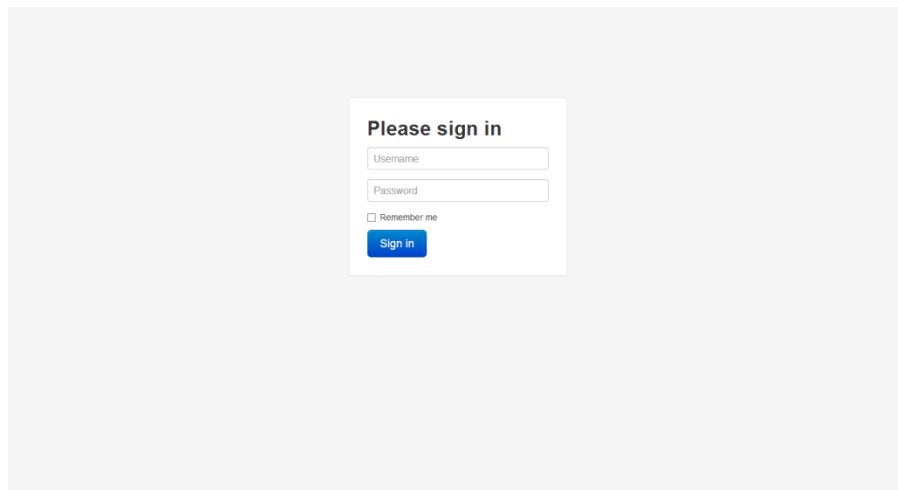


Figure 56 Login page

##### Dash Board

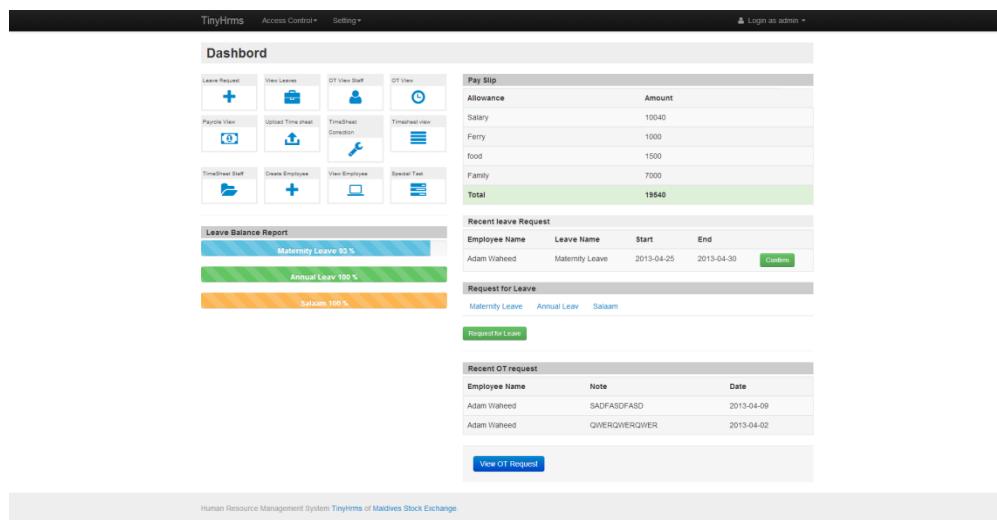


Figure 57 Dash Board

## User List

Employee Name	username	Email	Level
Hussain	dfl	dtg	Admin
Adam Waheed	admin	i@ivadasid	User

Figure 58 User List

## Create User

The screenshot shows the 'User Create' page of the TinyHrms system. The left sidebar contains navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main area is titled 'User Create' and contains the following form fields:

- Select Employee: Hussain Ali
- User Name: Email
- Email: Email
- User Level: Admin
- Password: Password

At the bottom of the form are two buttons: 'Create' and 'Reset'.

**Figure 59 Create User**

## View User Group

The screenshot shows the 'User Group' page of the TinyHrms system. The left sidebar contains navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main area is titled 'User Group' and displays a table of user groups:

User Group Name	Details	Action
Admin	adminsfadasdf	<input checked="" type="checkbox"/> <input type="button" value="X"/>
User	User	<input checked="" type="checkbox"/> <input type="button" value="X"/>
Guest	asdfasd	<input checked="" type="checkbox"/> <input type="button" value="X"/>

At the top right of the table is a search bar with the placeholder 'Username'.

**Figure 60 View User Group**

## Create User Group

User Group Create

User Group Name: User Groupe

Details:

Create Reset

Figure 61 Create User Group

## User Group Update

User Group Update

User Group Name: User

Details: User

Access Rights

<input checked="" type="checkbox"/> User_Form	<input checked="" type="checkbox"/> User_View	<input checked="" type="checkbox"/> UserGroupe_Form
<input checked="" type="checkbox"/> UserGroupe_View	<input checked="" type="checkbox"/> Leave_Form	<input checked="" type="checkbox"/> Leave_View
<input checked="" type="checkbox"/> Allowance_Form	<input checked="" type="checkbox"/> Allowance_View	<input checked="" type="checkbox"/> Department_Form
<input checked="" type="checkbox"/> Department_View	<input checked="" type="checkbox"/> LeaveRequest	<input checked="" type="checkbox"/> Calender
<input checked="" type="checkbox"/> Calender_View	<input checked="" type="checkbox"/> LeaveRequest_Staff	<input checked="" type="checkbox"/> Employee_Form
<input checked="" type="checkbox"/> Designation_Form	<input checked="" type="checkbox"/> Designation_View	<input checked="" type="checkbox"/> Employee_view
<input checked="" type="checkbox"/> index	<input checked="" type="checkbox"/> LeaveRequest_View	<input checked="" type="checkbox"/> LeaveRequest_amend
<input checked="" type="checkbox"/> Shift_Form	<input checked="" type="checkbox"/> Shift_View	<input checked="" type="checkbox"/> speciatask_Form
<input checked="" type="checkbox"/> speciatask_View	<input checked="" type="checkbox"/> timesheet_View	<input checked="" type="checkbox"/> timesheet_Error_View

Update Reset

Figure 62 User Group Update

## View Allowance

Name	Calculation type	Types	Amount
Ferry	Variable	General	120
food	Fixed	Special	1500
Family	Fixed	Special	700

Figure 63 View Allowance

## Create Allowance

**Allowance Create**

Allowance Name:

Amount:

Calculation Type:

Type:

**Create** **Reset**

Figure 64 Create Allowance

## Allowance Update

The screenshot shows the 'Allowance Update' page. At the top, there is a navigation bar with 'TinyHrms', 'Home', 'Notification', and a login link. On the left, a sidebar lists 'Leave Management' (Create leave, View Leave, Leave Request admin, Leave Request staff, Current leave Request, Leave Master), 'Employee Management' (Employee list), 'User Management' (User list), 'Allowance Management' (Allowance list), and 'Organization Setting'. The main area is titled 'Allowance Update' and contains fields for 'Allowance Name' (set to 'food'), 'Amount' (set to '1500'), 'Calculation Type' (set to 'Fixed'), and 'Type' (set to 'Special'). Below these fields are 'Create' and 'Reset' buttons. At the bottom of the page, a footer bar displays 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 65 Allowance Update

## Assign Allowance to Employee

The screenshot shows the 'Employee list' page with a modal window titled 'Select Allowance'. The modal lists two allowances: 'food' (amount 1500) and 'Family' (amount 700). The 'Family' row has a checked checkbox in the last column. The background shows a sidebar with 'Leave Management', 'Employee Management' (Employee list), 'User Management', 'Allowance Management' (Allowance list), and 'Organization Setting'. To the right of the modal, there is a 'Designation' section with tables for 'Director' and 'Manager' roles, each with edit and delete icons. A footer bar at the bottom of the page shows 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 66 Assign Allowance to Employee

## View Calendar event list

Calender list			
	Date	Details	Type
4	2013-02-03	asdasd	holiday
5	2013-02-19	asdasd	holiday
9	2013-02-20	asdasdas	holiday

Figure 67 Assign Allowance to Employee

## Create Calendar event

**Calender Create**

Date:

Details:

Details:

Figure 68 Create Calendar event

## Time Sheet List

The screenshot shows a web-based Human Resource Management System interface. At the top, there's a navigation bar with 'TinyHrms' logo, 'Home', 'Notification', and a login dropdown. Below the navigation is a sidebar with links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main content area is titled 'Time Sheet list'. It contains a table with columns: Employee Name, Request Date, Date, Time, and Note. Two entries are listed:

Employee Name	Request Date	Date	Time	Note
Adam Waheed	2013-03-14	2013-04-14	2013-04-14 09:00:00	I forgot to checking
Adam Waheed	2013-03-15	2013-04-15	2013-04-15 09:00:00	adasasdcfasdfasdfsdf

At the bottom of the page, there's a footer bar with the text 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

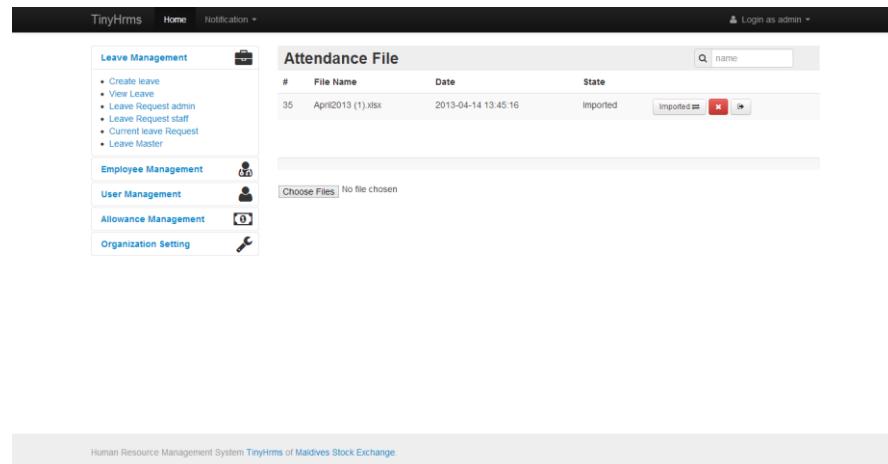
Figure 69 Time Sheet List

## Time Sheet Reading

This screenshot shows the same HRM system interface as Figure 69, but it's focused on a single time entry. The table from Figure 69 is present here, but only the second row (the entry for Adam Waheed on 2013-03-15) is highlighted with a yellow background. Below the table, a message 'Timesheet of Adam Waheed on 2013-04-15' is displayed. At the bottom of the page, there are 'Back' and 'Save' buttons.

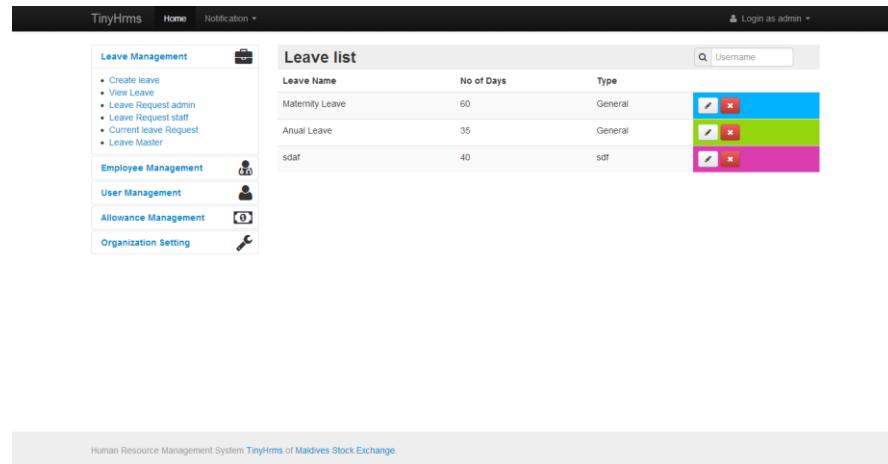
Figure 70 Time Sheet Reading

## Update Attendance File



**Figure 71 Update Attendance File**

## View Leave list



**Figure 72 View Leave list**

## Create Leave

Human Resource Management System TinyHrms of Maldives Stock Exchange

**Figure 73 Create Leave**

## Leave request by Staff

Human Resource Management System TinyHrms of Maldives Stock Exchange

**Figure 74 Leave request by Staff**

## Leave request by Admin

Figure 75 Leave request by Admin

## Requested Leave

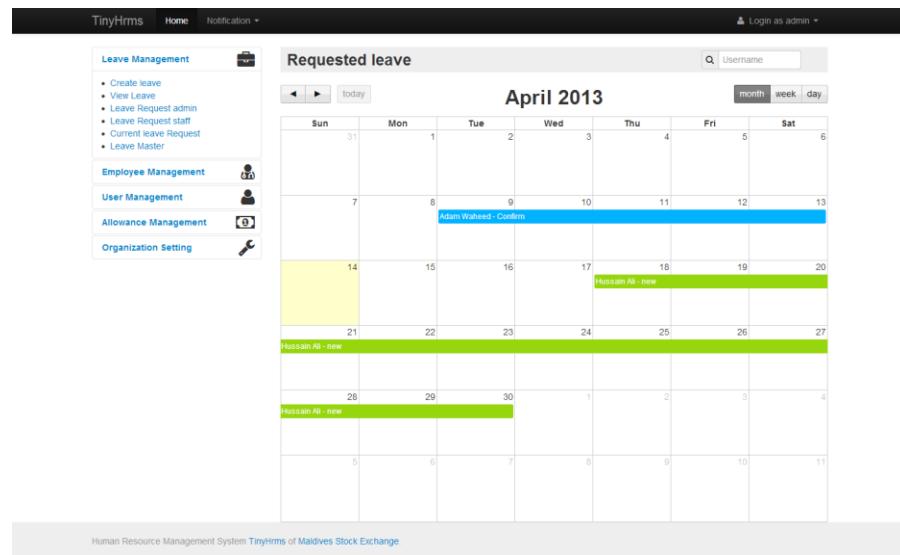


Figure 76 Requested Leave

## Amend Leave Request

The screenshot shows the 'Amend Leave Request' page. On the left, there is a sidebar with navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main area has a title 'Amend Leave Request' and a sub-section 'Reason : asdasdasd'. It contains fields for 'Start Date' (2013-04-09), 'End Date' (2013-04-13), and 'Type' (Confirm). At the bottom are 'Create' and 'Reset' buttons. A footer at the bottom of the page reads 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 77 Amend Leave Request

## View Department

The screenshot shows the 'Department list' page. On the left, there is a sidebar with navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main area displays a table with columns 'Id' and 'Name'. The data is as follows:

Id	Name
3	Human Resource Management
4	Finance Department
5	Procurement

A search bar labeled 'Username' is located at the top right of the table area. A footer at the bottom of the page reads 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 78 View Department

## Update Department

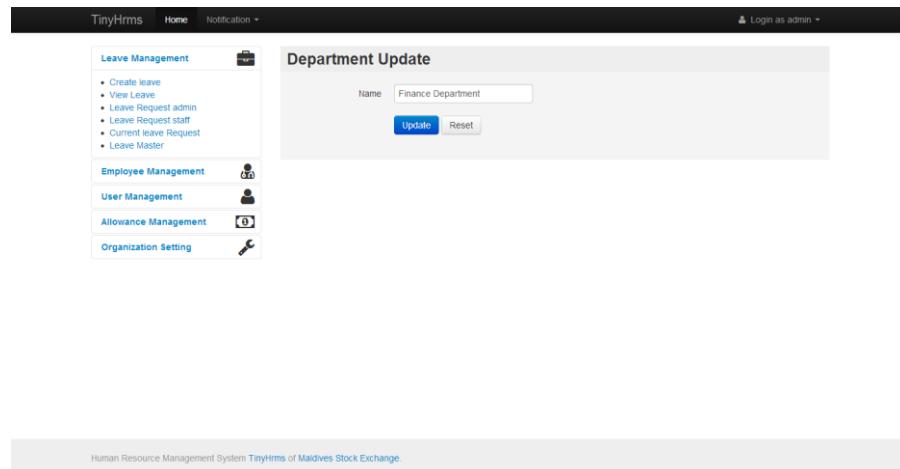


Figure 79 Update Department

## Assign Employee to department

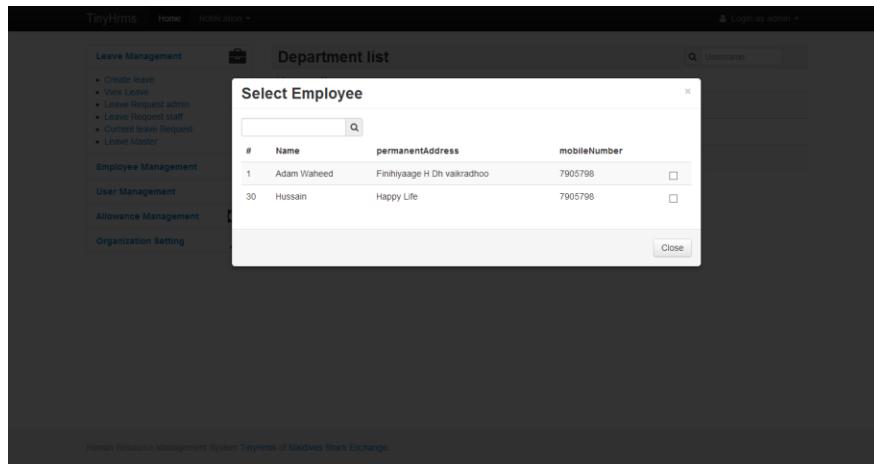


Figure 80 Assign Employee to department

## View Designation

The screenshot shows the 'Designation list' page of the TinyHrms system. At the top, there is a navigation bar with links for 'Leave Management', 'Employee Management', 'User Management', 'Allowance Management', and 'Organization Setting'. The main content area displays a table titled 'Designation list' with columns for 'Name', 'Rank', and 'Salary'. Two entries are listed: 'Director' with rank 2 and salary 10000, and 'Manager' with rank 1 and salary 20000. Each entry has edit and delete icons next to it. A search bar at the top right allows users to search by username.

