|  |  |
| --- | --- |
|  | **MINISTRY OF EDUCATION AND TRAINING** |

**FPT UNIVERSITY**

|  |  |
| --- | --- |
| Capstone Project Document | |
| **Human Resources Management System Module** | |
| **Group 04** | |
| **Group members** | Nguyễn Việt Thắng – SE05071 (Leader)  Trần Đăng Hùng – SE62533  Nguyễn Học Huy – SE62370 |
| **Supervisor** | Ngô Đăng Hà An |
| **Project Manager** | Jason Ngo – Track&Roll Company |
| **Ext. Supervisor** | N/A |
| **Project Code** | HRMSM |

– **Brunei Darussalam, 20 Nov, 2018** –

**Table of Content**

[**List of Figure 7**](#_Toc522440201)

[**List of Table 11**](#_Toc522440202)

[**A. Introduction 16**](#_Toc522440203)

[1. Project Information 16](#_Toc522440204)

[2. Introduction 16](#_Toc522440205)

[3. Current Situation 16](#_Toc522440206)

[4. Problem Definition 16](#_Toc522440207)

[5. Proposed Solution 17](#_Toc522440208)

[5.1. Feature Functions 17](#_Toc522440209)

[5.2. Advantages 17](#_Toc522440210)

[5.3. Disadvantages 18](#_Toc522440211)

[6. Functional Requirements 18](#_Toc522440212)

[7. Roles and Responsibilities 19](#_Toc522440213)

[**B. Software Project Management Plan 20**](#_Toc522440214)

[1. Problem Definition 20](#_Toc522440215)

[1.1. Name of this Capstone Project 20](#_Toc522440216)

[1.2. Problem Abstract 20](#_Toc522440217)

[1.3. Project Overview 20](#_Toc522440218)

[1.4. Development Environment 21](#_Toc522440219)

[2. Project Organization 22](#_Toc522440220)

[2.1. Software Process Model 22](#_Toc522440221)

[2.2. Roles and Responsibilities 23](#_Toc522440222)

[2.3. Tools and Techniques 24](#_Toc522440223)

[3. Project Management Plan 24](#_Toc522440224)

[3.1. Product Backlog 24](#_Toc522440225)

[3.2. Deliverables 25](#_Toc522440226)

[4. Coding Convention 26](#_Toc522440227)

[**C. Software Requirement Specification 27**](#_Toc522440228)

[1. User Requirement Specification 27](#_Toc522440229)

[1.1 User Requirement 27](#_Toc522440230)

[1.2 HR Manager Requirement 27](#_Toc522440231)

[1.3 Employee Requirement 28](#_Toc522440237)

[2. System Requirement Specification 28](#_Toc522440238)

[2.1 External Interface Requirement 28](#_Toc522440239)

[2.2 System Overview Use Case 29](#_Toc522440240)

[2.3 List of Use Cases 31](#_Toc522440241)

[3. Software System Attributes 104](#_Toc522440242)

[3.1. Usability 104](#_Toc522440243)

[3.1.1. Graphic User Interface 104](#_Toc522440244)

[3.1.2. Usability 104](#_Toc522440245)

[3.2. Reliability 104](#_Toc522440246)

[3.3. Availability 104](#_Toc522440247)

[3.5. Maintainability 105](#_Toc522440248)

[3.6. Portability 105](#_Toc522440249)

[3.7. Performance 105](#_Toc522440250)

[4. Conceptual Diagram 105](#_Toc522440251)

[**D. Software Design Description 106**](#_Toc522440252)

[1. Design Overview 106](#_Toc522440253)

[2. System Architectural Design 106](#_Toc522440254)

[3. Component Diagram 107](#_Toc522440255)

[4. Detail Description 109](#_Toc522440256)

[4.1 Class Diagram 109](#_Toc522440257)

[4.2 Class Diagram Explanation 111](#_Toc522440258)

[4.3 Interaction Diagram 118](#_Toc522440259)

[5. User Interface Design 121](#_Toc522440260)

[5.1. Home Page 121](#_Toc522440261)

[5.2. Login Page 122](#_Toc522440262)

[5.3. Personal Interface Design 123](#_Toc522440263)

[5.4. Manager Interface Design 125](#_Toc522440264)

[5.4.1. Manager’s menu slidebar 125](#_Toc522440265)

[5.4.2. Staff management 126](#_Toc522440266)

[5.4.2.1. View staff list 126](#_Toc522440267)

[5.4.2.2. Create staff 127](#_Toc522440268)

[5.4.3. Customer management 128](#_Toc522440269)

[5.4.3.1. View customer list 128](#_Toc522440270)

[5.4.3.2. Create customer 129](#_Toc522440271)

[5.4.4. Product management 131](#_Toc522440272)

[5.4.4.1. View product list 131](#_Toc522440273)

[5.4.4.2. Create product 132](#_Toc522440274)

[5.4.5. Template Management 133](#_Toc522440275)

[5.4.5.1. View Template List 133](#_Toc522440276)

[5.4.5.2. View template detail 134](#_Toc522440277)

[5.4.5.3. Create template 135](#_Toc522440278)

[5.5. Accountant Interface Design 136](#_Toc522440279)

[5.5.1. Accountant’s menu slidebar 136](#_Toc522440280)

[5.5.2. Invoice management 137](#_Toc522440281)

[5.5.2.1. View invoice list 137](#_Toc522440282)

[5.5.2.2. Create invoice 139](#_Toc522440283)

[5.5.3. Proforma invoice management 142](#_Toc522440287)

[5.5.3.1. View proforma invoice list 142](#_Toc522440288)

[5.5.3.2. Create proforma invoice 143](#_Toc522440289)

[5.5.4. Liabilities management 146](#_Toc522440292)

[5.5.4.1. View liabilities list 146](#_Toc522440293)

[5.5.4.2. Liabilities detail 147](#_Toc522440294)

[5.5.4.3. Payment popup 149](#_Toc522440295)

[5.6. Customer Interface Design 149](#_Toc522440296)

[5.6.1. Customer’s menu slidebar 149](#_Toc522440297)

[5.6.2. Invoice list 150](#_Toc522440298)

[5.6.3. Proforma invoice list 151](#_Toc522440299)

[5.6.4. Liabilities 152](#_Toc522440300)

[5.6.5. Liabilities detail 153](#_Toc522440301)

[5.7. Admin Interface Design 155](#_Toc522440302)

[5.7.1. Admin’s menu slidebar 155](#_Toc522440303)

[5.7.2. Company management 156](#_Toc522440304)

[5.7.2.1. View company list 156](#_Toc522440305)

[5.7.2.2. Create company 157](#_Toc522440306)

[5.7.3. Account management 158](#_Toc522440307)

[5.7.3.1. View admin list 158](#_Toc522440308)

[5.7.3.2. Create Admin 160](#_Toc522440309)

[5.8. Windows Application 161](#_Toc522440310)

[6. Database Design 163](#_Toc522440311)

[6.1. Entity Relationship Diagram (ERD) 163](#_Toc522440312)

[6.2. Entity Dictionary 165](#_Toc522440313)

[**E. System Implementation & Test 166**](#_Toc522440314)

[1. Introduction 166](#_Toc522440315)

[1.1. Overview 166](#_Toc522440316)

[1.2. Test Approach 166](#_Toc522440317)

[2. Database Relationship Diagram 167](#_Toc522440318)

[2.1. Physical Diagram 167](#_Toc522440319)

[2.2. Data Dictionary 169](#_Toc522440320)

[3. Test Plan 169](#_Toc522440321)

[3.1. Features To Be Tested 169](#_Toc522440322)

[3.2. Features Not To Be Tested 169](#_Toc522440323)

[4. System Testing Test Case 169](#_Toc522440324)

[4.1. Test Case 169](#_Toc522440325)

[**F1. Software User’s Manual 177**](#_Toc522440326)

[1. Installation Guide 177](#_Toc522440327)

[1.1. Hardware Requirement 177](#_Toc522440328)

[1.2. Software Requirement 177](#_Toc522440329)

[2. User Manual 177](#_Toc522440330)

[2.1. Web Application 177](#_Toc522440331)

[2.2. Windows Application 186](#_Toc522440332)

[**G. Appendix 199**](#_Toc522440340)