Figure 81 View Designation

## Create Designation

The screenshot shows the 'Designation Create' page of the TinyHrms system. It features a sidebar with management links and a main form for creating a new designation. The form includes fields for 'Designation Name' (with a placeholder 'Name'), 'Rank' (with a placeholder 'amount'), and 'Basic Salary' (with a placeholder 'amount'). Below the form are 'Create' and 'Reset' buttons. The footer of the page includes the text 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 82 Create Designation

## Update Designation

The screenshot shows the 'Designation Update' page. On the left, there's a sidebar with navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main area has a title 'Designation Update' and three input fields: 'Designation Name' (set to 'Director'), 'Rank' (set to '2'), and 'Basic Salary' (set to '10000'). Below these fields are 'Create' and 'Reset' buttons.

Figure 83 Update Designation

## Employee List

The screenshot shows the 'Employee list' page. The table has columns: Name, Permanent Address, Mobile Number, and Designation. The data is as follows:

Name	Permanent Address	Mobile Number	Designation
Adam Wahed	Fintihyaage H Dh valkradhoo	7905798	Director
Hussain Ali	Kamilathaa	7789886	Director
Hussain	Happy Life	7905798	Manager

Figure 84 Employee List

## Create Employee

**Create Employee**

fullname	name
permanentAddress	permanentAddress
Join Date	<input type="text"/>
Current Address	<input type="text"/>
Id Card Number	idCardNumber
Gender	Male
mobile Number	mobileNumber
date Of Birth	<input type="text"/>
email Address	emailAddress
designation id	Director
<input type="button" value="Create"/> <input type="button" value="Reset"/>	

Figure 85 Create Employee

## View Shift

**Shift list**

#	Name	In	Out	
5	Office	09:00:00	04:30:00	
6	Shop	08:30:00	06:00:00	

Figure 86 View Shift

## Create Shift

The screenshot shows the 'Shift Create' form within the TinyHrms HRM system. The left sidebar contains navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main form area is titled 'Shift Create' and includes fields for 'Shift Name' (set to 'name'), 'In Time' (set to 'in Time'), and 'Out Time' (set to 'in Time'). Below these fields are 'Create' and 'Reset' buttons. At the bottom of the page, a footer bar displays the text 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 87 Create Shift

## Update Shift

The screenshot shows the 'Shift Update' form within the TinyHrms HRM system. The left sidebar contains navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main form area is titled 'Shift Update' and includes fields for 'Shift Name' (set to 'Office'), 'In Time' (set to '09:00:00'), and 'Out Time' (set to '04:30:00'). Below these fields are 'Create' and 'Reset' buttons. At the bottom of the page, a footer bar displays the text 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 88 Update Shift

### Add employee to shift

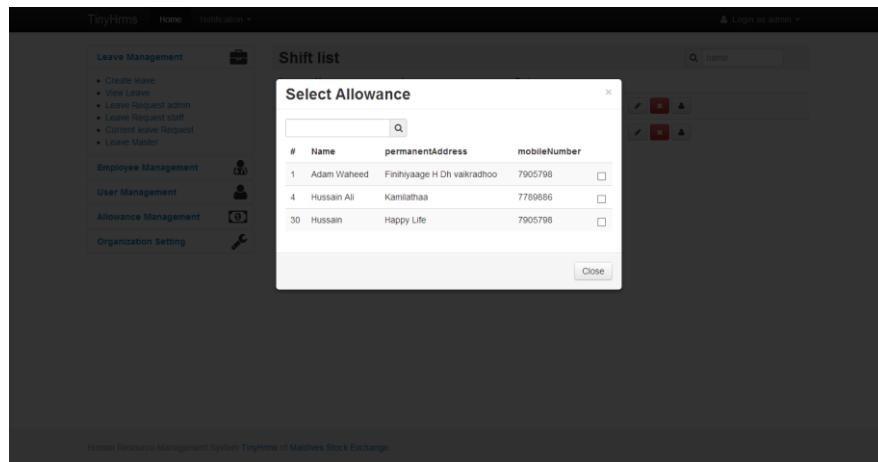


Figure 89 Add employee to shift

### Task List

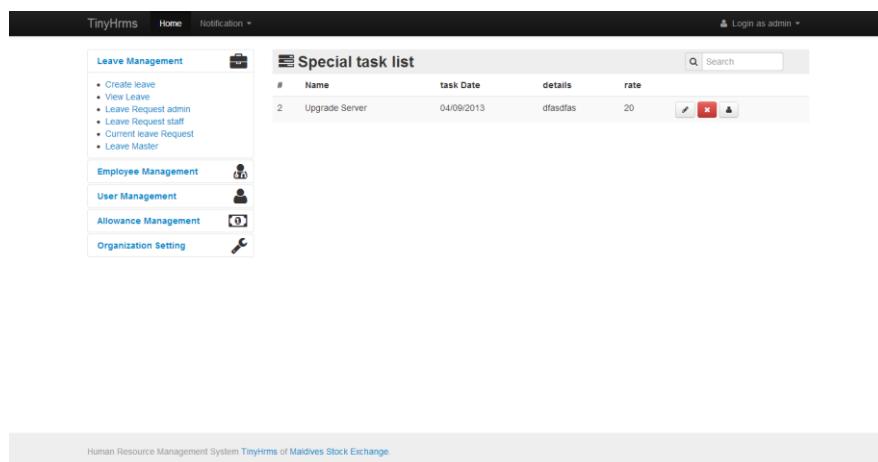


Figure 90 Task List

## Update Task

The screenshot shows the 'Special Task Update' page within the TinyHrms system. On the left, there is a sidebar with navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main content area is titled 'Special Task Update' and contains fields for 'Special Task Name' (set to 'Upgrade Server'), 'Task Date' (set to '04/09/2013'), 'Details' (containing 'dfasdas'), and 'Rate' (set to '20'). At the bottom right of the form are 'Update' and 'Reset' buttons.

Figure 91 Update Task

## Assign Employee to task

The screenshot shows the 'Select Employee' dialog box overlaid on the 'Special task list' page. The dialog has a search bar at the top and a table below with columns for '#', 'Name', 'Contact Number', and 'Select'. A 'Close' button is at the bottom right. The background page shows the same sidebar and navigation links as Figure 91.

Figure 92 Assign Employee to task

## Time Sheet

The screenshot shows the 'Your Time sheet' page of the TinyHrms system. At the top, there's a navigation bar with links for 'Leave Management', 'Employee Management', 'User Management', 'Allowance Management', and 'Organization Setting'. Below the navigation is a table titled 'Your Time sheet' with columns for date, IN, OUT, IN, OUT, IN, OUT, and Total Time. The table contains 30 rows of data, one for each day from March 21 to April 20, 2013. The last row shows summary totals at the bottom.

date	IN	OUT	IN	OUT	IN	OUT	Total Time
2013-03-21	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-03-22	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-03-23	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-03-24	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-03-25	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-03-26	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-03-27	08:01:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	7.98333333333333
2013-03-28	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-03-29	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	17:00:00	9
2013-03-30	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-03-31	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-01	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-02	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-03	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-04	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-05	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-06	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-07	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-08	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-09	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-10	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-11	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-12	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-13	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-14	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-15	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-16	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-17	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-18	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-19	08:00:00	09:30:00	10:00:00	12:00:00	13:00:00	16:00:00	8
2013-04-20							

Total Hours 248.98333333333333    Total Pay 10039.650537634

Human Resource Management System TinyHrms of Maldives Stock Exchange.

Figure 93 Time Sheet

# CHAPTER 5: IMPLEMENTATION & TESTING

## 5.1 Introduction

The earlier a defect is found in the development process of software implementation, the less expensive the fix. Testing early in the system life cycle will reduce risks such as schedule delays or cost overruns due to incomplete or unacceptable components.

Following a proper implementation method and testing of the system will prove that the system meets all its requirements, including those for performance and security.

## 5.2 Implementation Methodology

There are two types of implementation direct implementation and parallel implementation (hulasee Krishna & Pavan Kumar, 2012).

To implement this HRMS, parallel method is to be used. In this method, before completely changing over to new system, existing system and new system will be used simultaneously until the users are well trained to ready to roll out the system.

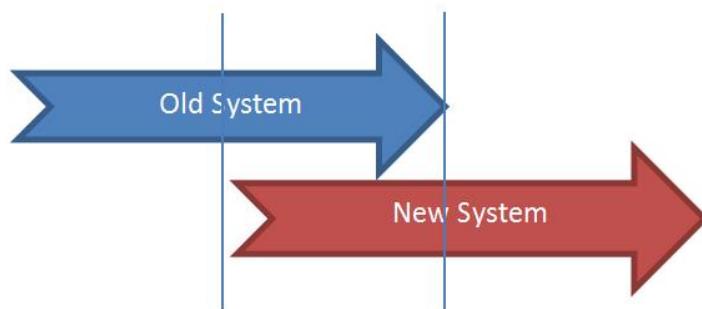


Figure 94 Parallel Implementation

## 5.3 Implementation

### 5.3.1 Work Items List

Task Name	Priority	State	Assign to	Duration
Develop Project Outline	Medium	DONE	Adam, Simadh, Maushamie	3 days
Develop Project Outline Approved	Medium	DONE	Simadh, Maushamie	1 day
Establish Scope Statement	Low	DONE	Adam, Simadh, Maushamie	2 days
Market Positing and Research	Medium	DONE	Adam, Simadh	8 days
Schedule Project	High	DONE	Maushamie	4 days
Initial Requirement Gathering	High	DONE	Adam, Maushamie	5 days
Identifying Required Resource	Medium	DONE	Adam, Simadh, Maushamie	1 day
Risk management plan	Medium	DONE	Maushamie	7 days
Validate Project Plain Document	Medium	DONE	Maushamie	4 days
Kick of Meeting with client	High	DONE	Adam, Simadh, Maushamie	1 day
Interviewing	High	DONE	Adam	3 days
Questionnaires	Medium	DONE	Simadh	4 days
Create use Case of the System	Medium	DONE	Adam, Simadh, Maushamie	2 days
Requirement Specification Documentation	Medium	DONE	Adam, Simadh, Maushamie	11 days
Evaluation Requirement Document	Medium	DONE	Maushamie	2 days
Overall System Architecture	High	DONE	Adam	3 days
Detailed Design	High	DONE	Adam	5 days
Interface design	High	DONE	Simadh	5 days
Design Documentation	Medium	DONE	Adam, Simadh, Maushamie	4 days
Validating Design Document	Low	DONE	Maushamie	2 days
Unit Programming	High	DONE	Adam, Simadh	20 days
Program Integration	High	DONE	Adam, Simadh, Maushamie	8 days

Black box testing	High	DONE	Adam, Simadh, Maushamie	4 days
White box testing	High	DONE	Adam, Simadh, Maushamie	4 days
Unit testing	High	DONE	Adam, Simadh, Maushamie	5 days
Integration testing	Medium	DONE	Adam, Simadh, Maushamie	4 days
Usability testing	Medium	DONE	Adam, Simadh, Maushamie	2 days
Acceptance testing	High	DONE	Adam, Simadh	3 days
Release	Medium	DONE	Adam, Simadh	24 hrs
Install and activate	Low	ONGOING	Adam, Simadh, Maushamie	24 hrs
User Manual	Low	DONE	Adam, Simadh, Maushamie	4 hrs

### 5.3.2 Work Items List

#### Key Milestones

Milestone	Start	Finish
<b>HRMS Project</b>		
<b>1. Initiation</b>	<b>Mon 9/24/12</b>	<b>Sun 4/14/13</b>
1.1. Develop Project Outline	Mon 9/24/12	Wed 9/26/12
1.2. Develop Project Outline Approved	Thu 9/27/12	Thu 9/27/12
<b>2. Project Planning</b>	<b>Sat 9/29/12</b>	<b>Tue 10/23/12</b>
2.1. Establish Scope Statement	Sat 9/29/12	Mon 10/1/12
2.2. Market Positing and Research	Mon 10/1/12	Wed 10/10/12
2.3. Schedule Project	Thu 10/4/12	Tue 10/9/12
2.4. Initial Requirement Gathering	Tue 10/9/12	Mon 10/15/12
2.5. Identifying Required Resource	Mon 10/15/12	Mon 10/15/12
2.6. Risk management plan	Mon 10/15/12	Tue 10/23/12
<b>3. Requirement Gathering and Analysis</b>	<b>Wed 10/24/12</b>	<b>Sat 12/1/12</b>
3.1. Validate Project Plain Document	Wed 10/24/12	Mon 10/29/12
3.2. Kick of Meeting with client	Mon 10/29/12	Mon 10/29/12
3.3. Interviewing	Mon 10/29/12	Wed 10/31/12
3.4. Questionnaires	Wed 10/31/12	Mon 11/5/12
3.5. Create use Case of the System	Mon 11/5/12	Tue 11/6/12
<b>4. Requirement Analysis</b>	<b>Tue 11/6/12</b>	<b>Mon 11/26/12</b>
4.1. Functional	Tue 11/6/12	Tue 11/13/12
4.2. Non-Functional	Tue 11/13/12	Wed 11/14/12
4.3. Constraints	Thu 11/15/12	Mon 11/19/12
4.4. Requirement Specification Documentation	Mon 11/19/12	Sat 12/1/12
<b>5. Design System and Software</b>	<b>Tue 1/1/13</b>	<b>Mon 1/21/13</b>
5.1. Evaluation Requirement Document	Tue 1/1/13	Wed 1/2/13
5.2. Overall System Architecture	Wed 1/2/13	Fri 1/4/13
5.3. Detailed Design	Fri 1/4/13	Thu 1/10/13
5.4. Interface design	Thu 1/10/13	Wed 1/16/13
5.5. Design Documentation	Wed 1/16/13	Mon 1/21/13
<b>6. Coding</b>	<b>Mon 1/21/13</b>	<b>Sun 3/31/13</b>
6.1. Validating Design Document	Mon 1/21/13	Mon 1/21/13
6.2. Unit Programing	Wed 1/23/13	Tue 3/19/13
6.3. Program Integration	Tue 3/19/13	Sun 3/31/13
<b>7. Testing</b>	<b>Mon 4/1/13</b>	<b>Wed 4/10/13</b>
7.1. Black box testing	Mon 4/1/13	Tue 4/2/13
7.2. White box testing	Wed 4/3/13	Thu 4/4/13
7.3. Unit testing	Fri 4/5/13	Fri 4/5/13
7.4. Integration testing	Fri 4/5/13	Fri 4/5/13
7.5. Usability testing	Sat 4/6/13	Mon 4/8/13
7.6. Acceptance testing	Mon 4/8/13	Tue 4/9/13

<b>8. Deployment</b>	<b>Wed 4/10/13</b>	<b>Sun 4/14/13</b>
8.1. Release	Wed 4/10/13	Thu 4/11/13
8.2. Install and activate	Thu 4/11/13	Fri 4/12/13
8.3. User Manual	Sat 4/13/13	Sun 4/14/13

### 5.3.3 Work Items Assignment

Name	Priority	Size estimate (points)	State	Reference material	Target iteration	Assigned to (name)	Hours worked	Estimated remaining hrs
<b>Initiation</b>							<b>16</b>	<b>0</b>
Develop Project Outline	2	3	Done			Maushamie	12	0
Develop Project Outline Approved	2	3	Done			Adam	4	0
Project Planning			Done				72	0
Establish Scope Statement	1	5	Done			Simad	8	0
Market Positioning and Research	2	4	Done	na.sage.com/sage-hrms		Adam	32	0
Schedule Project	2	4	Done			Adam	16	0
Initial Requirement Gathering	1	4	Done			Simad	20	0
Identifying Required Resource	3	3	Done			Simad	4	0
Risk management plan	3	3	Done			Maushame	28	0
<b>Requirement Gathering and Analysis</b>			Done				<b>116</b>	<b>0</b>
Validate Project Plain Document	1	4	Done			Maushame	16	0
Kick off Meeting with client	2	4	Done			Adam / Maushame / Simad	4	0
Interviewing	3	3	Done			Adam / Maushame / Simad	12	0
Questionnaires	3	3	Done			Adam / Maushame / Simad	16	0
Create use Case of the System	1	5	Done			Adam / Maushame / Simad	8	0
<b>Requirement Analysis</b>			Done				<b>60</b>	<b>0</b>
Functional	3	2	Done			Maushamie	24	0
Non-Functional	3	2	Done			Maushamie	8	0
Constraints	2	3	Done			Simad	12	0

Requirement Specification Documentation	1	4	Done			Adam Waheed	44	0
<b>Design System and Software</b>			Done				<b>60</b>	<b>0</b>
Evaluation Requirement Document	1	4	Done			Simad	8	0
Overall System Architecture	1	5	Done			Adam	12	0
Detailed Design	2	5	Done				20	0
Interface design	2	5	Done			Maushamie / Simad	20	0
Design Documentation	2	5	Done			Simad	16	0
<b>Coding</b>			Done				<b>204</b>	<b>0</b>
Validating Design Document	1	3	Done			Simad	4	0
Unit Programming	2	5	Done			Adam	160	0
Program Integration	2	5	Done			Adam	40	0
<b>Testing</b>			Done				<b>0</b>	<b>32</b>
Black box testing	3	3	Done			Adam / Maushame/ Simad	8	0
White box testing	2	3	Done			Adam / Maushame/ Simad	8	0
Unit testing	2	3	Done			Adam / Maushame/ Simad	4	0
Integration testing	2	3	Done			Adam / Maushame/ Simad	4	0
Usability testing	2	3	Done			Adam / Maushame/ Simad	8	0
Acceptance testing	2	3	Done			Adam / Maushame/ Simad	8	0

Deployment			In process				<b>0</b>	<b>16</b>
Release	1	2	Done			Adam	0	8
Install and activate	1	3	In process			Adam Waheed	0	8
User Manual	2	5	Done			Maushamie / Simad	8	0

## 5.4 Issues

Issue	Status	Notes
1. Unavailability of the client's upper management for decision making and approval of different phases of the project slackened the scheduled tasks.	Solved	By contacting management repeatedly, the information needed was gathered.
2. Accidental deletion of valuable data.	Solved	Backup of the system is taken repeatedly.

## 5.5 Evaluation criteria

By conducting testing, it was shown that 96% of test cases passed. There is favourable response in technical demonstration.

## 5.6 Assessment

### 5.6.1 Assessment Against Objectives

The high-level objectives of TinyHRMS and how these objectives are met are as follows:

1. ***To manage employee information by automating core HR, benefits, and payroll processes for increased efficiency and productivity.***

Currently, the payroll process is done manually using MS Excel sheets and all calculations and verifications are done by HR staff. In order to make the process faster and in an efficient way, the validation and verification of data and storing of data will be done by TinyHRMS. And all calculations such as salary calculations, allowance calculation will be done by the HR staff.

2. ***To reduce the time consumption taken between processes by timely generating the necessary reports and statistics***

In HR department, all the personal data and attendance data are stored and handled in MS Excel files. There are many similar files that cause difficulties in managing the documents. When a report is required to be made, staffs have to do burdensome work making the process time consuming.

For integration process all the data is fed to the system and it will have special report producing and filtering feature that will help in report generation.

**3. *To increase retention by providing a high level of service to employees.***

All the work of HR department such as process of leave management is handled manually.

In TinyHRMS, these types of work are maintained by the system where users can request for leave through the system and the supervisor and HR head will approve the leave through the system. If the employee has leave balance, system creates leave and maintains the leave level until it reaches the limit for the year.

**4. *To reduce redundant data, and error scope by easily creating accurate reporting and analysis.***

When handling data manually using MS Excel sheets, data redundancy is a problem, different versions of same data happens to be present in the computers result high level of inaccuracy of data and error scope.

By handling these data in one centralized system (TinyHRMS), no redundant data will be present. Thus, reduces the error scope and creates accurate reporting and analysis.

**5. *To secure employee information.***

Employee personal data should be kept and maintained securely. Currently, the manual system of handling employee data does not ensure data security.

TinyHRMS is a restricted system that allows only authorized users to access information. Different user levels can access different level of information. This ensures the security of employee data.

**6. *To reduce routine administration, and promote a paperless environment.***

The manual system requires a lot of paper work which increasing HR costs. TinyHRMS stores, manages and maintains the data automatically, and which reduces paper work and decrease costs.

### 5.6.2 Assessment against Schedule

Milestone	Start	Finish	Status
<b>HRMS Project</b>	<b>Mon 9/24/12</b>	<b>Sun 4/14/13</b>	
<b>Initiation</b>	<b>Mon 9/24/12</b>	<b>Thu 9/27/12</b>	
Develop Project Outline	Mon 9/24/12	Wed 9/26/12	Completed
Develop Project Outline Approved	Thu 9/27/12	Thu 9/27/12	Completed
<b>Project Planning</b>	<b>Sat 9/29/12</b>	<b>Tue 10/23/12</b>	
Establish Scope Statement	Sat 9/29/12	Mon 10/1/12	Completed
Market Positing and Research	Mon 10/1/12	Wed 10/10/12	Delayed, Completed
Schedule Project	Thu 10/4/12	Tue 10/9/12	Delayed, Completed
Initial Requirement Gathering	Tue 10/9/12	Mon 10/15/12	Delayed, Completed
Identifying Required Resource	Mon 10/15/12	Mon 10/15/12	Completed
Risk management plan	Mon 10/15/12	Tue 10/23/12	Completed
<b>Requirement Gathering and Analysis</b>	<b>Wed 10/24/12</b>	<b>Sat 12/1/12</b>	
Validate Project Plain Document	Wed 10/24/12	Mon 10/29/12	Completed
Kick of Meeting with client	Mon 10/29/12	Mon 10/29/12	Completed
Interviewing	Mon 10/29/12	Wed 10/31/12	Completed
Questionnaires	Wed 10/31/12	Mon 11/5/12	Completed
Create use Case of the System	Mon 11/5/12	Tue 11/6/12	Delayed, Completed
<b>Requirement Analysis</b>	<b>Tue 11/6/12</b>	<b>Mon 11/26/12</b>	
Functional	Tue 11/6/12	Tue 11/13/12	Completed
Non-Functional	Tue 11/13/12	Wed 11/14/12	Completed
Constraints	Thu 11/15/12	Mon 11/19/12	Completed
Requirement Specification Documentation	Mon 11/19/12	Sat 12/1/12	Completed
<b>Design System and Software</b>	<b>Tue 1/1/13</b>	<b>Mon 1/21/13</b>	
Evaluation Requirement Document	Tue 1/1/13	Wed 1/2/13	Delayed, Completed
Overall System Architecture	Wed 1/2/13	Fri 1/4/13	Delayed, Completed
Detailed Design	Fri 1/4/13	Thu 1/10/13	Delayed, Completed
Interface design	Thu 1/10/13	Wed 1/16/13	Completed
Design Documentation	Wed 1/16/13	Mon 1/21/13	Delayed, Completed
<b>Coding</b>	<b>Mon 1/21/13</b>	<b>Sun 3/31/13</b>	
Validating Design Document	Mon 1/21/13	Mon 1/21/13	Completed
Unit Programming	Wed 1/23/13	Tue 3/19/13	Completed
Program Integration	Tue 3/19/13	Sun 3/31/13	Completed
<b>Testing</b>	<b>Mon 4/1/13</b>	<b>Wed 4/10/13</b>	
Black box testing	Mon 4/1/13	Tue 4/2/13	Completed
White box testing	Wed 4/3/13	Thu 4/4/13	Completed
Unit testing	Fri 4/5/13	Fri 4/5/13	Completed
Integration testing	Fri 4/5/13	Fri 4/5/13	Completed
Usability testing	Sat 4/6/13	Mon 4/8/13	Completed
Acceptance testing	Mon 4/8/13	Tue 4/9/13	Completed

<b>Deployment</b>	<b>Wed 4/10/13</b>	<b>Sun 4/14/13</b>	
Release	Wed 4/10/13	Thu 4/11/13	Completed
Install and activate	Thu 4/11/13	Fri 4/12/13	Completed
User Manual	Sat 4/13/13	Sun 4/14/13	Completed

### 5.6.3 Assessment against Evaluation Criteria Test Results

For this system unit testing and acceptance testing were conducted.

Unit testing verified that the individual units of source code performed correctly. Unit testing was conducted on different modules, and three faults were identified. 6% of test case results showed that functions are not working properly and 94% of test cases passed.

After fixing the faults found while unit testing, acceptance testing was conducted. Acceptance testing was held on-site with the support from user (client) and technical team. No faults were identified in acceptance testing.

### 5.6.4 Other concerns and deviations

- The unavailability of the client's upper management for decision making and approval of different phases of the project slackened the scheduled tasks
- Even though, external threats and risk are addressed beforehand, some of the external events which we faced during this phase of the project include accidental deletion of valuable data. This unpleasant incident fined an astronomical amount of loss during the development phase of the project.
- Finger print machine integration was difficult but after extensive research on the system, the problem was solved.

## 5.7 Testing

Upon accomplishment of each use case of TinyHRMS implementation, the system is fully tested using the following strategies (unit testing, integration testing, and acceptance testing).

### 5.7.1 Testing Objectives

1. Identify all algorithmic faults in all the key components.
2. Identify any computation/precision faults in the view report feature.
3. Identify unhandled exceptions such as input of invalid/unexpected types.
4. Identify any faults in the interfaces between components through integration testing.

### 5.7.2 Testing Objectives

<b>Test case #:1.0</b>		<b>Test case name:Login</b>			
<b>System:</b> Human Resource Management System		<b>Subsystem:</b> Login page			
<b>Design by:</b> Mohamed Simad		<b>Design Date:</b> 02 March 2013			
<b>Executed by:</b> Mohamed Niyaz		<b>Execution date:</b> 10 March 2013			
<b>Short description:</b> Test the login function					
<b>Pre - Condition</b>					
User must have a login name and a password Enter the main page User name Simad must exist in the database with pass word 1234					
<b>Step</b>	<b>TCase 1.1 Action</b>	<b>Data</b>	<b>TCase 1.2 Expected System Response</b>		
1	Enter user name “Ali”	Ali	System checks if user name is available in the database login table. Message display “User does not exist”		
2	Enter user name “Simad”	Simad	System verifies and goes to password text box		
3	Enter password “1111”	1111	Message display “Wrong password”		

4	Enter password “1234”	1234	Welcome message display
<b>Post - condition</b>			
Login in to user name Simaad			

**For all test cases, refer to Appendix D**

## 5.8 Testing Phase Conclusion

The objective of the unit testing was to make sure the codes used to implement the system matches the requirements of the system. Therefore, the white box methodology was used to define both the structure and quality of codes used to build the system. This test shows that, the codes used are strong enough to meet the requirements of the system.

Besides, for the integration testing, the black box methodology was used to verify that the system functions well after different components are integrated together. This testing shows that the whole system functions smoothly with high performance and the response time of the system to shift from one page to another is approximately one to 0.2 seconds.

The beta testing was conducted as acceptance testing, and this testing has shown that user's requirement has been met.

After completing the testing phase, it was identified that 96% of test cases passed and 4% of test cases failed.

The main reason that these test cases did not pass is that some validation of data was not done. For example, a field where a number should be entered accepts characters. All faults identified during this phase were fixed.

Hence, it can be concluded that the testing phase was successful.

# CHAPTER 6: PROJECT REVIEW

## 6.1. Introduction

“TinyHRMS” project was undertaken to develop a software that can assist Maldives Stock Exchange efficiently manage their employee information by automating the core human resource functions which is currently done manually.

By the completion of the project, the product of this project, “TinyHRMS,” was fully developed to the requirements of the client.

“TinyHRMS” can increase efficiency and productivity of human resource department, can reduce the time consumption taken between processes, and the system can also reduce redundant data and error scope by easily creating accurate reports.

“TinyHRMS” also contains an intuitive user interface that provides quick access to information. Sophisticated, multi-level security options controls the accessing or viewing of information stored in the system.

Furthermore, “TinyHRMS” can increase easy retention of data by providing a high level of services to the employees. Altogether, the system can reduce routine administration and promote a paperless environment.

## 6.2. Aims and Objectives

The initial aim of this project was to provide and design a web-based application that will allow the client to streamline their human resource tasks and manage their employees in a more effective and efficient way. The system was to ensure effective utilization and maximum development of human resources, generate and maintain human resources records, and to allow proper interaction and timely access to accurate information for those who require the information.

The objectives are:

1. To reach the aim of the project, the following initial objectives of the system were attained.
2. To manage employee information by automating core HR, benefits, and payroll processes for increased efficiency and productivity was attained by developing a web-based software system that can generate required information by entering required data.
3. To reduce the time consumption taken between processes was attained by timely generating the necessary reports.
4. To increase retention was attained by providing a high level of service to employees.
5. To reduce redundant data and error scope was attained by easily creating accurate reporting and analysis.
6. To secure employee information was attained by incorporating sophisticated, multi-level security options controls for accessing or viewing of information stored in the system.
7. To reduce routine administration, and promote a paperless environment was attained by automating the required HR functions, thus eliminating the need for paperwork used by the manual system.

### **6.3. Work Items**

The estimated schedule of the project corresponded with the actual implementation times. Below is the project burndown chart based on work items and work item completion times, showing the actual progress of the project and estimated work item completion times.

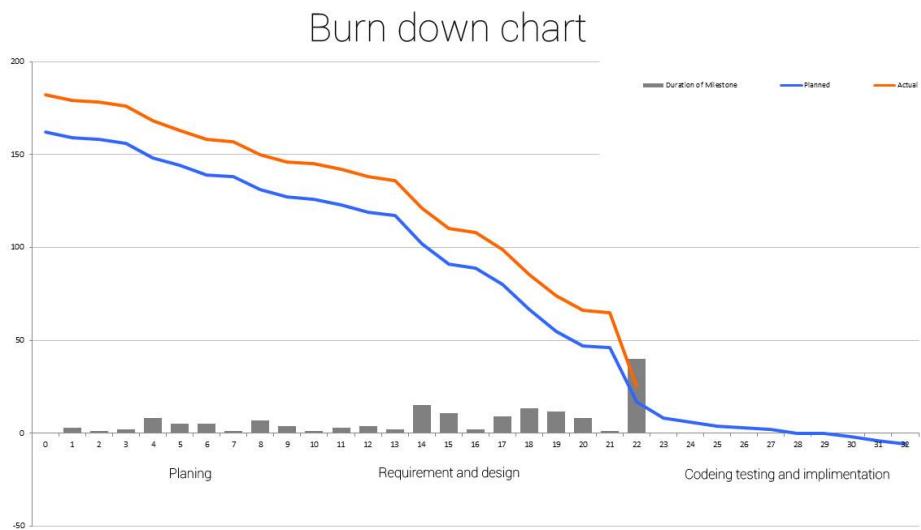


Figure 95: Project burn-down Chart

## 6.4.Successes

### 6.4.1. Team Composition and Communication

The team was a cross-functional, and representing members of the team participated in discussions that had divergent views and worked through to agreement. Teamwork and enthusiasm was evident and endured despite roadblocks. And there was a dedication to the goals of the project. The overall communication between members of the team went well.

### 6.4.2. Project Pace, Momentum and Support

The project pace was right. Team members met often enough to keep the momentum of the project. Progress reports, processes and deliverables were shared regularly with the supervisor which also kept the momentum of the project going. The client buy-in was attained because they were actively involved

## **6.5. Shortcomings**

### **6.5.1. Communication between Stakeholders**

The unavailability of the client's upper management for decision making and approval of different phases of the project slackened the scheduled tasks.

### **6.5.2. External Events**

Even though, external threats and risk are addressed beforehand, some of the external events which we faced during this phase of the project include accidental deletion of valuable data. This unpleasant incident fined an astronomical amount of loss during the development phase of the project.