# **List of Figure**

[Figure 1. Scrum framework 17](#_Toc530970211)

[Figure 2. System overview use case 23](#_Toc530970212)

[Figure 3. HR Master overview use cases 25](#_Toc530970213)

[Figure 4. <HR Master> Manage events 26](#_Toc530970214)

[Figure 5. <HR Master> Create new event One time. 27](#_Toc530970215)

[Figure 6. <HR Master> Create new event Recurring 29](#_Toc530970216)

[Figure 7. <HR Master> Update event One time 30](#_Toc530970217)

[Figure 8. <HR Master> Update event Recurring 32](#_Toc530970218)

[Figure 9. <HR Master> Remove events 34](#_Toc530970219)

[Figure 10. <HR Master> View yearly events 36](#_Toc530970220)

[Figure 11. <HR Master> View events in monthly and dailys 37](#_Toc530970221)

[Figure 12. <HR Master> Create leave request 39](#_Toc530970222)

[Figure 13. <HR Master> Update leave status 41](#_Toc530970223)

[Figure 14. <HR Master> Remove leave request 42](#_Toc530970224)

[Figure 15. <HR Master> View leave requests in yearly 44](#_Toc530970225)

[Figure 16. <HR Master> View leave requests in monthly and daily 45](#_Toc530970226)

[Figure 17. <HR Master> Search leave request using filter 47](#_Toc530970227)

[Figure 18. Employee overview use cases 49](#_Toc530970228)

[Figure 19. <Employee> Login 50](#_Toc530970229)

[Figure 20. <Employee> My Profile 51](#_Toc530970230)

[Figure 21. <Employee> View employee profile 52](#_Toc530970231)

[Figure 22. <Employee> Mobile checkin attendance 54](#_Toc530970232)

[Figure 23. <Employee> Mobile checkout attendance 56](#_Toc530970233)

[Figure 24. <Employee> View attendance report. 58](#_Toc530970234)

[Figure 25. <Employee> Create leave request. 60](#_Toc530970235)

[Figure 26. <Employee> View leave request history 61](#_Toc530970236)

[Figure 27. <Employee> Manage my claim. 63](#_Toc530970237)

[Figure 28. <Employee> Create claim request. 65](#_Toc530970238)

[Figure 29. <Employee> View claim history 66](#_Toc530970239)

[Figure 30. Conceptual diagram 69](#_Toc530970240)

[Figure 31. System architectural design 70](#_Toc530970241)

[Figure 32. Component diagram 71](#_Toc530970242)

[Figure 33. Class diagram 74](#_Toc530970243)

[Figure 34. <Sequence Diagram><HR Master> Create events 81](#_Toc530970244)

[Figure 35. <Sequence Diagram><HR Master > Create leave requests 81](#_Toc530970245)

[Figure 36. <Sequence Diagram><HR Master> View all events 82](#_Toc530970246)

[Figure 37. <Sequence Diagram><HR Master> Update event information 82](#_Toc530970247)

[Figure 38. <Sequence Diagram><HR Master> Delete events 83](#_Toc530970248)

[Figure 39. <Sequence Diagram><HR Master> Update status of leave request 83](#_Toc530970249)

[Figure 40. <Sequence Diagram><HR Master> Delete leave request 84](#_Toc530970250)

[Figure 41. <Sequence Diagram><Employee> View profile employee. 84](#_Toc530970251)

[Figure 42. <Sequence Diagram><Employee> Checkin. 85](#_Toc530970252)

[Figure 43. <Sequence Diagram><Employee> Checkout. 85](#_Toc530970253)

[Figure 44. <Sequence Diagram><Employee> Create leave request. 86](#_Toc530970254)

[Figure 45. <Sequence Diagram><Employee> Create claim request. 86](#_Toc530970255)

[Figure 46. <Sequence Diagram><Employee> View leave report. 87](#_Toc530970256)

[Figure 47. <Sequence Diagram><Employee> View claim report. 87](#_Toc530970257)

[Figure 48. <Sequence Diagram><Employee> View attendance. 88](#_Toc530970258)

[Figure 49. <Sequence Diagram><Employee> View my attendance. 88](#_Toc530970259)

[Figure 50. Home Page (Year view) 89](#_Toc530970260)

[Figure 51. View event list 90](#_Toc530970261)

[Figure 52. Create Event one time. 90](#_Toc530970262)

[Figure 53. Create Event Recurring 91](#_Toc530970263)

[Figure 54. View list Apply Leave 92](#_Toc530970264)

[Figure 55. Create Apply Leave 94](#_Toc530970265)

[Figure 56. Employee’s menu slidebar 95](file:///C:\Users\Snowfox1991\Desktop\Documents%20for%20capstone%20project\ECMM\ECMM_FullV12_UpdatingTestCase.docx#_Toc530970266)

[Figure 57. View Attendance 96](file:///C:\Users\Snowfox1991\Desktop\Documents%20for%20capstone%20project\ECMM\ECMM_FullV12_UpdatingTestCase.docx#_Toc530970267)

[Figure 58. Mobile checkin 97](file:///C:\Users\Snowfox1991\Desktop\Documents%20for%20capstone%20project\ECMM\ECMM_FullV12_UpdatingTestCase.docx#_Toc530970268)

[Figure 59. Mobile check out 98](#_Toc530970269)

[Figure 60. View leave history 99](file:///C:\Users\Snowfox1991\Desktop\Documents%20for%20capstone%20project\ECMM\ECMM_FullV12_UpdatingTestCase.docx#_Toc530970270)

[Figure 61. Create leave request 100](file:///C:\Users\Snowfox1991\Desktop\Documents%20for%20capstone%20project\ECMM\ECMM_FullV12_UpdatingTestCase.docx#_Toc530970271)

[Figure 62. Entity relationship diagram 101](#_Toc530970272)

[Figure 63. Physical database 105](#_Toc530970273)

[Figure 64. <HR Master> View event and apply leave in yearly calendar. 116](#_Toc530970274)

[Figure 65. <HR Master> Create event one time (1) 117](#_Toc530970275)

[Figure 66. < HR Master > Create event one time (2) 117](#_Toc530970276)

[Figure 67. < HR Master> Create recurring event (1) 118](#_Toc530970277)

[Figure 68. <HR Master> Create recurring event (2) 118](#_Toc530970278)

[Figure 69. <HR Master > Create recurring event (3) 119](#_Toc530970279)

[Figure 70. <HR Master> Create recurring event (4) 119](#_Toc530970280)

[Figure 71. <HR Master> Create recurring event (5) 120](#_Toc530970281)

[Figure 72. <HR Master> Select Year 121](#_Toc530970282)

[Figure 73. <HR Master> Select Month 122](#_Toc530970283)

[Figure 74. . <HR Master> Delete event and leave request (1) 122](#_Toc530970284)

[Figure 75. <HR Master> Delete event and leave request (2) 123](#_Toc530970285)

[Figure 76. <HR Master> Delete event and leave request (3) 123](#_Toc530970286)

[Figure 77. <HR Master> Delete event and leave request (4) 124](#_Toc530970287)

[Figure 78. <HR Master> Change Apply Leave Status (1) 125](#_Toc530970288)

[Figure 79. <HR Master> Change Apply Leave Status (2) 125](#_Toc530970289)

[Figure 80. <HR Master> Change Apply Leave Status (3) 126](#_Toc530970290)

[Figure 81. <HR Master> Filter in month view. 126](#_Toc530970291)

[Figure 82. <User Manual> Mobile application - Login 128](#_Toc530970292)

[Figure 83. <User Manual> Mobile application - Menu 129](file:///C:\Users\Snowfox1991\Desktop\Documents%20for%20capstone%20project\ECMM\ECMM_FullV12_UpdatingTestCase.docx#_Toc530970293)

[Figure 84. <User Manual> Mobile application - View Profile 129](file:///C:\Users\Snowfox1991\Desktop\Documents%20for%20capstone%20project\ECMM\ECMM_FullV12_UpdatingTestCase.docx#_Toc530970294)

[Figure 85. <User Manual> Mobile application - Attendance 130](file:///C:\Users\Snowfox1991\Desktop\Documents%20for%20capstone%20project\ECMM\ECMM_FullV12_UpdatingTestCase.docx#_Toc530970295)