## 6.6.Discussion

### 6.6.1. Development of the Project

For the development of this project, waterfall methodology was chose, mainly because of the time constraints of the project. Nowadays, projects have grown more complex and to support the changing business environments, clients/customers demand IT to respond their changing requirements more quickly.

Thus, in the future, it is thought that a more convenient methodology such as Agile method of development shall be adapted. This method is especially beneficial for small teams and rapid changes in requirements.

### 6.6.2. Team

One of the most important elements within project management is the matching up of resources to projects. This aspect is viewed as being a critical stage for success of the project (Woods, 2012).

Hence, resources such as project team will be composed and concluded based on the scope of the project.

## 6.7.Learning Outcomes

### 6.7.1. People Skills

Teamwork taught team members how to share in responsibility and coexist. When working as a team, it is needed to show respect to others by accepting their input. Working as a team also reminds that workloads can be shared, and produce synergy where the sum is greater than the parts.

### 6.7.2. Project Management Skills

**Scope Mangement:** Projects tend to expandas they go along, making it very hard to hit deadlines. To control this, it is crucial to define the scope at the very start of the project based on the requirements analysis, and then manage it closely against this signed-off definition.

**Schedule Management:** As a project's scope can easily grow, and so can the time needed to complete it. For a project to be completed successfully, despite all of the unknowns, it's important to clearly define the sequence of activities, estimate the time needed for each one, and build in sufficient contingency time to allow for the unexpected. With this information, you can develop a Project Schedule and then begin breaking it down into very specific pieces of work using a Work Breakdown Structure. To keep track of the various activities, tools such as Gantt Charts help to prepare and manage your schedule for maximum efficiency.

#### **6.7.3. Technical /Computing Skills**

To develop “TinyHRMS” using PHP, team members did extensive research on PHP, development frameworks, development methods, and especially in the field of object-oriented programming.

By the completion of the project, team members has gained confidence to work on projects using PHP more efficiently.

### **6.8. Project Management**

In excution of the project, the project management framework helped by making it able to balance the competing demands. The project management helped in identifying the problem and in defining the project scope statement.

By identifying project risks and documenting the risks helped to mitigate and avoid risks using mitigating and avoidance strategies.

The consistent method for monitoring and controlling project deliverables and milestones helped the project to be delivered on time.

The framework also improved communication among project team and stakeholders. The project team was able to measure project performance that helped to keep the project on track.

## 6.9. Cost Effectiveness

The project was developed as a requirement to Bachelors of Information Technology program. The team members or the students who undertook the project gained extensive knowledge in developing web-based software using PHP, learned project management skills, technical skills and people skills.

For the client, the developed system will contribute to overall cost reductions as the system can lower administrative burdens, increase efficiencies, drive productivity gains, maximize morale, and lessen paperwork.

# Chapter 7: CONCLUSION

## 7.1. Introduction

This project was done as a part of final year project for Bachelors of Information Technology (Hons) program offered by HELP University, Malaysia. It was undertaken to plan, design and develop a Human Resource Management system for the Maldives Stock Exchange.

By the completion of this project, the developed software was “TinyHRMS.” It is a web-based application tailored to the requirements of MSE. This software allows the client, to streamline their human resource tasks and manage their employees in a more effective and efficient way. The system facilitates good interaction and communication facilities between the employees and HR administration, and has the facility of viewing a detailed report regarding the employee.

## 7.2. Further Improvement

“TinyHRMS” is developed by analysing the requirement of Maldives Stock Exchange. The advantages of this software are that this can be enhanced, modified or changed to the growing requirements of the client, and it is developed using open-source software.

Some of the improvements that can be made to this HRMS are that it can be modified to include employee appraisal features that exist in many complex HRMS, so employee appraisal features can be added to the system.

Also, employee work assigning module for the heads can be added where heads of departments can assign task to the employees.

### **7.3.Future Work**

#### **Educate and Train Users**

Educating and training the users of the system are useful tools to help users make the most effective use of the system. An extensive training program must be developed when the system is rolled out. Also significant training must be conducted each time the software is upgraded. By reviewing the trainings of users, it will be easy to determine if more enhancements are possible.

### **7.4.Summary**

“TinyHRMS” is the product of final year project done as a requirement to Bachelors of Information Technology (Hons). “TinyHRMS” is a web-based application tailored to the requirements of MSE.

“TinyHRMS” allows the customer to streamline their human resource tasks and manage their employees in a more effective and efficient way. The system facilitates good interaction and communication facilities between the employees and HR administration, and has the facility of viewing a detailed report regarding the employee.

By the completion of project, the client was offered the solution of automated software developed to eradicate the hassle of doing their HR functions manually. The team members who undertook the project gained broad knowledge and experience both technically and professionally.

## REFERENCES

- Jibitesh Mishra, Ashok Mohanty. (2011, July 20). Software Engineering. In Software Engineering. Pearson Education India.
- Leschasin, D. (n.d.). The Meaning of Stakeholders. Retrieved November 2012, from eHow.com: [http://www.ehow.com/about\\_6615304\\_meaning-stakeholders.html](http://www.ehow.com/about_6615304_meaning-stakeholders.html)
- Malonis, E. J. (n.d.). Retrieved November 2012, from eNotes.com: <http://www.enotes.com/human-resource-information-system-hris-reference/>
- Micskei, Z., & Waeselynck, H. (2011). The many meanings of UML 2 Sequence Diagrams: a survey. *Software & Systems Modeling*, Vol., p345.
- Murugaiyan, D. (2012). WATEERFALLVs V-MODEL Vs AGILE: A COMPARATIVE STUDY ON SDLC. *International Journal of Information Technology and Business Management*.
- Obeidat, B. Y. (2010). The Relationship between Human Resource Information System (HRIS) Functions and Human Resource Management (HRM) Functionalities. *Journal of Management Research*.
- Orangehrm. (2012, 1 1). Product Features. Retrieved Oct 2012, 2012, from <http://www.orangehrm.com>: <http://www.orangehrm.com/product-features.shtml>
- Yasemin Bal, S. B. (n.d.). THE IMPORTANCE OF USING HUMAN RESOURCES INFORMATION SYSTEMS (HRIS) AND A RESEARCH ON DETERMINING THE SUCCESS OF HRIS. *Management, Knowledge and Learning*.

## APPENDIX A:

# Monthly Progress Reports

<p><b>Project Name:</b> Human Resource Management System (TinyHRMS)</p> <p><b>Team Member Names:</b> Adam / Simadh/ Maushamie</p> <p><b>Date:</b> Tue 10/23/12</p> <p><b>Reporting Period:</b> Mon 9/24/12 to Tue 10/23/12</p>
<p><b>Work completed this reporting period:</b></p> <ul style="list-style-type: none"> <li>✓ Develop Project Outline</li> <li>✓ Develop Project Outline Approved</li> <li>✓ Establish Scope Statement</li> <li>✓ Market Positing and Research</li> <li>✓ Schedule Project</li> <li>✓ Initial Requirement Gathering</li> <li>✓ Identifying Required Resource</li> <li>✓ Risk management plan</li> </ul>
<p><b>Work to complete next reporting period:</b></p> <ul style="list-style-type: none"> <li>✓ Validate Project Plan Document</li> <li>✓ Kick of Meeting with client</li> <li>✓ Interviewing</li> <li>✓ Make questionnaires</li> <li>✓ Create use case of the system</li> <li>✓ Requirement Analysis</li> <li>✓ Requirement Specification Documentation</li> </ul>
<p><b>What's going well and why:</b></p> <ul style="list-style-type: none"> <li>✓ <b>Client Corporation:</b> Meeting is held with the client and information about the current system and how it works were acquired. Project team was shown the manual system client uses for attendance, payroll and leave management.</li> <li>✓ <b>Supervisor help:</b> Supervisor helped the team to finalize the scope of the project, and gave technical explanation on how the system will work technically and what should be the objectives to fulfil the requirement of the client.</li> <li>✓ <b>Team Work:</b> Team members brainstormed on how the project will be carried out, and how to distribute workload among team members.</li> </ul>
<p><b>What's not going well and why:</b></p> <ul style="list-style-type: none"> <li>✓ There were difficulties in some technical parts such as identifying the business rules.</li> <li>✓ Numerous attempts were made to contact the client to clarify the doubts in some business rules.</li> </ul>
<p><b>Suggestions/Issues:</b></p> <ul style="list-style-type: none"> <li>✓ It was predicted that there would be issues of programming platform to be used. But it was accepted by the client.</li> <li>✓ Some risks were identified later.</li> </ul>
<p><b>Changes made:</b></p> <ul style="list-style-type: none"> <li>✓ There was no major change.</li> </ul>

**Project Name: Human Resource Management System (TinyHRMS)**

**Team Member Names: Adam / Simadh/ Maushamie**

**Date:** Sat 12/1/12

**Reporting Period:** Wed 10/24/12 to Sat 12/1/12

**Work completed this reporting period:**

- ✓ Validate Project Plan Document
- ✓ Kick off Meeting with client
- ✓ Interviewing
- ✓ Questionnaires
- ✓ Create use Case of the System
- ✓ Requirement Analysis
- ✓ Requirement Specification Documentation

**Work to complete next reporting period:**

- ✓ Evaluation Requirement Document
- ✓ Overall System Architecture
- ✓ Detailed Design
- ✓ Interface design
- ✓ Design Documentation

**What's going well and why:**

- ✓ Employees of the Maldives Stock Exchange were interviewed and questionnaires were filled to finalize the requirement.
- ✓ Drew concept map of the HRMS and brainstormed to identify what will be the modules and how modules will be integrated with each other.
- ✓ Use case for the HRMS was depicted.
- ✓ Class diagram was depicted.
- ✓ Requirement Specification Documentation was completed.

**What's not going well and why:**

- ✓ There were difficulties in contacting with the client in the process.
- ✓ Employees had difficulties in explaining the requirements and how they want the system to be.

**Suggestions/Issues:**

- ✓ Some requirements were proposed by the team such as some basic modules that need to be in HRMS. The client was unaware about these requirements until the team's proposal.

**Project changes:**

- ✓ New modules was introduced to the system

**Project Name: Human Resource Management System (TinyHRMS)**

**Team Member Names: Adam / Simadh/ Maushamie**

**Date:**to Fri 1/25/13

**Reporting Period:**Sat 12/1/12 to Fri 1/25/13

**Work completed this reporting period:**

- ✓ Evaluation Requirement Document
- ✓ Overall System Architecture
- ✓ Detailed Design
- ✓ Interface design
- ✓ Design Documentation

**Work to complete next reporting period:**

- ✓ Validating Design Document
- ✓ Unit Programming

**What's going well and why:**

- ✓ Sequence diagram was drawn to requirements.
- ✓ Database was designed.
- ✓ Interface prototype is designed and method to use in interface design is researched and verified.
- ✓ For programming, a framework was made by the team after series of tests and trials.

**What's not going well and why:**

- ✓ As some technical points of access control were missed by the team, access control had to be redesigned.
- ✓ There were difficulties in managing technical modules such as leave management.

**Suggestions/Issues:**

- ✓ Access control was redesigned
- ✓ There was an issue with development framework, but later it was solved by making a light and reusable frame work by the team.

**Project changes:**

- ✓ Access control was altered.

**Project Name: Human Resource Management System (TinyHRMS)**

**Team Member Names: Adam / Simadh/ Maushamie**

**Date:** Tue 3/19/13

**Reporting Period:** Mon 1/21/13 to Tue 3/19/13

**Work completed this reporting period:**

- ✓ Validating Design Document
- ✓ Unit Programming

**Work to complete next reporting period:**

- ✓ Program Integration
- ✓ Testing
- ✓ Black box testing
- ✓ White box testing
- ✓ Unit testing
- ✓ Integration testing
- ✓ Usability testing
- ✓ Acceptance testing
- ✓ Deployment
- ✓ Release
- ✓ Install and activate
- ✓ User Manual

**What's going well and why:**

- ✓ Database was designed in MySQL.
- ✓ The framework made by team which is used to program the system enabled reuse of methods and classes that saved lots of time in programming.
- ✓ Critical business logics were programmed.
- ✓ All the primary modules are programmed.

**What's not going well and why:**

- ✓ There were difficulties before choosing framework until we made our own framework.
- ✓ Some technical problem occurred such as data loss during programming.
- ✓ There were difficulties in meeting with client due to their busy schedule.
- ✓ There were difficulties in taking data from clients finger print machine because it is out dated.

**Suggestions/Issues:**

- ✓ Backup was made to prevent data loss.

**Project changes:**

- ✓ Some modules were included later to make the system more flexible.

## APPENDIX B:

# Gantt Chart

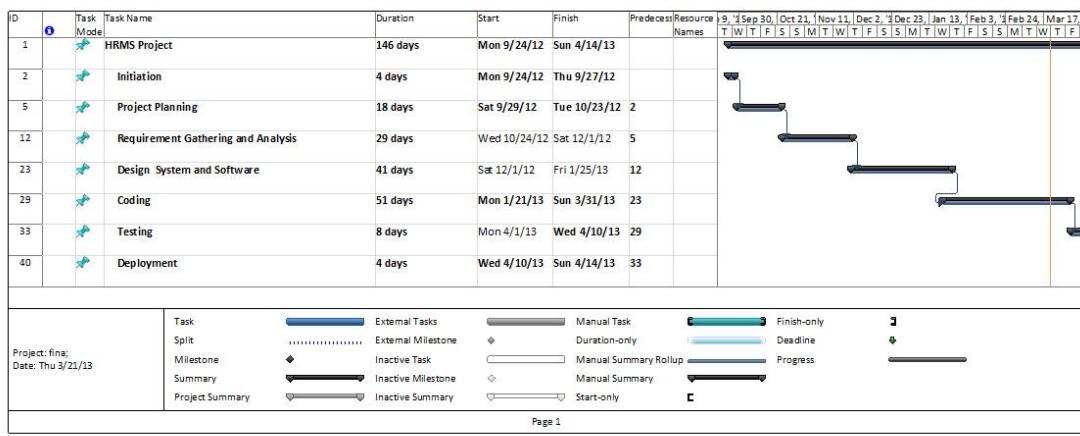


Figure 96 Gantt Chart

## APPENDIX C:

# HRMS Survey Report

For the development of this project we have conducted a firsthand survey which gave us a lot of information about the market we are targeting. We contacted 23 government offices, 20 private organizations and 20 resorts which mean altogether 63 firms were questioned about their HRMS. Out of these 63 firms, 39 firms stated that they are not using any Human Resource Management System (HRMS). These 39 firms are 19 government offices and 20 private organizations. None of the private organizations that we contacted are using HRMS .However; all the resorts claimed that they are using a full HRMS system, while only 4 of the government offices that we contacted are using HRMS. The government offices that are not using HRMS are using attendance software with fingerprint machine which collects the timesheet from this machine and is used for processing the payroll in the MS Excel Sheet. Similar to these government offices, all the private organizations that were surveyed used fingerprint machine for payroll management.

98% of those firms that are not using HRMS stated that the current market software is too expensive for them while users of the existing system say that the system is too complicated.

**APPENDIX D:****Test Cases****Unit Testing****Request Leave**

Test case #:1.0		Test case name: Create Leave							
System: Human Resource Management System		Subsystem: Leave page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Create Leave									
Pre - Condition									
Employee object must exist									
Department object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter leave type	Annual Leave		Pass					
2	Enter number of days	10	Message display “Balance not sufficient”	Pass					
3	Enter number of days	9	Message display “Leave added”	Pass					
Post - condition									
Leave is created									

Table 120 TEST CASE - REQUEST LEAVE

### View Leave

Test case #:2.0	Test case name: View Leave									
System: Human Resource Management System	Subsystem: Leave page									
Design by: Mohamed Simad	Design Date:02 March 2013									
Executed by: Mohamed Niyaz	Execution date:10 March 2013									
Short description: Test the View Shift										
Pre - Condition										
Employee object must exist Department object must exist Leave object must exist										
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail					
1	Enter leave page		Leave are shown on the page.	Pass						
Post – condition										
Leave is viewed.										

Table 121 TEST CASE - VIEW LEAVE

### Update Leave

Test case #:3.0		Test case name: Update Leave							
System: Human Resource Management System		Subsystem: Leave page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Update Shift									
Pre - Condition									
Employee object must exist									
Department object must exist									
Leave object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press edit button		Leave type and number of days view in a text box to edit.	Pass					
2	Change leave type	Family Responsibility		Pass					
3	Enter number of days and press save	2	Message display “Leave Updated”		Fail				
Post – condition									
Leave is Updated									

Table 122 TEST CASE - UPDATE LEAVE

Problem found on the Update leave was found and error was corrected and again did the test case and it was a pass.

### Delete Leave

Test case #:4.0		Test case name: Delete Leave							
System: Human Resource Management System		Subsystem: Leave page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Delete Shift									
Pre - Condition									
Employee object must exist									
Department object must exist									
Leave object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press Delete button		Message display “Leave Deleted” Leave object destroys	Pass					
Post – condition									
Leave is Deleted									

Table 123 TEST CASE - DELETE LEAVE

### Create Allowance

Test case #:5.0		Test case name: Create Allowance							
System: Human Resource Management System		Subsystem: Allowance page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Create Allowance									
Pre - Condition									
Employee object must exist									
Department object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter Allowance name	Food Allowance		Pass					
2	Enter amount	60		Pass					
3	Enter Calculation type	Dynamic		Pass					
4	Enter Type	General		Pass					
5	Press save		Allowance object created	Pass					
Post – condition									
Allowance object created									

Table 124 TEST CASE - CREATE ALLOWANCE

### **View Allowance**

Test case #:6.0		Test case name: View Allowance							
System: Human Resource Management System		Subsystem: Allowance page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the View Allowance									
Pre - Condition									
Employee object must exist									
Department object must exist									
Leave object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter Allowance page		Allowances are shown on the page.	Pass					
Post – condition									
Allowance is viewed.									

**Table 125 TEST CASE - VIEW ALLOWANCE**

## Update Allowance

Test case #:7.0		Test case name: Update Allowance							
System: Human Resource Management System		Subsystem: Allowance page							
Design by: Mohamed Simad		Design Date: 02 March 2013							
Executed by: Mohamed Niyaz		Execution date: 10 March 2013							
Short description: Test the Update Allowance									
Pre - Condition									
Employee object must exist									
Department object must exist									
Allowance object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press edit button		Allowance view in a text boxes to edit.	Pass					
2	Change Allowance type	Specialized		Pass					
3	Enter amount and press save	70	Message display “Allowance Updated”	Pass					
Post – condition									
Allowance is Updated									

Table 126 TEST CASE - UPDATE ALLOWANCE

### Delete Allowance

Test case #:8.0		Test case name: Delete Allowance							
System: Human Resource Management System		Subsystem: Allowance page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Delete Allowance									
Pre - Condition									
Employee object must exist									
Department object must exist									
Allowance object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press Delete button		Message display “Allowance Deleted” Allowance object destroys	Pass					
Post – condition									
Allowance is Deleted									

Table 127 TEST CASE - DELETE ALLOWANCE

### Search Allowance

Test case #: 9.0		Test case name: Search Allowance							
System: Human Resource Management System		Subsystem: Assign Allowance page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test search Allowance									
Pre - Condition									
Employee object must exist									
Department object must exist									
Allowance object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter name of the Allowance	Food Allowance	Allowance information views	Pass					
Post – condition									

Table 128 TEST CASE - SEARCH ALLOWANCE

### Assign Allowance

Test case #: 10.0		Test case name: Assign Allowance							
System: Human Resource Management System		Subsystem: Assign Allowance page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Assign Allowance									
Pre - Condition									
Employee object must exist									
Department object must exist									
Allowance object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Select the Allowance	Food Allowance	Staff already assigned to Food allowance is seen	Pass					
2	Select the Staff and Save	Ahmed Mohamed	Food allowance assigned to Ahmed Mohamed	Pass					
Post – condition									
Allowance is Assigned to Ahmed Mohamed									

Table 129 TEST CASE - ASSIGN ALLOWANCE

### Withdraw Allowance from employee

Test case #: 11.0		Test case name: Withdraw Allowance from role							
System: Human Resource Management System		Subsystem: Allowance page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Withdraw Allowance									
Pre - Condition									
Employee object must exist Department object must exist Allowance object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Select the Allowance	Food Allowance	Staff already assigned to Food allowance is seen	Pass					
2	Select the Staff and Withdraw	Ahmed Mohamed	Food allowance Withdraw from Ahmed Mohamed	Pass					
Post – condition									
Allowance Withdrawal									

Table 130 TEST CASE - WITHDRAW ALLOWANCE FROM EMPLOYEE

### Create Calendar

Test case #:12.0		Test case name: Create Calendar							
System: Human Resource Management System		Subsystem: Calendar page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Create Calendar									
Pre - Condition									
Employee object must exist									
Department object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter date	3/3/2012		Pass					
2	Enter Details	FitriEid		Pass					
5	Press save		Calendar object created	Pass					
Post – condition									
Calendar object created									

Table 131 TEST CASE - CREATE CALENDAR

### **View Calendar**

Test case #:13.0		Test case name: View Calendar							
System: Human Resource Management System		Subsystem: Calendar page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the View Calendar									
Pre - Condition									
Employee object must exist									
Department object must exist									
Calendar object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter Calendar page		Calendar is shown on the page.	Pass					
Post – condition									
Calendar is viewed.									

**Table 132 TEST CASE - VIEW CALENDAR**

## Update Calendar

Test case #:14.0		Test case name: Update Calendar							
System: Human Resource Management System		Subsystem: Calendar page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Update Calendar									
Pre - Condition									
Employee object must exist Department object must exist Calendar object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press edit button		Calendar view in a text boxes to edit.	Pass					
2	Change Calendar Details and press save	Alhaaa Eid	Message display “Calendar Updated”	Pass					
Post – condition									
Calendar is Updated									

Table 133 TEST CASE - UPDATE CALENDAR

## Delete Calendar

Test case #:15.0		Test case name: Delete Calendar							
System: Human Resource Management System		Subsystem: Calendar page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Delete Calendar									
Pre - Condition									
Employee object must exist									
Department object must exist									
Calendar object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response		Pass				
1	Press Delete button		Message display “Calendar Deleted” Calendar object destroys		Pass				
Post – condition									
Calendar is Deleted									

Table 134 TEST CASE - DELETE CALENDAR

## Search Calendar

Test case #: 16.0		Test case name: Search Calendar							
System: Human Resource Management System		Subsystem: Calendar page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Search Calendar									
Pre - Condition									
Employee object must exist									
Department object must exist									
Calendar object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter name of the Calendar	Alha Eid	Calendar information views	Pass					
Post – condition									

Table 135 TEST CASE - SEARCH CALENDAR

### Create Designation

Test case #:17.0		Test case name: Create Designation							
System: Human Resource Management System		Subsystem: Designation page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Create Designation									
Pre - Condition									
Employee object must exist									
Department object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter name	Computer Technician		Pass					
2	Enter rank	MM		Pass					
5	Enter basic Salary Press save	6000	Designation object created	Pass					
Post – condition									
Designation object created									

Table 136 TEST CASE - CREATE DESIGNATION

### **View Designation**

Test case #:18.0		Test case name: View Designation							
System: Human Resource Management System		Subsystem: Designation page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the View Designation									
Pre - Condition									
Employee object must exist									
Department object must exist									
Designation object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter Designation page		Designations are shown on the page.	Pass					
Post – condition									
Designation is viewed.									

Table 137 TEST CASE - VIEW DESIGNATION

### Update Designation

Test case #:19.0		Test case name: Update Designation							
System: Human Resource Management System		Subsystem: Designation page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Update Designation									
Pre - Condition									
Employee object must exist									
Department object must exist									
Designation object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected Response	System	Pass Fail				
1	Press edit button		Designation view in a text boxes to edit.	Pass					
2	Change name and press save	Computer Technician grade 2	Message display “Designation Updated”	Pass					
Post – condition									
Designation is Updated									

Table 138 TEST CASE - UPDATE DESIGNATION

### Delete Designation

Test case #:20.0		Test case name: Delete Designation							
System: Human Resource Management System		Subsystem: Designation page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Delete Designation									
Pre - Condition									
Designation object must exist									
Department object must exist									
Designation object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press Delete button		Message display “Designation Deleted” Designation object destroys	Pass					
Post – condition									
Designation is Deleted									

Table 139 TEST CASE - DELETE DESIGNATION

### Search Designation

Test case #: 21.0		Test case name: Search Destination							
System: Human Resource Management System		Subsystem: Destination page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Search Destination									
Pre - Condition									
Employee object must exist									
Department object must exist									
Destination object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter name of the Destination	It Officer	Destination information views	Pass					
Post – condition									

Table 140 TEST CASE - SEARCH DESIGNATION

### Assign Employee to Designation

Test case #: 22.0		Test case name: Employee to Designation							
System: Human Resource Management System		Subsystem: Assign Employee page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Employee to Designation									
Pre - Condition									
Employee object must exist Department object must exist Desigantion object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Select the Designation	Office Assistant	Staff already assigned to Designation is seen	Pass					
2	Select the Staff and Save	Ahmed Mohamed	Designation assigned to Ahmed Mohamed	Pass					
Post – condition									
Designation is Assigned to Ahmed Mohamed									

Table 141 TEST CASE - ASSIGN EMPLOYEE TO DESIGNATION

### Withdraw Employee from Designation

Test case #: 23.0		Test case name: Withdraw Employee from Designation							
System: Human Resource Management System		Subsystem: Designation page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Withdraw Employee from Designation									
Pre - Condition									
Employee object must exist Department object must exist Designation object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Select the Designation	Office Assistant	Staff already assigned to Designation is seen	Pass					
2	Select the Staff and Withdraw	Ahmed Mohamed	Designation Withdraw from Ahmed Mohamed	Pass					
Post – condition									
Designation Withdrawal									

Table 142 TEST CASE - WITHDRAW EMPLOYEE FROM DESIGNATION

### Create Department

Test case #:24.0		Test case name: Create Department							
System: Human Resource Management System		Subsystem: Department page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Create Department									
Pre - Condition									
Employee object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter name and Press save	Human Resource	Designation object created	Pass					
Post – condition									
Department object created									

Table 143 TEST CASE - CREATE DEPARTMENT

### **View Department**

Test case #:25.0		Test case name: View Department							
System: Human Resource Management System		Subsystem: Department page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the View Department									
Pre - Condition									
Employee object must exist									
Department object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter Departmentpage		Departments are shown on the page.	Pass					
Post – condition									
Department is viewed.									

**Table 144 TEST CASE - VIEW DEPARTMENT**

## Update Department

Test case #:26.0		Test case name: Update Department							
System: Human Resource Management System		Subsystem: Department page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Update Department									
Pre - Condition									
Employee object must exist									
Department object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press edit button		Department view in a text boxes to edit.	Pass					
2	Change name and press save	IT	Message display ‘‘Department Updated’’	Pass					
Post – condition									
Department is Updated									

Table 145 TEST CASE - UPDATE DEPARTMENT

### Delete Department

Test case #:27.0		Test case name: Delete Department							
System: Human Resource Management System		Subsystem:Department page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Delete Department									
Pre - Condition									
Designation object must exist									
Department object must exist									
Designation object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press Delete button		Message display “Department Deleted” Department object destroys	Pass					
Post – condition									
Department is Deleted									

Table 146 TEST CASE - DELETE DEPARTMENT

## Search Department

Test case #: 28.0		Test case name: Search Department							
System: Human Resource Management System		Subsystem: Department page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Search Department									
Pre - Condition									
Employee object must exist									
Department object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter name of the Department	Human Resource Section	Department information views	Pass					
Post – condition									

Table 147 TEST CASE - SEARCH DEPARTMENT

### Assign Employee to Department

Test case #: 29.0		Test case name: Employee to Department							
System: Human Resource Management System		Subsystem: Assign Department page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Employee to Department									
Pre - Condition									
Employee object must exist									
Department object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response		Pass				
1	Select the Department	HR	Staff already assigned to Department is seen		Pass				
2	Select the Staff and Save	Ahmed Mohamed	Department assigned to Ahmed Mohamed		Pass				
Post – condition									
Department is Assigned to Ahmed Mohamed									

Table 148 TEST CASE - ASSIGN EMPLOYEE TO DEPARTMENT

### Withdraw Employee from Department

Test case #: 30.0		Test case name: Withdraw Employee from Department							
System: Human Resource Management System		Subsystem: Department page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Withdraw Employee from Department									
Pre - Condition									
Employee object must exist									
Department object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Select the Department	IT	Staff already assigned to Department is seen	Pass					
2	Select the Staff and Withdraw	Ahmed Mohamed	Ahmed Mohamed Department Withdraw from Department	Pass					
Post – condition									
Department Withdrawal									

Table 149 TEST CASE - WITHDRAW EMPLOYEE FROM DEPARTMENT

### Create User

Test case #:31.0		Test case name: Create User					
System: Human Resource Management System		Subsystem: User page					
Design by: Mohamed Simad		Design Date:02 March 2013					
Executed by: Mohamed Niyaz		Execution date:10 March 2013					
Short description: Test the Create User							
Pre - Condition							
Designation object must exist							
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail		
1	Enter name, address, join date, id card, gender, mobile number, date of Birth, email address and designation ID save	Ahmed Mohamed, H.green,A21212,Male,787878, 3/5/1989, ahmed@gmail.com,3	Userobject created	Pass			
Post – condition							
User object created							

Table 150 TEST CASE - CREATE USER

### View User

Test case #:32.0		Test case name: View User							
System: Human Resource Management System		Subsystem: User page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the User									
Pre - Condition									
Employee object must exist									
Designation object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter User page		Users are shown on the page.	Pass					
Post – condition									
User is viewed.									

Table 151 TEST CASE - VIEW USER

### Update User

Test case #:33.0		Test case name: Update User							
System: Human Resource Management System		Subsystem: User page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Update User									
Pre - Condition									
Employee object must exist									
Designation object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press edit button		User view in a text boxes to edit.	Pass					
2	Change name and press save	Ahmed Ali	Message display “User Updated”	Pass					
Post – condition									
User is Updated									

Table 152 TEST CASE - UPDATE USER

## Delete User

Test case #:34.0		Test case name: Delete User							
System: Human Resource Management System		Subsystem: User page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Delete User									
Pre - Condition									
Designation object must exist									
Employee object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press Delete button		Message display “User Deleted” User object destroys		Fail				
Post – condition									
User is Deleted									

Table 153 TEST CASE - DELETE USER

User did not delete there was problem in code. Code was fixed and when again case was done and it passed.

### Search User

Test case #: 35.0		Test case name: Search User							
System: Human Resource Management System		Subsystem: User page							
Design by: Mohamed Simad		Design Date: 02 March 2013							
Executed by: Mohamed Niyaz		Execution date: 10 March 2013							
Short description: Test the Search User									
Pre - Condition									
Employee object must exist									
User object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter name of the User	Ahmed Mohamed	User information views		Fail				
Post – condition									

Table 154 TEST CASE - SEARCH USER

There was a problem in SQL string and it was solved and when again test case was done and result was Pass

**Login User**

Test case #: 36.0			Test case name: Login					
System: Human Resource Management System			Subsystem: Login page					
Design by: Mohamed Simad			Design Date:02 March 2013					
Executed by: Mohamed Niyaz			Execution date:10 March 2013					
Short description: Test the Login								
Pre - Condition								
Employee object must exist								
User object must exist								
User name Simad must exist in the database with pass word 1234								
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail			
1	Enter name of the User	Ahmed Mohamed	System checks if user name is available in the database login table.  Message display “User does not exist”	Pass				
2	Enter user name “Simad”	Simad	System verifies and goes to password text box	Pass				
3	Enter password “1111”	1111	Message display “Wrong password”	Pass				
4	Enter password “1234”	1234	Welcome message display	Pass				
Post – condition								
Login in to user name Simad								
Session Created								

**Table 155 TEST CASE - LOGIN USER**

### **Logout User**

Test case #: 37.0		Test case name: Logout							
System: Human Resource Management System		Subsystem: Main page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Logout									
Pre - Condition									
Employee object must exist									
User object must exist									
Session must be Created									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press logout button		Go back to Login page and Session destroyed.	Pass					
Post – condition									
Session destroyed									

Table 156 TEST CASE - LOGOUT USER

### Create User Group

Test case #:38.0		Test case name: Create User Group							
System: Human Resource Management System		Subsystem: User Group page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Create User Group									
Pre - Condition									
Employee object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter name, detail and Press save	HR Admin, Change HR functions	User Group object created	Pass					
Post – condition									
Department object created									

Table 157 TEST CASE - CREATE USER GROUP

### View User Group

Test case #:39.0		Test case name: View User Group							
System: Human Resource Management System		Subsystem: User Group page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the View User Group									
Pre - Condition									
Employee object must exist									
User Group object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter User Group page		User Group are shown on the page.	Pass					
Post – condition									
User Group is viewed.									