[Figure 86. <User Manual> Mobile application - View Attendance 131](file:///C:\Users\Snowfox1991\Desktop\Documents%20for%20capstone%20project\ECMM\ECMM_FullV12_UpdatingTestCase.docx#_Toc530970296)

[Figure 87. <User Manual> Mobile application - Apply Leave 132](file:///C:\Users\Snowfox1991\Desktop\Documents%20for%20capstone%20project\ECMM\ECMM_FullV12_UpdatingTestCase.docx#_Toc530970297)

[Figure 88. <User Manual> Mobile application - My Leave 133](file:///C:\Users\Snowfox1991\Desktop\Documents%20for%20capstone%20project\ECMM\ECMM_FullV12_UpdatingTestCase.docx#_Toc530970298)

[Figure 89. <User Manual> Mobile application - Apply Claim 134](file:///C:\Users\Snowfox1991\Desktop\Documents%20for%20capstone%20project\ECMM\ECMM_FullV12_UpdatingTestCase.docx#_Toc530970299)

[Figure 90. <User Manual> Mobile application - My Claim 135](file:///C:\Users\Snowfox1991\Desktop\Documents%20for%20capstone%20project\ECMM\ECMM_FullV12_UpdatingTestCase.docx#_Toc530970300)

**List of Table**

[Table 1. Roles and responsibilities 15](#_Toc530970301)

[Table 2. Software requirements 18](#_Toc530970302)

[Table 3. Roles and responsibilities details 19](#_Toc530970303)

[Table 4. Tools and techniques 19](#_Toc530970304)

[Table 5. Product backlog 21](#_Toc530970305)

[Table 6. Deliverables 21](#_Toc530970306)

[Table 7. <HR Master> View events 28](#_Toc530970307)

[Table 8. <HR Master> Create events one time 29](#_Toc530970308)

[Table 9. <HR Master> Create events recurring 31](#_Toc530970309)

[Table 10. <HR Master> Update event one time 33](#_Toc530970310)

[Table 11. <HR Master> Update event Recurring 35](#_Toc530970311)

[Table 12. <HR Master> Remove events 37](#_Toc530970312)

[Table 13. <HR Master> View yearly events 38](#_Toc530970313)

[Table 14. <HR Master> View events and leave request in monthly 39](#_Toc530970314)

[Table 15. <HR Master> Create leave request 41](#_Toc530970315)

[Table 16. <HR Master> Update leave status 43](#_Toc530970316)

[Table 17. <HR Master> Remove leave request 45](#_Toc530970317)

[Table 18. <HR Master> View leave requests in yearly 46](#_Toc530970318)

[Table 19. <HR Master> View leave requests in monthly and yearly 47](#_Toc530970319)

[Table 20. <HR Master> Search leave request using filter. 49](#_Toc530970320)

[Table 21. <Employee> Login 52](#_Toc530970321)

[Table 22. <Employe> Logout 53](#_Toc530970322)

[Table 23. <Employee> View employee profile. 54](#_Toc530970323)

[Table 24. <Employee> Mobile checkin attendance 56](#_Toc530970324)

[Table 25. <Employee> Mobile checkout attendance. 59](#_Toc530970325)

[Table 26. <Employee> View attendance history. 60](#_Toc530970326)

[Table 27. <Employee> Create leave request. 62](#_Toc530970327)

[Table 28. <Employee> View leave request history. 63](#_Toc530970328)

[Table 29. <Employee> Manage my claim. 65](#_Toc530970329)

[Table 30. <Employee> Create claim requests. 67](#_Toc530970330)

[Table 31. <Employee> View claim history. 68](#_Toc530970331)

[Table 32. Component diagram dictionary 73](#_Toc530970332)

[Table 33. Class Diagram Dictionary 77](#_Toc530970333)

[Table 34. <Class Diagram Attributes> User 77](#_Toc530970334)

[Table 35. Class Diagram Methods> User 78](#_Toc530970335)

[Table 36. <Class Diagram Attributes> Attendance 79](#_Toc530970336)

[Table 37. <Class Diagram Methods> Attendance 79](#_Toc530970337)

[Table 38. < Class Diagram Attributes> Employee 79](#_Toc530970338)

[Table 39. <Class Diagram Methods> Employee 79](#_Toc530970339)

[Table 40. <Class Diagram Attributes> EventType 80](#_Toc530970340)

[Table 41. <Class Diagram Methods> EventType 80](#_Toc530970341)

[Table 42. <Class Diagram Attributes> Events 80](#_Toc530970342)

[Table 43. <Class Diagram Methods> Events 80](#_Toc530970343)

[Table 44. <Class Diagram Attributes> LeaveRequest 81](#_Toc530970344)

[Table 45. <Class Diagram Methods> LeaveRequest 81](#_Toc530970345)

[Table 46. <Class Diagram Attributes> My Claim 81](#_Toc530970346)

[Table 47. <Class Diagram Methods> My Claim 81](#_Toc530970347)

[Table 48. <Buttons/Hyperlinks> Home page (Year View) 91](#_Toc530970348)

[Table 49. <Fields> View event list 91](#_Toc530970349)

[Table 50. <Buttons/Hyperlinks> View event list 91](#_Toc530970350)

[Table 51. <Fields> Create Event one time. 91](#_Toc530970351)

[Table 52. <Buttons/Hyperlinks> Create Event one time. 92](#_Toc530970352)

[Table 53. <Fields> Create Event Recurring 92](#_Toc530970353)

[Table 54. <Buttons/Hyperlinks> Create Event Recurring 93](#_Toc530970354)

[Table 55. <Fields> View list Apply Leave. 94](#_Toc530970355)

[Table 56. <Buttons/Hyperlinks> View list Apply Leave. 95](#_Toc530970356)

[Table 57. <Fields> Create Apply Leave 95](#_Toc530970357)

[Table 58. <Buttons/Hyperlinks> Create Apply Leave 96](#_Toc530970358)

[Table 59. <Fields> Employee’s menu slidebar 96](#_Toc530970359)

[Table 60. <Buttons/Hyperlinks> Employee’s menu slidebar 97](#_Toc530970360)

[Table 61. <Fields> Attendance interface 97](#_Toc530970361)

[Table 62. <Fields> Check in interface 98](#_Toc530970362)

[Table 63. <Buttons/Hyperlinks> Check in 98](#_Toc530970363)

[Table 64. <Fields> Check out interface 99](#_Toc530970364)

[Table 65. <Buttons/Hyperlinks> Check out 99](#_Toc530970365)

[Table 66. View leave history 100](#_Toc530970366)

[Table 67. <Buttons/Hyperlinks> View leave history 100](#_Toc530970367)

[Table 68. <Fields> Create leave request 101](#_Toc530970368)

[Table 69. <Buttons/Hyperlinks> Create leave request 101](#_Toc530970369)

[Table 70. Entity dictionary 104](#_Toc530970370)

[Table 71. Database dictionary 108](#_Toc530970371)

[Table 72. <Test Case> View events 111](#_Toc530970372)

[Table 73. View lists of events 112](#_Toc530970373)

[Table 74. <Test Case> Manage leave request 114](#_Toc530970374)

[Table 75. <Test Case> Apply leave request for type 1 and type 2 events 116](#_Toc530970375)

[Table 76. Server requirement 117](#_Toc530970376)

[Table 77. Backend requirement 117](#_Toc530970377)

[Table 78. <User Manual> View event and apply leave in yearly calendar. 118](#_Toc530970378)

[Table 79. <HR Master> Create new one time event. 119](#_Toc530970379)

[Table 80. <User manual> Create recurring event. 122](#_Toc530970380)

[Table 81. <User manual> Select Year 122](#_Toc530970381)

[Table 82. <User manual> Select Month 123](#_Toc530970382)

[Table 83. <User manual> Filter in month view. 125](#_Toc530970383)

[Table 84. <User manual> Change Apply Leave Status. 127](#_Toc530970384)

[Table 85. <User manual> Filter in month view. 128](#_Toc530970385)

[Table 86. <User Manual> Mobile application - Login 129](#_Toc530970386)

[Table 87. <User Manual> Mobile application - View Attendance 131](#_Toc530970387)

[Table 88. <User Manual> Mobile application - Check in Online 132](#_Toc530970388)

[Table 89. <User Manual> Mobile application - Apply Leave 133](#_Toc530970389)

[Table 90. <User Manual> Mobile application - View My Leave 134](#_Toc530970390)

[Table 91. <User Manual> Mobile application - Apply Claim 135](#_Toc530970391)

[Table 92. <User Manual> Mobile application - View My Claim 136](#_Toc530970392)

# **A. Introduction**

## **Project Information**

* Project name:  **Human Resources Management System Module**
* Project Code: **HRMSM**
* Product Type: **Web Application + Mobile Application**
* Start Date: **Wednesday, October 3rd, 2018.**
* End Date: **Friday, November 30th, 2018 (Estimated).**

## **Introduction**

In this document, we introduce a solution for people who works on events in companies, manages leave request and take attendance of employees. Nowadays, companies have lots of internal events for meeting other companies, for visiting industries,… That they usually manage on paper calendar. And also when the employee has some problem, they usually send request to leave by message for fast, employee take attendance also like that. So it’s very hard to get summarize report on working days of the employee

Therefore, we are building a web application, which helps Companies to solve the current problems. Companies can manage the events and view it in web, and if user wants to view event summary, they can click on the day occurring the events. Not only for summarizing the events, managing the leave request is also very easy for those who apply for leaving on that day. And a Mobile application, which helps employee to take attendance or send apply leave request easier.

This document also describes our working process in 4 months including our perspective on the system, component designs and detailed core workflows. We hope our solution will help to resolve some problems from businesses in Brunei and other developing countries.

## **Current Situation**

There are three problems that the customer meet:

- The first problem is managing the events manually on the calendar that they have to mark on that days occurring the events. Until there are too many events in one day, they don’t have enough space to write. It will lead to time-consuming and effective of work is decreased a lot.

- The second problem that when an employee wants to apply for the leave request in other day by paper or by message. But it will get inconvenience when tracking the message, or spending time on signing the paper to apply the request. This will waste the time too much, the manager’s work is affected and low down the cost of work.

- The third problem: Change outdated attendance system.

## **Problem Definition**

Manage events in paper calendar and leave request through paper or message presents a number of challenges that result in unnecessary costs while reducing the work progress efficiency:

* **Time – consuming:** Manage event will spend time to write down on paper calendar. But it is not only one event, even more than 10 events in one day with the small area written down. But the problem bigger than is to Confirm the leave request of an employee would spend lots of times when confirm for many employees with lots of papers.
* **Difficulty in managing and monitoring:** When managing events in paper with small size and have to write down to another place, it is easy to get lost the information. Also papers for confirm leave request can easy to get lost for a few people.
* **Using too old procedure to apply for leave:** Some company still use the punch card or employee have to write down on paper, or even the employee wants to request for leave and they have to wait for a long time to get the application approved or rejected.

## **Proposed Solution**

The proposed solution is to build a Web Application for Admin (in here is the HR Manager) to manage the Events and Leave Requests of the company. Also the Mobile Application is used for Employe in Company to send apply leave request, take attendance and manage information of each person through account. The mobile application will be updated as soon as possible.

### **Feature Functions** For the web application, because this is a part of a module that are developed for new feature and it is used by HR Manager, so the login will be skipped and moved to HR Manager Page:

* View the summary information of events and leave request in yearly calendar view.
* View the event information in monthly and detailed information in daily.
* View the request for leave application of the employee and the status of the employee’s request.
* Manage events (if events occur in a short term duration or repeated events in daily, weekly, monthly or yearly).
* Manage the employee’s leave request and update the status of the request application form.

For the mobile application, the employee will be given an account from the HR Manager so that they can access to the mobile feature:

* Login – Logout for employee
* View employee profile using image and QR Code
* Send leave request of the employee using their account.
* View leave request history and status of the request
* Check attendance of the employee.
* View the history attendance of the employee.

### **Advantages**

* Efficient management: easy to lookup and manage events and leave request information.
* Time saving: minimize administrative procedures.
* Shorten the procedure: The employee can check their status through mobile app and don’t have to wait for long.
* Manage the attendance of the employee better: old technology like punch card or using paper will be replaced using mobile so that they can view all the employee working on that day and the status when check-in.

### **Disadvantages**

* Because it is written in web application so the user have to view in website, so it is uncomfortable to view in website in mobile if HR Manager is outside.
* The problem of internet will be very hard if the employee use the mobile application in area without active internet connection.
* The performance of the mobile application could be slow due to many records and material usage.
* Also, smartphone is a problem for those who use the old smartphone or don’t use smartphone.

## **Functional Requirements**

**Functional requirements are listed as below:**

* **HR Manager:**
  + Manage events.
  + Manage leave request.
* **Employee:**
  + Apply leave request form.
  + View profile of the employee (using image and QR Code).
  + View history of the leave request and status of the request.
  + Check attendance of the employee.
  + View attendance history of the employee.

## **Roles and Responsibilities**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Full name** | **Role** | **Position** | **Contact** |
| 1 | Ngô Đăng Hà An | Supervisor | Supervisor | AnNDH@fpt.edu.vn |
| 2 | Jason Ngo – T&R Company | Project Manager | Project Manager | jasonngo@tracknroll.com |
| 2 | Nguyen Viet Thang | Developer | Leader | Thangnvse05071@fpt.edu.vn |
| 3 | Tran Dang Hung | Developer | Member | Hungtdse62533@fpt.edu.vn |
| 4 | Nguyen Hoc Huy | Developer | Member | HuyNHSE62370@fpt.edu.vn |

Table 1. Roles and responsibilities

# **B. Software Project Management Plan**

## **Problem Definition**

* 1. **Name of this Capstone Project**
* Official name: **Human Resources Management System Module**
* Abbreviation: HRMSM
  1. **Problem Abstract**

Nowadays, many people are using online calendar services for managing events, also there are lots if application are downloaded and used for effective in managing a lot. Also in the calendar, the manager can view the request for leave of the employee without passing through the paper or by message. Companies can enhance the performance in managing and scheduling faster, effective of work is increased a lot.

Besides that, employee when using the mobile application, they wil be given an account for login to use the feature of mobile application – which is increased the effective the work and reduce the time consuming when waiting for approved the leave request and also check for attendance to calculate for their salary, and some other features.

* 1. **Project Overview**
     1. **Current Situation**

Some problems encountered in this project.

**Business behaviors:** depending on how a business is organized, the work of managing human resources and managing leave request can become complicated.

**Mobile application:** Because one employee can exist in database of more than 1 company so the check in for attendance could be a problem.

* + 1. **The Proposed System**

The proposed system is an enhanced calendar management module application. This is a small part of an HR System developed by T&R Company. The system focus on HR Manager Web Application management page and a Mobile Application for Employee to view the events, take attendance and send the request for leave.

In details, the system provides following features:

**For HR Managers:**

* + Managers can manage events.
  + Managers can manage request for leave of the employee.
  + Managers can approve the status of the leave request.

**For Employee:**

* + Employee can log in to system
  + Employee can view their profile information.
  + Employee can apply their leave request and view their leave history.
  + Employee can check attendance and view their attendance history.
    1. **Boundaries of the System**

This section supposes that users of the system have stable Internet connection while using.

**The system can:**

* + - Allow managers to manage events information and leave request information.
    - Allow users to send request for leave.
    - Allow managers to approve the request status.
    - Allow user to take attendance online.

**The system cannot:**

* + - Run without internet connection and database.
    1. **Future Plan**

In the future, we want to upgrade and develop the system with following features:

* Support employee management on website.
* Develop the employee registration in website and mobile.
* Develop new updated on using in offline mode (and will synchronize when having active internet connection).
  1. **Development Environment**

#### **Hardware Requirements** The Human Resources Management System Module (called HRMSM) is a small part of a big system developed by our team and cooperated with Track&Roll Company. All the hardware server settings is sponsored by Track&Roll Co.