Table 158 TEST CASE - VIEW USER GROUP

## Update User Group

Test case #:40.0		Test case name: Update Department							
System: Human Resource Management System		Subsystem: User Group page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Update User Group									
Pre - Condition									
Employee object must exist									
User Group object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press edit button		User Group view in a text boxes to edit.	Pass					
2	Change name and press save	Adimin	Message display “User Group Updated”	Pass					
Post – condition									
User Group is Updated									

Table 159 TEST CASE - UPDATE USER GROUP

## Delete User Group

Test case #:41.0		Test case name: Delete Department							
System: Human Resource Management System		Subsystem: User Group page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Delete User Group									
Pre - Condition									
User Group object must exist									
Employee object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press Delete button		Message display “User Group Deleted” User Group object destroys	Pass					
Post – condition									
User Group is Deleted									

Table 160 TEST CASE - DELETE USER GROUP

## Search User Group

Test case #: 42.0		Test case name: User Group					
System: Human Resource Management System		Subsystem: User Group page					
Design by: Mohamed Simad		Design Date:02 March 2013					
Executed by: Mohamed Niyaz		Execution date:10 March 2013					
Short description: Test the Search User Group							
Pre - Condition							
Employee object must exist							
User object must exist							
User Group object must exist							
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail		
1	Enter name of the User Group	Admin	User Group information views	Pass			
Post – condition							

Table 161 TEST CASE - SEARCH USER GROUP

### Assign User role

Test case #: 43.0		Test case name: Assign User role							
System: Human Resource Management System		Subsystem: User role page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Assign User role									
Pre - Condition									
Employee object must exist									
User object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Select the User Role	Admin	Staff already assigned to user role is seen	Pass					
2	Select the Staff and Save	Ahmed Mohamed	user role assigned to Ahmed Mohamed	Pass					
Post – condition									
user role is Assigned to Ahmed Mohamed									

Table 162 TEST CASE - ASSIGN USER ROLE

### Withdraw User role

Test case #: 44.0		Test case name: Withdraw Employee User Role							
System: Human Resource Management System		Subsystem: User Role page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Withdraw Employee User Role									
Pre - Condition									
Employee object must exist									
User Role object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response		Pass				
1	Select the Staff and Withdraw	Ahmed Mohamed	User Role Withdraw from Ahmed Mohamed		Pass				
Post – condition									
User Role Withdrawal									

Table 163 TEST CASE - WITHDRAW USER ROLE

### Create Leave Type

Test case #:45.0		Test case name: Create Leave Type							
System: Human Resource Management System		Subsystem: Leave Typepage							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Create Leave Type									
Pre - Condition									
Employee object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter name, detail and Press save	Annual Leave, 30 days Leave	Leave Type object created	Pass					
Post – condition									
Leave Type object created									

Table 164 TEST CASE - CREATE LEAVE TYPE

### View Leave Type

Test case #:46.0		Test case name: View Leave Type							
System: Human Resource Management System		Subsystem: Leave Type page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the View Leave Type									
Pre - Condition									
Leave Type object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Enter Leave Type page		Leave Type are shown on the page.	Pass					
Post – condition									
Leave Type is viewed.									

Table 165 TEST CASE - VIEW LEAVE TYPE

## Update Leave Type

Test case #:47.0		Test case name: Update Leave Type							
System: Human Resource Management System		Subsystem: Leave Type page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Update Leave Type									
Pre - Condition									
Leave Type object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press edit button		Leave Type view in a text boxes to edit.	Pass					
2	Change name and press save	Family Leave	Message display "Leave Type Updated"	Pass					
Post – condition									
Leave Type is Updated									

Table 166 TEST CASE - UPDATE LEAVE TYPE

### Delete Leave Type

Test case #:48.0		Test case name: Delete Leave Type							
System: Human Resource Management System		Subsystem: User Leave Type page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Delete Leave Type									
Pre - Condition									
Leave Type object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Press Delete button		Message display “Leave Type Deleted” Leave Type object destroys	Pass					
Post – condition									
Leave Type is Deleted									

Table 167 TEST CASE - DELETE LEAVE TYPE

### Search Leave Type

Test case #: 49.0		Test case name: Leave Type					
System: Human Resource Management System		Subsystem: Leave Type page					
Design by: Mohamed Simad		Design Date:02 March 2013					
Executed by: Mohamed Niyaz		Execution date:10 March 2013					
Short description: Test the Search Leave Type							
Pre - Condition							
Employee object must exist							
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail		
1	Enter name of the Leave Type	Annual	Leave Type information views	Pass			
Post – condition							

Table 168 TEST CASE - SEARCH LEAVE TYPE

### Approve Leave

Test case #:50.0		Test case name: Approve Leave							
System: Human Resource Management System		Subsystem: Leave page							
Design by: Mohamed Simad		Design Date:02 March 2013							
Executed by: Mohamed Niyaz		Execution date:10 March 2013							
Short description: Test the Leave Approval									
Pre - Condition									
Leave object must exist									
Step	TCase 1.1 Action	Data	TCase 1.2 Expected System Response	Pass	Fail				
1	Select Employee from the Leave page		Employees Leaves display	Pass					
2	Select the Leave and press approve	Annual Leave 3/3/2012 3 days	Message display “Leave Approved”	Pass					
Post – condition									
Leave approved									

Table 169 TEST CASE - APPROVE LEAVE

## Acceptance Test

Test Case #	Name	Result	Comments
1.0	Request Leave	Pass	
2.0	View Leave	Pass	
3.0	Update Leave	Pass	
4.0	Delete Leave	Pass	
5.0	Create Allowance	Pass	
6.0	View Allowance	Pass	
7.0	Update Allowance	Pass	
8.0	Delete Allowance	Pass	
9.0	Search Allowance	Pass	
10.0	Assign Allowance	Pass	
11.0	Withdraw Allowance from role	Pass	
12.0	Create Calendar	Pass	
13.0	View Calendar	Pass	
14.0	Update Calendar	Pass	
15.0	Delete Calendar	Pass	
16.0	Search Calendar	Pass	
17.0	Create Designation	Pass	
18.0	View Designation	Pass	
19.0	Update Designation	Pass	
20.0	Delete Designation	Pass	
21.0	Search Designation	Pass	
22.0	Assignee Employee to Designation	Pass	
23.0	Withdraw Employee from Designation	Pass	
24.0	Create Department	Pass	
25.0	View Department	Pass	
26.0	Update Department	Pass	
27.0	Delete Department	Pass	
28.0	Search Department	Pass	
29.0	Assignee Employee to Department	Pass	
30.0	Withdraw Employee from Department	Pass	
31.0	Create User	Pass	
32.0	View User	Pass	
33.0	Update User	Pass	
34.0	Delete User	Pass	
35.0	Search User	Pass	
36.0	login	Pass	
37.0	logout	Pass	
38.0	Create User Group	Pass	
39.0	View User Group	Pass	
40.0	Update User Group	Pass	

41.0	Delete User Group	Pass	
42.0	Search User Group	Pass	
43.0	Assignee user role	Pass	
44.0	Withdraw user role	Pass	
45.0	Create Leave Type	Pass	
46.0	View Leave Type	Pass	
47.0	Update Leave Type	Pass	
48.0	Delete Leave Type	Pass	
49.0	Search Leave Type	Pass	
50.0	Approve Leave	Pass	

Table 170 TEST CASE - ACCEPTANCE TEST

**APPENDIX E:**

## **Minutes of Meetings**

BIT304 Final Year Project I

HRMS

### **Minutes No. 01**

Date 13 Oct 2013

Time 3:00 PM

Venue Light Corner / Male'

**Present :**

Mr. Ibrahim waheed	Supervisor
Mr. Waris	Supervisor
Adam Waheed	Project Developer and designer
Mohamed Simad	Project Developer and UI Developer
Mariyam MAUSHAMIEIE	Project Manager,Strategic Planner and UI Devoper\

**Absent with apologies:**

Mr.Mauroof	Supervisor
------------	------------

Represented by Mr. Waris

No.	Matters Discussed	Action By
<b>1.0</b>	Topic 1: Assigning Supervisor	
1.1	Introduction of the Supervisors	Mr. Ibrahim Waheed
1.2	Give brief introduction of the Supervisor	Mr. Ibrahim Waheed
<b>2.0</b>	<b>Topic 2: Discuss on the topics</b>	
2.1	Discuss what are the topics that can be selected and what topics that cannot be selected	All
<b>4.0</b>	<b>Topic 4: Plan Supervisor Meetings</b>	
4.1	Plan how to manage the meetings with the supervisor	All
4.2	Suggest Meeting times and days	Mr. Waris

<b>5.0</b>	<b><u>AOB</u></b>	
	There being no other matters, the meeting adjourned at 3:00PM	

Minuted by: Mohamed Simad

Verified by: Adam Waheed

## BIT304 Final Year Project I

HRMS

### Minutes No. 02

Date 14 Oct 2013

Time 1:00 PM

Venue Ligth Sky / Male'

#### Present :

Adam Waheed	Project Developer and designer
Mohamed Simad	Project Developer and UI Developer
Mariyam Maushamie	Project Manager,Strategic Planner and UI Devoper

No.	Matters Discussed	Action By
<b>1.0</b>	Topic 1: Choose topics	
1.1	Brain storm on the topics	All
1.2	Tabulate the topics	Maushamie
<b>2.0</b>	<b>Topic 2: Research on the Topics</b>	
2.1	Search internet	All
2.2	Call some Clients	Adam Waheed
<b>4.0</b>	<b>Topic 4: Select the suggestions</b>	
4.1	Take relevant topics	All
<b>5.0</b>	<b>AOB</b>	
	There being no other matters, the meeting adjourned at 1:00PM	

Minuted by: Mariyam Maushamie

Verified by: Adam Waheed

## BIT304 Final Year Project I

HRMS

### Minutes No. 03

Date 15 Oct 2013

Time 2:00 PM

Venue Ligh Sky / Male'

#### Present :

Adam Waheed	Project Developer and designer
Mohamed Simad	Project Developer and UI Developer
Mr. Musthafaa	Deputy Manager of Maldives Stock Exchange

#### Absent with apologies:

Mariyam MAUSHAMIEIE	Project Manager,Strategic Planner and UI Devoper
---------------------	--

No.	Matters Discussed	Action By
1.0	Topic 1: Request for a Project	
1.1	Talk about IT specialties need in Maldives Stock exchange	All
2.0	<b>Topic 2: Find the area to implement a system</b>	Mr. Musthafaa
4.0	<b>Topic 4: Finalize the topic go on a agreement</b>	Mr. Musthafaa
5.0	<b>Topic 4: Arrange next meeting</b>	All
5.0	<b>AOB</b>	
	There being no other matters, the meeting adjourned at 2:00 PM	

Minuted by: Mariyam Maushamie

Verified by: Adam Waheed

## BIT304 Final Year Project I

HRMS

### **Minutes No. 04**

Date 17 Oct 2013

Time 2:00 PM

Venue Ligh Sky / Male'

#### **Present :**

Adam Waheed	Project Developer and designer
Mohamed Simad	Project Developer and UI Developer
Mr. Musthafaa	Deputy Manager of Maldives Stock Exchange

#### **Absent with apologies:**

Mariyam MAUSHAMIEIE	Project Manager, Strategic Planner and UI Devoper
---------------------	---

<b>No.</b>	<b>Matters Discussed</b>	<b>Action By</b>
<b>1.0</b>	Topic 1: To Introducing about meeting	
1.1	Introduce each other and explain the Objectives of meeting Brief about Origination	Mr. Musthafaa
1.2	Deputy Manager Abdullah Mustafa Gave Introduction about Origination	Mr. Musthafaa
1.3	Explain Work Floor of the Origination	Mr. Musthafaa
1.4	Importance of Stock Exchange	Mr. Musthafaa
1.5	Why Stock Exchange exists	Mr. Musthafaa
2.0	<b>Topic 4:</b> Discuss Problems in Current HRMS System	
2.1	Difficulties of Managing employee in Manual System	Mr. Musthafaa
2.2	Difficulties of Manual Payroll System	Mr. Musthafaa
2.3	Time Consumption of Processes of Report Generation	Mr. Musthafaa
2.4	Difficulties of Getting past Information about Employee	Mr. Musthafaa
2.5	redundant of Data	Mr. Musthafaa
2.6	Inaccuracy of Report	Mr. Musthafaa
2.7	Insecure of Employee Information	Mr. Musthafaa
<b>5.0</b>	<b>AOB</b>	
	There being no other matters, the meeting adjourned at 2:00 PM	

Minuted by: Mohamed Simad

Verified by: Adam Waheed

## BIT304 Final Year Project I

HRMS

### Minutes No. 05

Date 18 Oct 2013

Time 3:00 PM

Venue Light Corner / Male'

Present :

Mr. Waris	Supervisor
Mr.Mauroof	Supervisor
Adam Waheed	Project Developer and designer
Mohamed Simad	Project Developer and UI Developer
Mariyam MAUSHAMIEIE	Project Manager,Strategic Planner and UI Devoper

No.	Matters Discussed	Action By
1.0	Topic 1: Introducing Project to the Supervisor	
1.1	Introduction of the Maldives Stock exchange	Adam Waheed
2.0	<b>Topic 2: Discussing About Current HRMS of Stock exchange</b>	
2.1	What will be the changes that can be brought	All
4.0	<b>Topic 4: Discussing about proposal for proposed projects</b>	All
5.0	<b>Topic 5: Arrange next Meeting Date</b>	Mr. Waris
6.0	<b>Topic 6: Discuss on Using Google doc to meet, share and comment on the project work</b>	
7.0	<b>AOB</b>	
	There being no other matters, the meeting adjourned at 3:00 PM	

Minuted by: Adam Waheed

Verified by: Mohamed Simad

## BIT304 Final Year Project I

HRMS

### Minutes No. 06

Date 21 Oct 2013

Time 7:30 PM – 8:45 PM

Venue Light Corner / Male'

Present :

Mr. Waris	Supervisor
Mr.Mauroof	Supervisor
Adam Waheed	Project Developer and designer
Mohamed Simad	Project Developer and UI Developer
Mariyam MAUSHAMIEIE	Project Manager,Strategic Planner and UI Devoper

<u>No.</u>	Matters Discussed	<u>Action</u> <u>By</u>
1.0	Topic 1: highlighted on risk management	Mr. Waris
1.1	Identify the Risks	All
1.2	Discuss the risks	Mr. Waris
2.0	<b>Topic 2: Discussing about literature review</b>	
2.1	What is literature review	Mr. Waris
2.2	How to do literature review	Mr. Waris
3.0	<b><u>AOB</u></b>	
	There being no other matters, the meeting adjourned at 7:30 PM	

Minuted by: Mariyam MAUSHAMIEIE

Verified by: Adam Waheed

## BIT305 Final Year Project II

Human Resource Management System (TinyHRMS)

### Minutes No. 01

Date : 2<sup>nd</sup> February 2013

Time : 1700hrs

Venue : Cyryx College, LightSky Campus

**Present :**

Mr. Naushad (*Supervisor*)

Adam Waheed

Mohamed Simadh

Mariyam Expanded

#### Absent with apologies:-

## Matters Discussed

### Topic 1: Project Briefing

Team members briefed newly appointed supervisor on project objectives, project plan, requirement and design specification of the project were also discussed with the supervisor.

### Topic 2: Recommendations by Supervisor

- To add business rules to the documentation, and discussed the importance of business rules in developing the project to the requirements of the client.
- To reconstruct the class diagram in a way that reflects the requirements of the client.
- To design the database based on business rules.
- To approve UI layouts from the client.
- To acknowledge team member roles in the acknowledgement.

There being no other matters, the meeting adjourned at 1815hrs.

**Minuted by:** Mariyam Maushamie

**Verified by:** Mr. Naushad (Supervisor)

## BIT305 Final Year Project II

Human Resource Management System (TinyHRMS)

### Minutes No. 02

Date : 14<sup>th</sup> February 2013

Time : 1745hrs

Venue : Cyryx College, Light Sky Campus

---

**Present :**

Mr. Naushad (*Supervisor*)

Adam Waheed

Mohamed Simadh

Mariyam Expanded

**Absent with apologies:-**

### Matters Discussed

#### Topic 1: Project Briefing

- The supervisor was briefed on the development of the system.
- Team members discussed about the development platforms with the supervisor.

#### Topic 2: Recommendations by Supervisor

- Advised to match the actual system with the documentation.

There being no other matters, the meeting adjourned at 1830hrs.