#### **Software Requirements**

|  |  |  |
| --- | --- | --- |
| **Software** | **Name/ Version** | **Description** |
| **Operating System** | Windows 8 or above | Operating system and platform for development |
| **IDE** | Sublime Text, PhpStorm, Visual Studio Code. | Used for implementing website |
| **Design Model Tool** | Visual Paradigm Enterprise | Used for creating model and diagram |
| **DBMS** | MySQL | Used for creating & managing database |
| **Document storage** | Google Drive | Used for storing documents |
| **Store and manage source code** | Github | Used for storing source code |
| **Communication** | Facebook, email,whatsapp | Used for exchanging information, online meeting, communication |
| **Task management** | Trello | Used for managing task of project. |
| **Web browser** | Chrome 28 or higher | Testing browser |

Table 2. Software requirements

1. **Project Organization**
2. **Software Process Model**

In this project, we choose Scrum, an agile framework that allows step by step to set up a plan successfully. We choose this model because of the following reasons:

* Time-saving: daily meetings ensure that the process is at correct stage, as established at the beginning of the project.
* Easy to use: suitable model for small and medium project.
* Fast response to changes: product owner may change requirements or extend / reduce scope and we can adapt better.
* Encourage teamwork: roles and tasks are divided and assigned efficiently.
* Functional tests are frequent in the entire process.

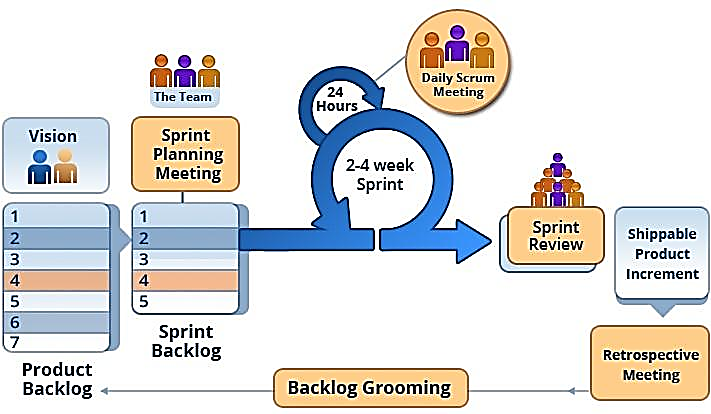


Figure 1. Scrum framework

1. **Roles and Responsibilities**

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Full name** | **Role in Group** | **Responsibilities** |
| 1 | Ngô Đăng Hà An | Supervisor | * Receive the report and send feedback on the process of work * Supervise on the work and process of team. |
| 2 | Jason Ngo | Project Manager | * Give questions for the problem uncleared and specified the solution. * Approved or Given feedback on the updated solution that team recommend |
| 3. | Nguyen Viet Thang | Team Leader,  BA,  Developer,  Tester | * Manage process * Clarify requirements * Design GUI * Code * Write document and report * Test |
| 4. | Tran Dang Hung | Team member,  Developer,  Tester | * Clarify requirements * Design GUI * Code * Write document and report * Create test plan * Test |
| 5 | Nguyen Hoc Huy | Team member,  Developer,  Tester | * Plan Scrum implementations * Design database * Write document and report * Code * Test |

Table 3. Roles and responsibilities details

**Tools and Techniques**

|  |  |
| --- | --- |
| **Tool/Technique** | **Name/Version** |
| Front-end | HTML5, CSS3, Bootstrap, Angular,FireBase |
| Back-end | PHP, Laravel Framework Framework |
| Design Pattern | Requests and Repository Pattern (Laravel Framework) |
| Managing Database | XAMPP Server with MySQL |
| Task Management | Trello (and require meetup in lab) |
| Source Control | Sourcetree (server github.com) / Github |
| Modeling Tool | StarUML v2.8.1 |
| Web Browser | Google Chrome |

Table 4. Tools and techniques

## **Project Management Plan**

1. **Product Backlog**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Story**  **ID** | **Features** | **Task**  **ID** | **Task description** | **Sprint** |
| 1 | Create Product Backlog | 1.1 | Create product backlog | 1 |
| 2 | Create Introduction document | 2.1 | Create introduction document | 1 |
| 2.2 | Review introduction document | 1 |
| 3 | Learning PHP,Angular, Ionic, Design Partern. | 3.1 | Learning PHP (Laravel framework), Angular (Ionic mobile framework) | 2 |
| 4 | Create UI mockups | 4.1 | Search for suitable UI | 2 |
| 4.2 | Mobile UI | 2 |
| 4.3 | Web Calendar UI | 2 |
| 4.4 | HR Manager UI function | 2 |
| 4.5 | Employee mobile UI | 2 |
| 4.6 | Attendance and Leave request UI | 2 |
| 5 | Create Software Project Management Plan | 5.1 | Problem definition | 2 |
| 5.2 | Project organization | 2 |
| 5.3 | Project management plan | 2 |
| 5.4 | Coding convention | 2 |
| 6 | Create Software Requirement Specifications | 6.1 | User requirement specification | 3 |
| 6.2 | External interface requirement | 3 |
| 6.3 | Use case diagram | 2 |
| 6.4 | Software system attributes | 3 |
| 6.5 | Database diagram | 3 |
| 7 | Create Software Design Description | 7.1 | Design overview | 4, 5 |
| 7.2 | System architectural design | 4, 5 |
| 7.3 | Component diagram | 4, 5 |
| 7.4 | Detailed description of components | 4, 5 |
| 7.5 | Sequence diagram | 4, 5 |
| 7.6 | User interface design | 2 |
| 7.7 | Database design | 4 |
| 7.8 | Coding | 5 đến 11 |
| 8 | Create Software Test Documentation | 8.1 | Test Plan | 12 |
| 8.2 | Test Cases | 12 |
| 8.3 | Checklists | 12 |
| 9 | Software User’s Manual | 9.1 | Installation Guide | 13 |
| 9.2 | User’s Guide | 13 |

Table 5. Product backlog

1. **Deliverables**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Deliverable** | **Delivery date** | **Delivery location** | **Note** |
| 1 | Introduction Document, Task list | 13/10/2018 | FPT-UBD Lab | Report No.1 |
| 2 | Software Project Management Plan | 13/10/2018 | FPT-UBD Lab | Report No.2 |
| 3 | Software Requirements Specification | 20/10/2018 | FPT-UBD Lab | Report No.3 |
| 4 | Software Design Description | 30/10/2018 | FPT-UBD Lab | Report No.4 |
| 5 | Software Test Documentation Guide Implementation (Coding) | 10/11/2018 | FPT-UBD Lab | Report No.5 |
| 6 | Software User’s Manual | 20/11/2018 | FPT-UBD Lab | Report No.6 |

Table 6. Deliverables

\* For each sprint, deliverables are potentially shippable products, which can be a part of documents or runnable software application which is implementation of some program features.

## **Coding Convention**

* **Naming conventions:**
  + Use pascal case for class names and method names.
  + Use camel case for method arguments and local variables.
  + Do not use underscore in identifiers. Except: prefix private static variables with an underscore.
  + Use noun or noun phrases to name a class.
  + Prefix interfaces with letter ‘I’.
* **Layout conventions:**
  + Tabs must be set exactly 4 spaces.
  + Avoid lines longer than 80 characters.
  + Vertically align curly brackets.
  + Write only one statement per line.
  + Write only one declaration per line.
  + Add one blank line between method definitions and property definitions.
* **Declaration:**
  + Use implicit type ‘var’ for local variable declarations. Except: primitive types (int, string, double, etc.) use predefined names.
  + Organize namespaces with a clearly defined structure.
* **Commenting conventions:**
  + Place comment on a separate line.
  + Begin comment text with an uppercase letter and end with a period.
  + Add one space between comment delimiter (//) and comment text.

# **C. Software Requirement Specification**

**1. User Requirement Specification**

**1.2 HR Master Requirement**

HR Manager is a person from development team. HR Manager can use following functions:

* View all events in Calendar
* Manage Events (One time and Recurring)
* Manage Leave Request of an employee.
* View events and leave request summary in yearly/monthly/daily.

**1.3 Employee Requirement**

Employee is a person who works for the company. Employee can use following functions:

* Login to mobile using account set up by HR Manager
* Check-in and check out Attendance.
* View attendance history.
* View the employee profile.
* View events and leave request summary in yearly/monthly/daily.

**2. System Requirement Specification**

**2.1 External Interface Requirement**

**2.1.1 User Interface**

* + Web application use Calendar interface, suitable for all users working in different kinds of work..
    1. **Hardware Interface**: N/A

**2.1.3 Software Interface**

* The Web application: work with browsers Firefox (v52 or higher), Chromes (v28 or higher), Internet Explorer (v10 or above) or with any web browser that supports HTML5 & CSS3

**2.1.4 Communication Protocol**

* It uses REST API + HTTP protocol for communication between Front-end (using Angular) and Back-end server-side (using Laravel framework of PHP).