---

**Minuted by:** Mariyam Expanded

**Verified by:** Mr. Naushad (Supervisor)

## BIT305 Final Year Project II

Human Resource Management System (TinyHRMS)

### Minutes No. 03

Date : 16<sup>th</sup> February 2013

Time : 1700hrs

Venue : Cyryx College, LightSky Campus

**Present :**

Mr. Naushad (*Supervisor*)

Adam Waheed

Mohamed Simadh

Mariyam Expanded

#### Absent with apologies:-

## Matters Discussed

#### Topic 1: Project Briefing

- The supervisor was briefed on the development of the system.
- Discussed about validating leaves (Leave module).

#### Topic 2: Recommendations by Supervisor

- To add an annual calendar module.
- Add leave balance to the database.
- Discussed about the importance of UML diagrams in developing a system.

There being no other matters, the meeting adjourned at 1915hrs.

---

**Minuted by:** Mariyam Expanded

**Verified by:** Mr. Naushad (Supervisor)

**APPENDIX F:**

**Team Members' Contributions**

Student Name	Contribution				
	Proposal	Requirement analysis	System design	Implementation	Project Review
<b>ADAM WAHEED</b>	✓	✓	✓	✓	✓
<b>MOHAMED SIMAD</b>	✓	✓	✓	✓	✓
<b>MARIYAM MAUSHAMIEIE</b>	✓	✓	✓	✓	✓

## APPENDIX G: User Manual

### Login page

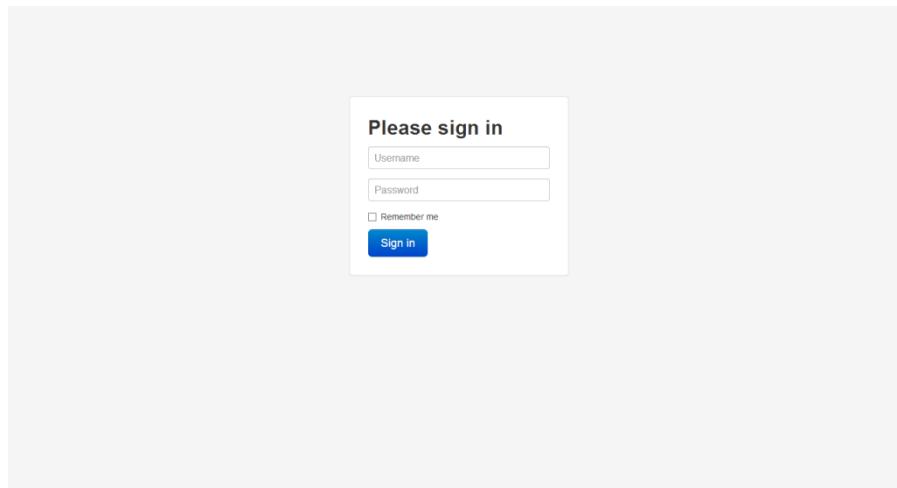


Figure 97 Login page

This page is used to log in to the system, user name and password are required. User has to enter the user name and password and press sign in. If user wants to remember user name and password tick on remember me check box.

### Dash Board

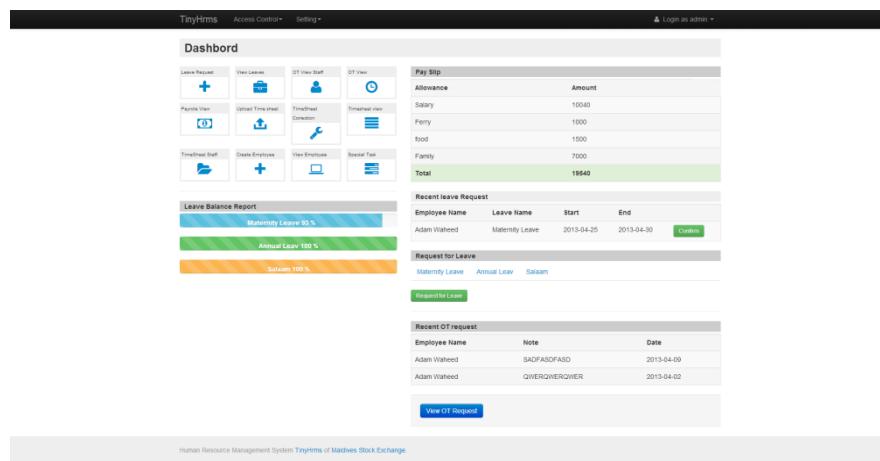


Figure 98 Dash Board

Dashboard shows all the widgets and module buttons

## User List

Employee Name	username	Email	Level
Hussain	dfw	dfg	Admin
Adam Waheed	admin	i@ivasdasd	User

Figure 99 User List

User list shows employee user names, email, level and password. It can be edited or deleted. It can be searched from the search box.

## Create User

Select Employee: Hussain Ali

User Name:

Email:

User Level: Admin

Password:

**Create** **Reset**

Figure 100 Create User

User can be created from create user. Employee is selected. Add username, password, email and user level. Press create button to save record. To reset text boxes press reset button.

## View User Group

User Group Name	Details
Admin	adminsdfasasd
User	User
Guest	asdfsad

Figure 101 View User Group

User group list shows user group names and details. It can be edited or deleted. It can be searched from the search box.

## Create User Group

User Group Name: User Gruppe

Details:

Create    Reset

Figure 102 Create User Group

User Group can be created from create User Group, Add User Group name and details. Press create button to save record. To reset text boxes press reset button.

## User Group Update

The screenshot shows the 'User Group Update' page in the TinyHrms application. The left sidebar has the following navigation links:

- Leave Management
  - Create leave
  - View Leave
  - Leave Request admin
  - Leave Request staff
  - Cancle leave Request
  - Leave Master
- Employee Management
- User Management
- Allowance Management
- Organization Setting

The main content area has a title 'User Group Update'. It contains two input fields: 'User Group Name' (set to 'User') and 'Details' (set to 'User'). Below these are two sections:

- Access Rights**: This section contains a grid of checkboxes for various permissions, grouped into three columns. Some checked items include: User\_Form, User\_View, UserGroupe\_Form, UserGroupe\_View, Allowance\_Form, Allowance\_View, Department\_View, Calender\_View, Designation\_View, designation\_Form, index, Shift\_Form, specialtask\_View, Leave\_Form, Leave\_View, LeaveRequest\_Form, LeaveRequest\_View, LeaveRequest\_Staff, LeaveRequest\_Status, LeaveRequest\_View, Shift\_View, timsheet\_View, UserGroupe\_Form, Leave\_View, Department\_Form, Calender, Employee\_Form, Employee\_view, LeaveRequest\_amend, specialtask\_Form, timsheet\_Error\_View.
- Footer**: Human Resource Management System TinyHrms of Maldives Stock Exchange

Figure 103 User Group Update

To update press Pencil icon which will go to update form enter User Group name and details. Press update button to save record. To reset text boxes press reset button. Rights can be assigned by ticking tick boxes which will give rights to user group.

## View Allowance

Name	Calculation type	Types	Amount
Ferry	Variable	General	120
food	Fixed	Special	1500
Family	Fixed	Special	700

Figure 104 View Allowance

Allowance list shows Allowance names, calculation type, types and amount. It can be edited or deleted. It can be searched from the search box.

## Create Allowance

Allowance Name	<input type="text" value="name"/>
Amount	<input type="text" value="amount"/>
Calculation Type	<input type="text" value="Fixed"/>
Type	<input type="text" value="Special"/>

Figure 105 Create Allowance

Allowance can be created from create Allowance. Add Allowance name, amount, calculation type and type. Press create button to save record. To reset text boxes press reset button.

## Allowance Update

The screenshot shows the 'Allowance Update' page. On the left is a sidebar with navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main area has a title 'Allowance Update'. It contains four input fields: 'Allowance Name' with the value 'food', 'Amount' with the value '1500', 'Calculation Type' set to 'Fixed', and 'Type' set to 'Special'. At the bottom are two buttons: 'Create' and 'Reset'.

**Figure 106 Allowance Update**

To update press Pencil icon which will go to update form enter Allowance name, amount, calculation type and type. Press update button to save record. To reset text boxes press reset button.

## Assign Allowance to Employee

The screenshot shows the 'Employee list' page with a modal window titled 'Select Allowance'. The modal displays a table with two rows. The first row is for 'food' with an amount of 1500 and an unchecked checkbox. The second row is for 'Family' with an amount of 700 and a checked checkbox. There is a search bar at the top of the modal and a 'Close' button at the bottom right.

**Figure 107 Assign Allowance to Employee**

To assign allowance to employee tick on the check box then press close button to assign. It can be searched from the search box.

## View Calendar event list

ID	Date	Details	Type
4	2013-02-03	asdasd	holiday
5	2013-02-19	asdasd	holiday
9	2013-02-20	asdasdas	holiday

Figure 108 Assign Allowance to Employee

Calendar event list shows date, details and type. It can be edited or deleted. It can be searched from the search box.

## Create Calendar event

Date:

Details:

Details: holiday

Create Reset

Figure 109 Create Calendar event

Calendar event can be created from create Calendar event. Add date, details and Type. Press create button to save record. To reset text boxes press reset button.

## Time Sheet List

The screenshot shows a web-based HR management system interface. At the top, there's a navigation bar with 'TinyHrms' and 'Home' buttons, and a 'Notification' dropdown. On the right, there's a 'Login as admin' button and a search bar. Below the navigation is a sidebar with several management modules: 'Leave Management' (with sub-options like Create leave, View Leave, etc.), 'Employee Management', 'User Management', 'Allowance Management', and 'Organization Setting'. The main content area is titled 'Time Sheet list' and contains a table with columns: Employee Name, Request Date, Date, Time, and Note. Two rows of data are visible: one for Adam Waheed on 2013-03-14 and another for Adam Waheed on 2013-03-15. A search bar for 'Username' is located at the top right of the table area. At the bottom of the page, a footer bar displays the text 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Employee Name	Request Date	Date	Time	Note
Adam Waheed	2013-03-14	2013-04-14	2013-04-14 09:00:00	i forgot to checking
Adam Waheed	2013-03-15	2013-04-15	2013-04-15 09:00:00	adasasdfsdfsdasf

Figure 110 Time Sheet List

Time Sheet list shows Employee name, request date, date, time and note. It can be edited or deleted. It can be searched from the search box.

## Time Sheet Reading

This screenshot shows the same HR management system interface as Figure 110. The 'Time Sheet list' table is present, but the focus is on a single row for Adam Waheed on 2013-04-15. This row is highlighted with a yellow background. Below the table, a message box displays the text 'Timesheet of Adam Waheed on 2013-04-15'. At the bottom of the page, there are 'Back' and 'Save' buttons. The footer bar at the bottom is identical to Figure 110.

Employee Name	Request Date	Date	Time	Note
Adam Waheed	2013-03-15	2013-04-15	2013-04-15 09:00:00	adasasdfsdfsdasf

Figure 111 Time Sheet Reading

Shows the time sheet of specific employee

## Time Sheet

The screenshot shows the 'Your Time sheet' section of the TinyHrms HRMS. On the left, there's a sidebar with navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main area displays a table of time entries from March 22 to April 20, 2013. The table has columns for date, IN (entry time), OUT (exit time), and Total Time. At the bottom of the table, it shows a total of 248.98 hours and a total pay of 10039.65.

Date	IN	OUT	Total Time
2013-03-22	08:00:00	09:30:00	8
2013-03-23	08:00:00	09:30:00	8
2013-03-24	08:00:00	09:30:00	8
2013-03-25	08:00:00	09:30:00	8
2013-03-26	08:00:00	09:30:00	8
2013-03-27	08:01:00	09:30:00	7.98
2013-03-28	08:00:00	09:30:00	8
2013-03-29	08:00:00	09:30:00	9
2013-03-30	08:00:00	09:30:00	8
2013-03-31	08:00:00	09:30:00	8
2013-04-01	08:00:00	09:30:00	8
2013-04-02	08:00:00	09:30:00	8
2013-04-03	08:00:00	09:30:00	8
2013-04-04	08:00:00	09:30:00	8
2013-04-05	08:00:00	09:30:00	8
2013-04-06	08:00:00	09:30:00	8
2013-04-07	08:00:00	09:30:00	8
2013-04-08	08:00:00	09:30:00	8
2013-04-09	08:00:00	09:30:00	8
2013-04-10	08:00:00	09:30:00	8
2013-04-11	08:00:00	09:30:00	8
2013-04-12	08:00:00	09:30:00	8
2013-04-13	08:00:00	09:30:00	8
2013-04-14	08:00:00	09:30:00	8
2013-04-15	08:00:00	09:30:00	8
2013-04-16	08:00:00	09:30:00	8
2013-04-17	08:00:00	09:30:00	8
2013-04-18	08:00:00	09:30:00	8
2013-04-19	08:00:00	09:30:00	8
2013-04-20			

Total Hours 248.98 Total Pay 10039.65

Human Resource Management System TinyHrms of Maldives Stock Exchange

**Figure 112 Time Sheet**

Time sheet contains the times enter into the finger machine.

## Update Attendance File

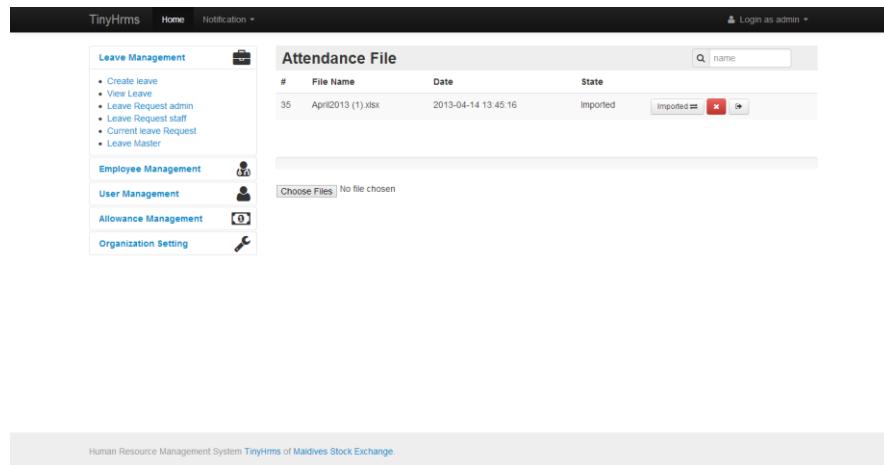


Figure 113 Update Attendance File

To upload attendance file choose file from the button and press enter

## View Leave list

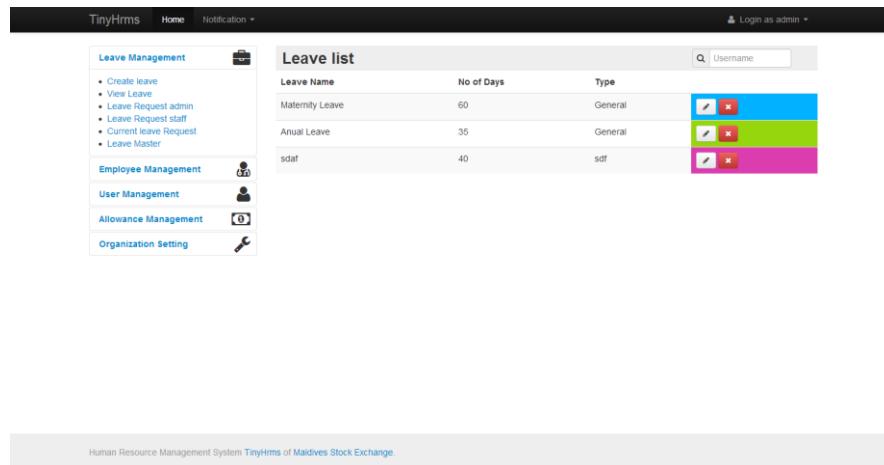


Figure 114 View Leave list

Leave list shows types of Leave names, number of days and types. It can be edited or deleted. It can be searched from the search box.

## Create Leave

The screenshot shows the 'Leave Create' page. On the left, there's a sidebar with 'Leave Management' (Create leave, View Leave, Leave Request admin, Leave Request staff, Current leave Request, Leave Master) and other management sections. The main area has a title 'Leave Create' and four input fields: 'Leave Name' (with placeholder 'Name'), 'No Of Days' (with placeholder 'noOfDay'), 'Type' (with placeholder 'type'), and 'colors' (with placeholder 'colors'). At the bottom are 'Create' and 'Reset' buttons.

**Figure 115 Create Leave**

Leave can be created from create Leave. Add name, no of days, type and colors. Press create button to save record. To reset text boxes press reset button.