**2.2 System Overview Use Case**

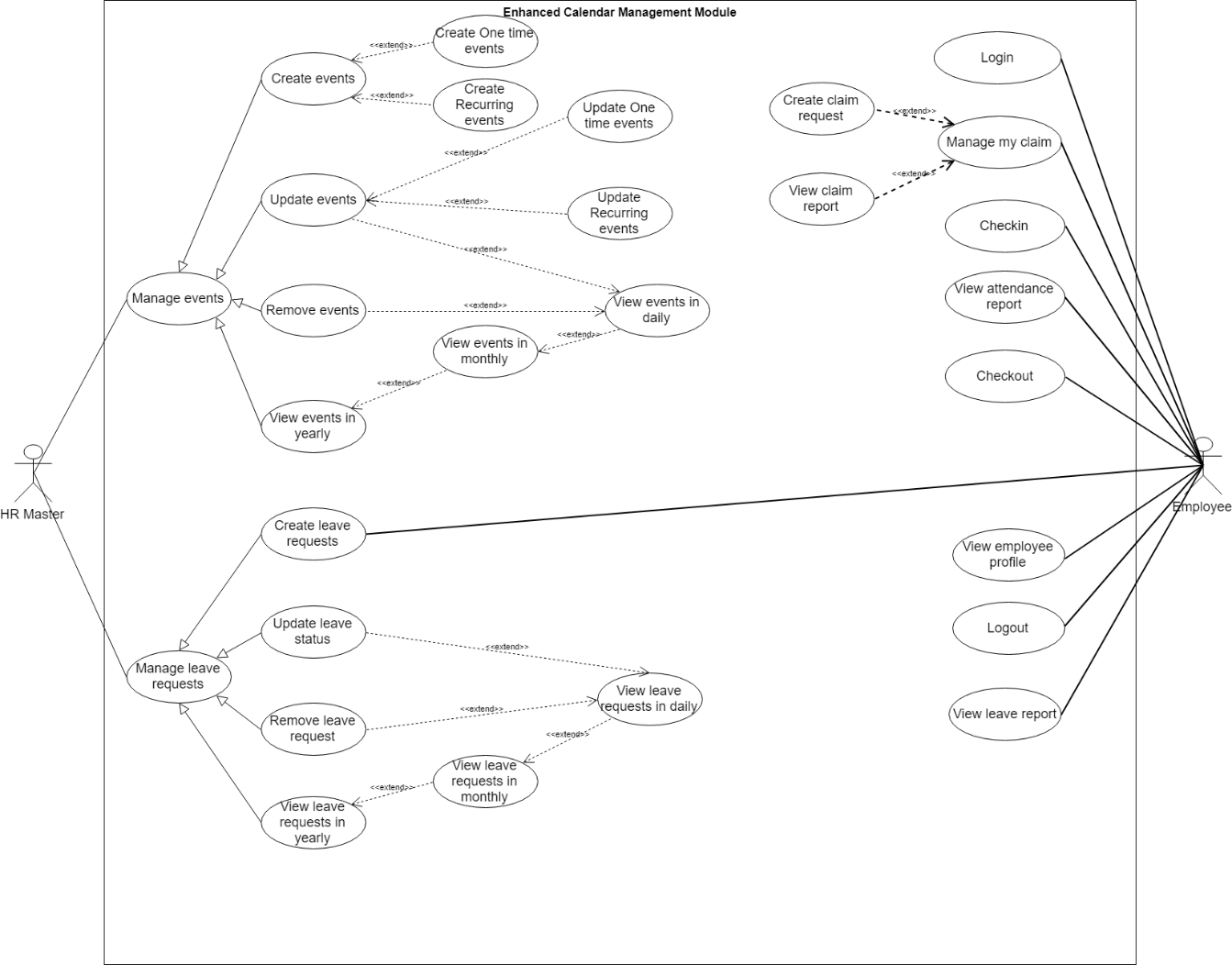


Figure 2. System overview use case

2.3 List of Use Cases

* + - 1. HR Master Overview Use Cases

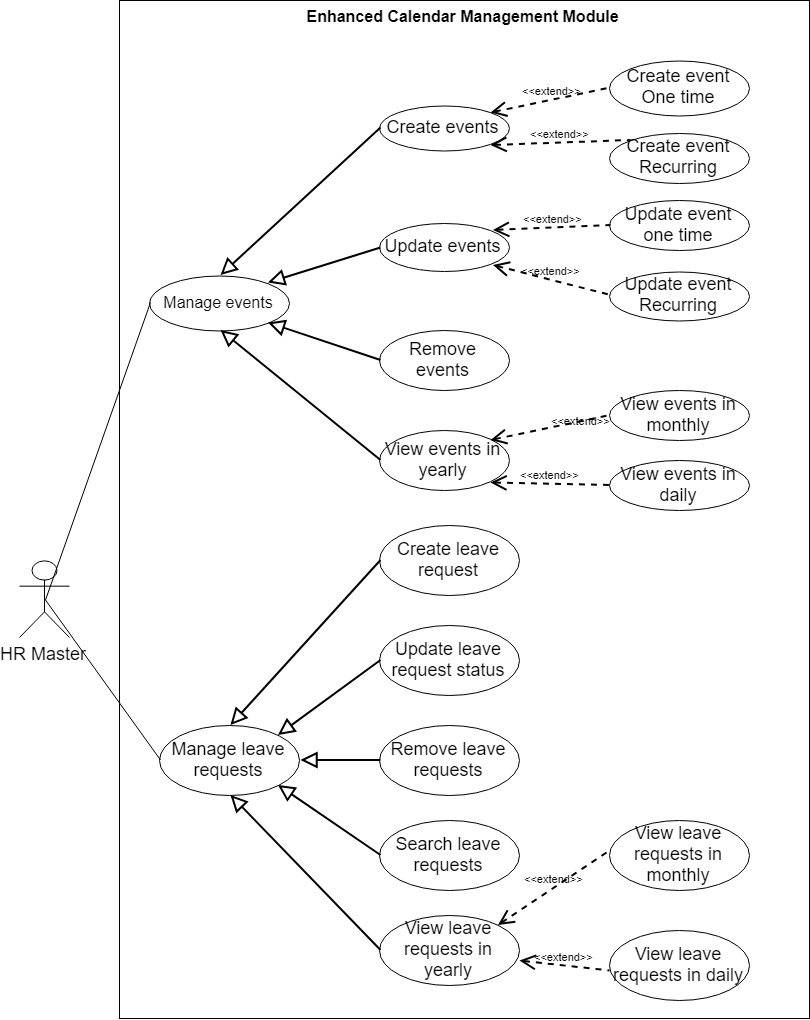


Figure 3. HR Master overview use cases

* + - 1. <HR Master> View events

**Use Case Diagram**

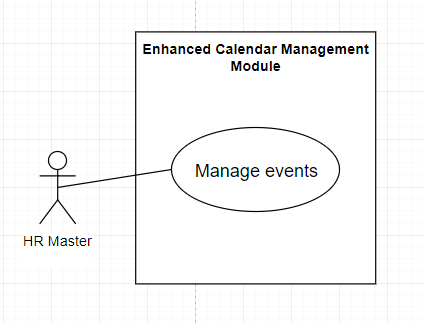


Figure 4. <HR Master> Manage events

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_01** | | | |
| **Use Case No.** | 01 | **Use Case Version** | 1.0 |
| **Use Case Name** | View events | | |
| **Author** | HuyNH | | |
| **Date** | 20/11/2018 | **Priority** | Normal |
| **Actor:** HR Master  **Summary:**  This use case allows HR Master to view all events in Manage event page.  **Goal:**  HR Master can view all events in manage events page.  **Triggers:**  Click on "View event" in menu slidebar.  **Preconditions:**  **Post Conditions:**  **Success:** Navigate to manage events page.  **Fail:** No event data in manage events page.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on “View event” in menu slidebar. | Navigate to manage events page.  [Exception 1] |   **Alternative:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Database occurs exception. | No data is shown in UI. HR Master will inform to IT Team for checking. |   **Relationships:** N/A  **Business Rules:** N/A | | | |

Table 7. <HR Master> View events

* + - 1. <HR Master> Create events One time

**Use Case Diagram**

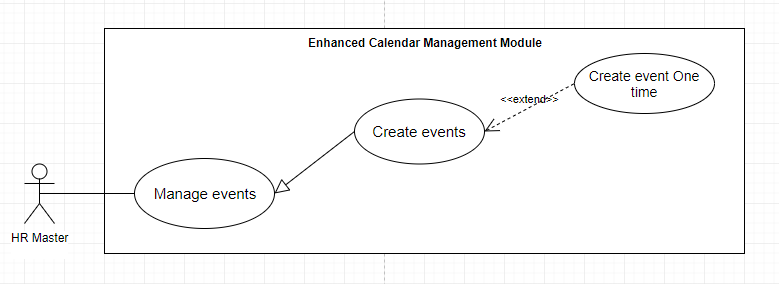


Figure 5. <HR Master> Create new event One time.

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_02** | | | |
| **Use Case No.** | **02** | **Use Case Version** | 1.1 |
| **Use Case Name** | Create events one time | | |
| **Author** | HuyNH | | |
| **Date** | 20/11/2018 | **Priority** | Normal |
| **Actor:** HR Master  **Summary:**  This use case allows HR Master to create new One time Event.  **Goal:**  New event is added to database and shown in UI.  **Triggers:**  Click on "New Event(One Time)" button in menu on the top right corner.  **Preconditions:**  **Post Conditions:**  **Success:** The event is added into DB success and shown in Calendar UI.  **Fail:** Show popup error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on “New Event(One time)” button in menu box on the top right. | Popup Create Event One time form appear. The HR Master will input: - Choose the event type. - Input the event title: required to input.  - Choose start date and end date in small Calendar box.  - Input remarks for the event: required to input. | | 2 | HR Master input all the information |  | | 3 | HR Master click on “Submit” button. | [Exception 1]  [Exception 2]  Create new event with inputted information.  Redirect to current page and view in yearly. |   **Alternative:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Required fields are missing or fields are invalid. | Show instruction message below those input fields. | | 2 | Database occurs exception. | Show error message and ask user to try again later. |   **Relationships:**  Generalized from Manage Events List use case and extended from Create new event List use case.  **Business Rules:**  Validation:   * Choose event type – There are 3 types of events: Non working Day (public holiday), Blocked Day for Leave (no apply leave request on that day), and Event Notification. * Choose event have start and end day. * Event must have a title, cannot be blank. | | | |

Table 8. <HR Master> Create events one time

* + - 1. <HR Master> Create event Recurring

**Use Case Diagram**



Figure 6. <HR Master> Create new event Recurring

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_03** | | | |
| **Use Case No.** | **03** | **Use Case Version** | 1.1 |
| **Use Case Name** | Create events recurring | | |
| **Author** | HuyNH | | |
| **Date** | 20/11/2018 | **Priority** | Normal |
| **Actor:** HR Master  **Summary:**  This use case allows HR Master to create new event recurring.  **Goal:**  New event is added to database and shown in UI.  **Triggers:**  Click on "New Event (Recurring)" button in menu box on the top right corner.  **Preconditions:**  **Post Conditions:**  **Success:** The event is added into DB success and shown in Calendar UI in yearly view.  **Fail:** Show popup error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on “New Event New Event (Recurring)” button in menu box on the top right. | Popup Create Event Recurring form appear. The HR Master will input: - Choose the event type. - Input the event title: required to input.  - Choose the frequency of the event occuring (daily, weekly, monthly, yearly).  - Input remarks for the event: required to input. | | 2 | HR Master input all the information |  | | 3 | HR Master click on “Submit” button. | [Exception 1]  [Exception 2]  Create new event with inputted information.  Redirect to current page and view in yearly. |   **Alternative:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Required fields are missing or fields are invalid. | Show instruction message below those input fields. | | 2 | Database occurs exception. | Show error message and ask user to try again later. |   **Relationships:**  Generalized from Manage Events List use case and extended from Create new event List use case.  **Business Rules:**  Validation:   * Choose event type - there are 3 types of events: | | | |

Table 9. <HR Master> Create events recurring

* + - 1. <HR Master> Update event One Time

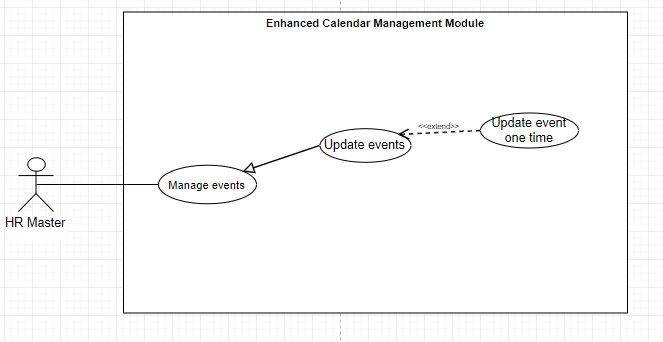


Figure 7. <HR Master> Update event One time

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_04** | | | |
| **Use Case No.** | **04** | **Use Case Version** | 1.1 |
| **Use Case Name** | Update event One time | | |
| **Author** | HuyNH | | |
| **Date** | 20/11/2018 | **Priority** | Normal |
| **Actor:** HR Master  **Summary:** This use case allows HR Master to change the information of the existed One time events.  **Goal:**  New event information is updated.  **Triggers:**  Click on “Update” button (pencil icon) in the Calendar Day View UI and choose days and events want to update/Choose “view events” in left menu slidebar and press “Update” button (pencil icon) on the events that HR Master wants to update.  **Preconditions:**  The chosen event is already existed in the Calendar UI.  **Post Conditions:**  **Success:** New event information is updated.  **Fail:** Show error message.  **Main Success Scenario: If HR Master update the event in day view UI.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Month view | The Calendar Month view UI appear with all the events occurring on days of that month. | | 2 | Choose the day that occur the event | The Day view UI appear to view all events in that day with Update button | | 3 | Choose the event that HR Master wants to update and press Update (pencil icon) button | The popup update form dialog appear | | 4 | When HR Master input all the required field and press Update . | [Exception]  Update event successfully.  Popup message “Updated successfully” and return to View Daily Calendar UI. |   **If HR Master update the event in Manage Event page**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on “View Event” in the left menu slidebar | The manage event page appears with all the events occuring. | | 2 | Choose the event that HR Master wants to update the information and press Update (pencil icon) button | The popup update form dialog appear with the form:  - Event type: optional. - Event title: optional input.  - Start\_date and end\_date: optional choice.  - Remarks: optional. | | 3 | When HR Master input all the required field and press “Update”. | [Exception 1]  Update event successfully.  Popup message “Updated successfully” and redirect to Manage event page. |   **Alternative:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Required fields are missing or fields are invalid. | Show instruction message below those input fields. | | 2 | Database occurs exception. | Show error message and ask user to try again later. |   **Relationships:**  Generalized from Manage Events List use case.  **Business Rules:**  Validation:   * Update the start date and end date of the Recurring and One Time event. * Optional for update the event title, event type and remarks. | | | |