## Leave request by Staff

The screenshot shows the 'Create Leave request' page. At the top, there's a summary table for 'Maternity Leave Report of Adam Waheed' with columns: Start Date, End Date, Total days, Friday / Saturday, Public Holiday, and Total Leave. The data shows: Start Date 2013-04-09, End Date 2013-04-13, Total days 5, Friday / Saturday 2, Public Holiday 0, Total Leave 3. Below this is a section titled 'Create Leave request' with fields: 'Select Leave Type' (set to Maternity Leave), 'Start Date' (2013-04-14), 'End Date' (2013-04-14), and 'Reason' (an empty text area). At the bottom are 'Create' and 'Reset' buttons.

**Figure 116 Leave request by Staff**

Leave can be requested by the staff by selecting leave type, start date, end date and reason. Press create to save the record.

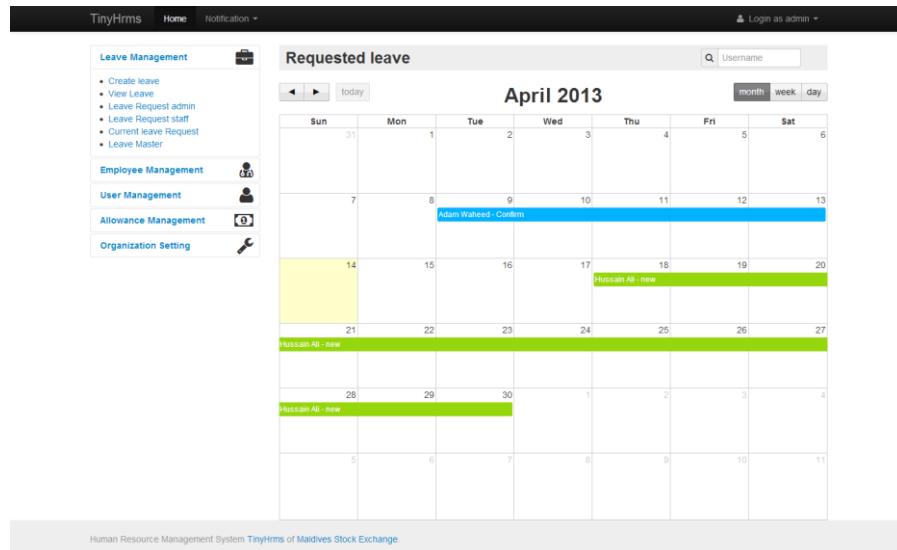
## Leave request by Admin

The screenshot shows the 'Leave Management' section of the TinyHrms interface. At the top, there's a navigation bar with 'Home' and 'Notification'. On the right, it says 'Login as admin'. Below the navigation is a sidebar with links like 'Create leave', 'View leave', etc., under 'Leave Management'. The main area has a title 'Maternity Leave Report of Adam Waheed'. It shows a table with columns: Start Date (2013-04-09), End Date (2013-04-13), Total days (5), Friday / Saturday (2), Public Holiday (0), and Total Leave (3). Below the table, it says 'Total leave taken : 3'. Underneath this, there's a 'Create Leave request' form. It includes fields for 'Select Employee' (set to Adam Waheed), 'Select Leave Type' (set to Maternity Leave), 'Start Date' (2013-04-14), 'End Date' (2013-04-14), and a 'Reason' input field. To the right of these fields is a calendar for April 2013, with the 14th highlighted. At the bottom of the form are 'Create' and 'Reset' buttons.

**Figure 117 Leave request by Admin**

Leave can be requested by the Admin by selecting Employee name, leave type, start date, end date and reason. Press create to save the record.

## Requested Leave



**Figure 118 Requested Leave**

Requested leaves by the employees can view from the Calendar.

## Amend Leave Request

The screenshot shows the 'Amend Leave Request' form. At the top, there's a navigation bar with 'TinyHrms', 'Home', 'Notification', and a login link. On the left is a sidebar with 'Leave Management' (Create leave, View leave, Leave Request admin, Leave Request staff, Current leave Request, Leave Master), 'Employee Management', 'User Management', 'Allowance Management', and 'Organization Setting'. The main form area has a title 'Amend Leave Request' and a reason field containing 'asdasdasd'. It includes date fields for 'Start Date' (2013-04-09) and 'End Date' (2013-04-13), a dropdown 'Type' set to 'Confirm', and buttons for 'Create' and 'Reset'. At the bottom, it says 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 119 Amend Leave Request

To Amend will go to Amend Leave Request form enter start date, end date and type.  
Press create button to save record. To reset text boxes press reset button.

## View Department

The screenshot shows the 'Department list' page. At the top, there's a navigation bar with 'TinyHrms', 'Home', 'Notification', and a login link. On the left is a sidebar with 'Leave Management', 'Employee Management', 'User Management', 'Allowance Management', and 'Organization Setting'. The main area has a title 'Department list' with a search bar for 'Username'. A table lists departments with columns 'id' and 'Name'. The data is as follows:

ID	Name	Action
3	Human Resource Management	
4	Finance Department	
5	Procurement	

At the bottom, it says 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 120 View Department

Department list shows Department names. It can be edited or deleted. It can be searched from the search box.

## Update Department

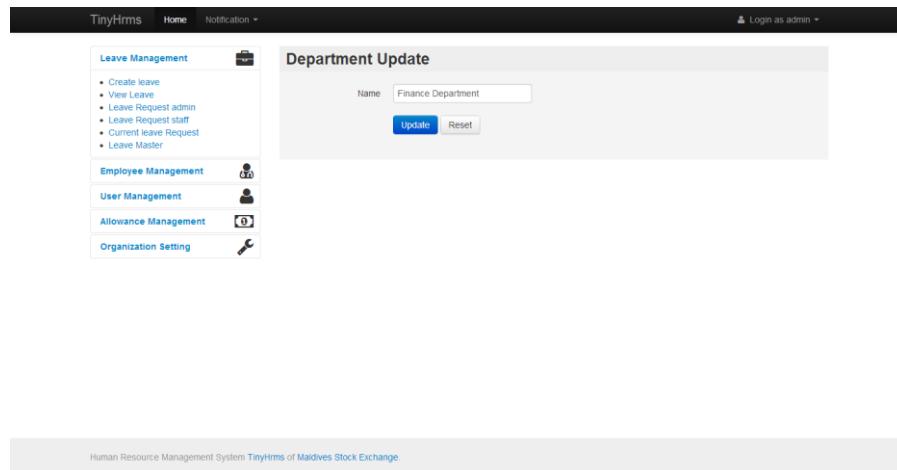


Figure 121 Update Department

To update press Pencil icon which will go to update form enter name. Press update button to update record. To reset text boxes press reset button.

## Assign Employee to department

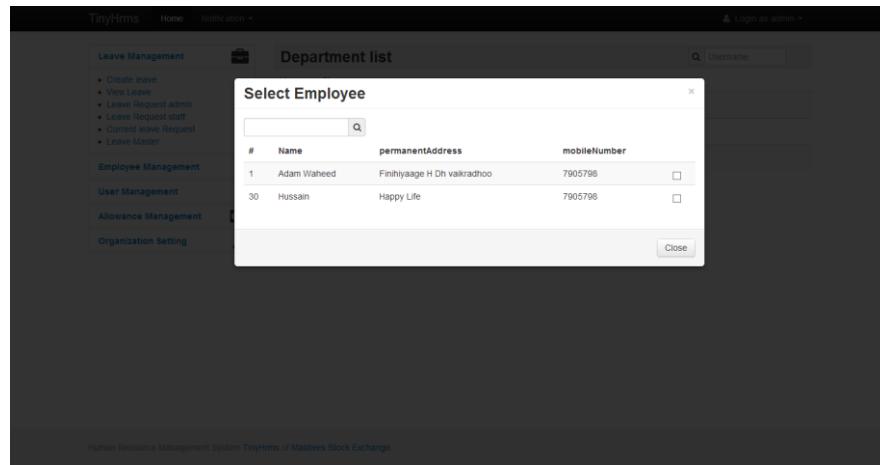


Figure 122 Assign Employee to department

To assign employee to department tick on the check box then press close button to assign a department.

## View Designation

The screenshot shows the 'Designation list' page. At the top, there's a navigation bar with 'TinyHrms', 'Home', 'Notification', and a login link. On the left, a sidebar lists 'Leave Management' (Create leave, View leave, Leave Request admin, Leave Request staff, Current leave Request, Leave Master), 'Employee Management' (User Management), 'Allowance Management' (Organization Setting). The main area is titled 'Designation list' and contains a table with two rows:

Name	Rank	Salary	Action
Director	2	10000	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
Manager	1	20000	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

A search bar at the top right says 'Username'. At the bottom, it says 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 123 View Designation

Designation list shows Designation names, rank and salary. It can be edited or deleted. It can be searched from the search box.

## Create Designation

The screenshot shows the 'Designation Create' page. At the top, there's a navigation bar with 'TinyHrms', 'Home', 'Notification', and a login link. On the left, a sidebar lists 'Leave Management' (Create leave, View leave, Leave Request admin, Leave Request staff, Current leave Request, Leave Master), 'Employee Management' (User Management), 'Allowance Management' (Organization Setting). The main area is titled 'Designation Create' and contains three input fields:

- Designation Name:
- Rank:
- Basic Salary:

At the bottom are 'Create' and 'Reset' buttons. At the very bottom, it says 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 124 Create Designation

Designation can be created from create Designation. Add name, rank and basic salary. Press create button to save record. To reset text boxes press reset button.

## Update Designation

The screenshot shows the 'Designation Update' page. On the left is a sidebar with navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main area is titled 'Designation Update' and contains three input fields: 'Designation Name' (set to 'Director'), 'Rank' (set to '2'), and 'Basic Salary' (set to '10000'). Below these fields are two buttons: 'Create' and 'Reset'.

**Figure 125 Update Designation**

To update press Pencil icon which will go to update form enter name, rank and Basic salary. Press update button to update record. To reset text boxes press reset button.

## Employee List

The screenshot shows the 'Employee list' page. On the left is a sidebar with navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main area is titled 'Employee list' and displays a table with four columns: 'Name', 'Permanent Address', 'Mobile Number', and 'Designation'. The table contains three rows of data. At the top right of the table is a search bar labeled 'Q: Fullname'. Each row has edit and delete icons.

**Figure 126 Employee List**

Employee list shows Employee names, Permanent address, mobile number and designation. It can be edited or deleted. It can be searched from the search box.

## Create Employee

The screenshot shows the 'Create Employee' page. On the left is a sidebar with navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main area is titled 'Create Employee' and contains the following fields:

- fullName: name
- permanentAddress: permanentAddress
- Join Date: [input field]
- Current Address: [input field]
- Id Card Number: idCardNumber
- Gender: Male (dropdown)
- mobile Number: mobileNumber
- date Of Birth: [input field]
- email Address: emailAddress
- designation Id: Director (dropdown)

At the bottom are 'Create' and 'Reset' buttons.

**Figure 127 Create Employee**

Employee can be created from create Employee. Add Full name, Permanent Address, Join Date, Current Address, ID card number, Gender, Mobile number, date of birth, email and Designation. Press create button to save record. To reset text boxes press reset button.

## View Shift

The screenshot shows the 'Shift list' page. On the left is a sidebar with navigation links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main area is titled 'Shift list' and displays a table of shifts:

#	Name	In	Out	Actions
5	Office	09:00:00	04:30:00	
6	Shop	08:30:00	06:00:00	

A search bar for 'name' is located at the top right. At the bottom is a footer with the text 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

**Figure 128 View Shift**

Shift list shows Shift names, in card time and out card time. It can be edited or deleted. It can be searched from the search box.

## Create Shift

The screenshot shows the 'Shift Create' form. At the top right is a 'Login as admin' button. On the left is a sidebar with 'Leave Management' (Create leave, View Leave, etc.), 'Employee Management', 'User Management', 'Allowance Management', and 'Organization Setting'. The main area has three input fields: 'Shift Name' (name), 'In Time' (in Time), and 'Out Time' (out Time). Below these are 'Create' and 'Reset' buttons. The footer says 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 129 Create Shift

Shift can be created from create Shift. Add name, In Time and Out Time. Press create button to save record. To reset text boxes press reset button.

## Update Shift

The screenshot shows the 'Shift Update' form. At the top right is a 'Login as admin' button. On the left is a sidebar with 'Leave Management' (Create leave, View Leave, etc.), 'Employee Management', 'User Management', 'Allowance Management', and 'Organization Setting'. The main area has three input fields: 'Shift Name' (Office), 'In Time' (09:00:00), and 'Out Time' (04:30:00). Below these are 'Create' and 'Reset' buttons. The footer says 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 130 Update Shift

To update press Pencil icon which will go to update form enter name, in time and out time. Press update button to update record. To reset text boxes press reset button.

## Add employee to shift

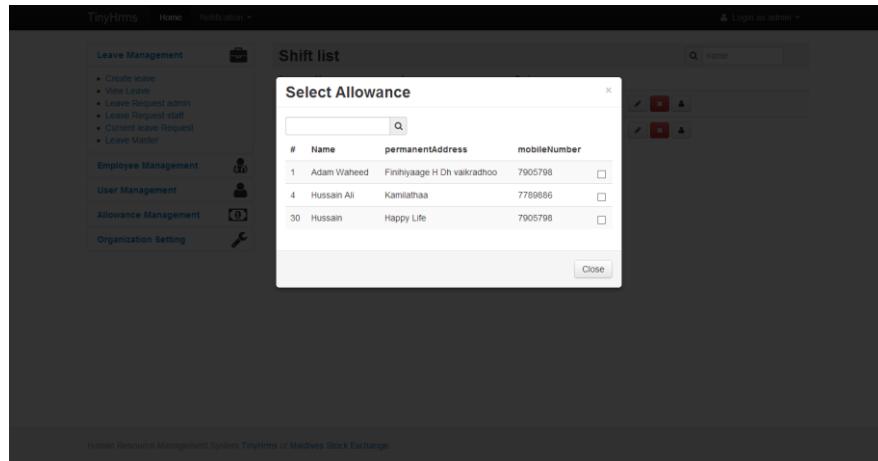


Figure 131 Add employee to shift

To assign employee to Shift tick on the check box then press close button to assign.

## Task List

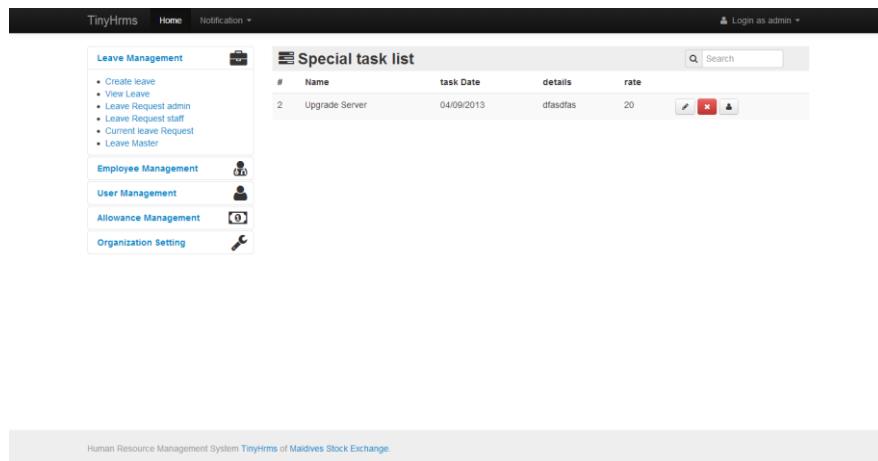


Figure 132 Task List

Task list shows Task names, Task date, details and out rate. It can be edited or deleted. It can be searched from the search box.

## Update Task

The screenshot shows a web-based HR management system interface. At the top, there's a navigation bar with 'TinyHrms', 'Home', 'Notification', and a login dropdown. Below the navigation is a sidebar with links for Leave Management, Employee Management, User Management, Allowance Management, and Organization Setting. The main content area is titled 'Special Task Update'. It contains four input fields: 'Special Task Name' (set to 'Upgrade Server'), 'Task Date' (set to '04/09/2013'), 'Details' (containing 'dfasdas'), and 'Rate' (set to '20'). At the bottom of the form are 'Update' and 'Reset' buttons. A footer at the bottom of the page reads 'Human Resource Management System TinyHrms of Maldives Stock Exchange'.

Figure 133 Update Task

To update press Pencil icon which will go to update form enter Task names, Task date, details and out rate. Press update button to update record. To reset text boxes press reset button.

## Assign Employee to task

The screenshot shows a modal dialog box titled 'Select Employee'. It has a search bar at the top and a table below it with columns for '#', 'Name', 'Contact Number', and 'Select'. There is a 'Close' button at the bottom right of the dialog. The background of the main page is dark, showing the 'Special task list' header and some menu items.

Figure 134 Assign Employee to task

To assign employee to task tick on the check box then press close button to assign.