Table 10. <HR Master> Update event one time

* + - 1. <HR Master> Update event Recurring

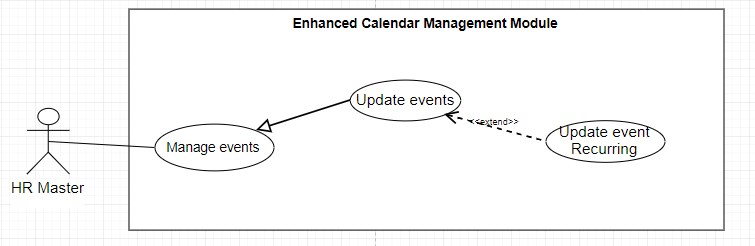


Figure 8. <HR Master> Update event Recurring

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_05** | | | |
| **Use Case No.** | **05** | **Use Case Version** | 1.1 |
| **Use Case Name** | Update event Recurring | | |
| **Author** | HuyNH | | |
| **Date** | 20/11/2018 | **Priority** | Normal |
| **Actor:** HR Master  **Summary:** This use case allows HR Master to change the information of the existed events.  **Goal:**  New event information is updated.  **Triggers:**  Click on “Update” button (pencil icon) in the Calendar Day View UI or choose (View all event in left menu slidebar) and choose days and events want to update/Press view all events and choose “Update” button (pencil icon).  **Preconditions:**  The chosen event is already existed in the Calendar UI.  **Post Conditions:**  **Success:** New event information is updated.  **Fail:** Show error message.  **Main Success Scenario: If HR Master update the event in day view UI.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Month view | The Calendar Month view UI appear with all the events occurring on days of that month. | | 2 | Choose the day that occur the event | The Day view UI appear to view all events in that day with Update button | | 3 | Choose the event that HR Master wants to update and press Update (pencil icon) button | The popup update form dialog appear, HR Master will change the information:  - Event type: optinal.  - Occuring frequency: optinal.  - Event remarks: optinal. - Event title: optinal. | | 4 | When HR Master input all the required field and press Update . | [Exception 1]  Update event successfully.  Popup message “Updated successfully” and return to View Daily Calendar UI. |   **If HR Master update the event in Manage Event page**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on View event at the Menu slidebar on the top left corner | The manage event page appear with all the events occuring. | | 2 | Choose the event that HR Master wants to update the information and press Update (pencil icon) button | The popup update form dialog appear, HR Master will change the information:  - Event type: optinal.  - Occuring frequency: optinal.  - Event remarks: optinal. - Event title: optinal. | | 3 | When HR Master input all the required field and press “Update”. | [Exception 1]  Update event successfully.  Popup message “Updated successfully” and redirect to Manage event page. |   **Alternative:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Required fields are missing or fields are invalid. | Show instruction message below those input fields. | | 2 | Database occurs exception. | Show error message and ask user to try again later. |   **Relationships:**  Generalized from Manage Events List use case.  **Business Rules:**  Validation:   * Update the start date and end date of the Recurring and One Time event. * Optional for update the event title, event type and remarks. | | | |

Table 11. <HR Master> Update event Recurring

* + - 1. <HR Master> Remove events

**Use Case Diagram**

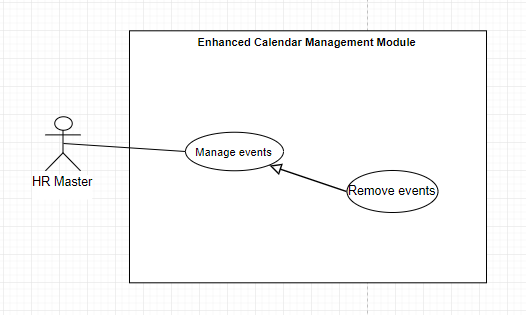
****

Figure 9. <HR Master> Remove events

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE –UC\_ ECMM\_06** | | | |
| **Use Case No.** | **06** | **Use Case Version** | 1.1 |
| **Use Case Name** | Remove event | | |
| **Author** | HuyNH | | |
| **Date** | 20/11/2018 | **Priority** | Normal |
| **Actor:** HR Master  **Summary:** This use case allows HR Master to remove event out of the Calendar UI.  **Goal:** The events is removed out of database and Calendar UI.  **Triggers:** Click on “Delete” button (Recycle bin icon) in the Calendar Month View UI and choose days and events want to update/Press view all events and choose “Delete” button (Recycle bin icon).  **Preconditions:**  The chosen event already exists in the system.  **Post Conditions:**  **Success:** The chosen event is removed out of Calendar UI and database.  **Fail:** Show error message.  **Main Success Scenario: If HR Master remove the event in day view UI.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Month view in menu slidebar. | The Calendar Month view UI appear with all the events with count number occurring on days of that month. | | 2 | Choose the day that occur the event | The Day view UI appear to view all events in that day with Update button | | 3 | Choose the event that HR Master wants to remove | The popup show confirm delete message appear. | | 4 | When HR Master confirm to delete the event. | [Exception 1]  Delete event successfully.  Popup message “Deleted successfully” and return to View Daily Calendar UI. |   **If HR Master update the event in Manage Event page**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on View event at the Menu slidebar. | The manage event page appear with all the events occuring. | | 2 | Choose the event that HR Master wants to delete (Recycle bin icon) button. | The popup show confirm delete message appear | | 3 | When HR Master confirm to delete the event. | [Exception 1]  Delete event successfully.  Popup message “Deleted successfully” and redirect to Manage event page. |   **Alternative:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Database occurs exception. | Show error message and ask user to try again later. |   **Relationships:**  Generalized from Manage event List use case.  **Business Rules:** N/A | | | |

Table 12. <HR Master> Remove events

* + - 1. <HR Master> View events in year

**Use Case Diagram**

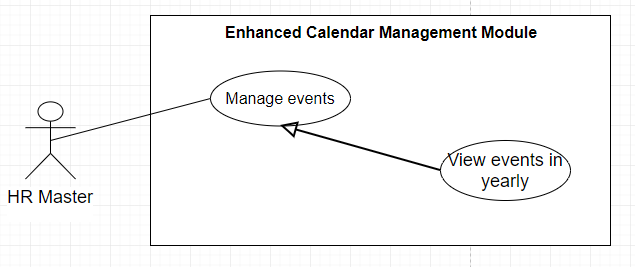


Figure 10. <HR Master> View yearly events

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ECMM\_07** | | | |
| **Use Case No.** | 07 | **Use Case Version** | 1.1 |
| **Use Case Name** | View yearly event | | |
| **Author** | HuyNH | | |
| **Date** | 20/11/2018 | **Priority** | Normal |
| **Actor:**  HR Master  **Summary:**  This use case allows HR Master to view the summary of all events in yearly view.  **Goal:**  View all events in Yearly view.  **Triggers:** Press on View Yearly in the Menu slidebar.  **Preconditions:**  The events records already exists in the system.  **Post Conditions:**  **Success:** View events (in blue color) in Calendar UI. (the red dot represents for more than 1 event and 1 leave request on that day of month)  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Press on View Yearly in the menu slidebar. | View all the events into Calendar UI. User can hover on the day of month to view all events in that day. |   **Alternative:** N/A  **Exceptions:**  **Relationships:**  Generalized from Manage Event List use case.  **Business Rules:** N/A | | | |

Table 13. <HR Master> View yearly events

* + - 1. <HR Master> View events in monthly and daily

**Use Case Diagram**

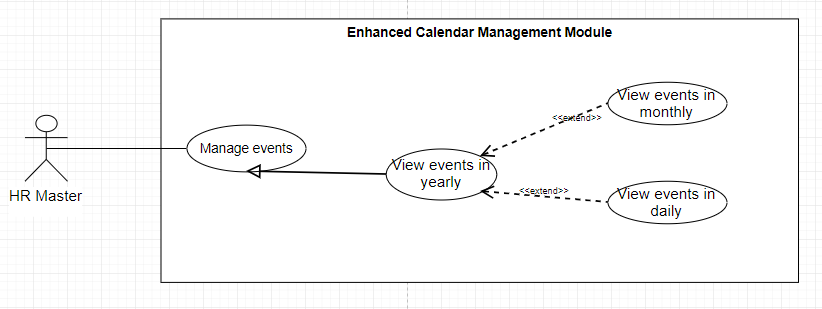


Figure 11. <HR Master> View events in monthly and dailys

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ECMM\_08** | | | |
| **Use Case No.** | 08 | **Use Case Version** | 1.1 |
| **Use Case Name** | View events in monthly and daily | | |
| **Author** | HuyNH | | |
| **Date** | 20/11/2018 | **Priority** | Normal |
| **Actor:**  HR Master  **Summary:**  This use case allows HR Master to view the summary of all events and leave requests in monthly view and daily view.  **Goal:**  View all events and leave requests in Monthly view, choose a day in month to view all events and leave request in daily view.  **Triggers:** Click on Monthly to view all the events and leave requests summary count.  **Preconditions:**  The events and leave requests record already exists in the system.  **Post Conditions:**  **Success:**  **Fail:**  Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Press on View Monthly on the menu slidebar. | View all the events and leave requests total count into Calendar UI in that month. The event and leave request is shown in 4 position:  - 1st is events.  - 2nd is PENDING leave requests. - 3rd is ON HOLD leave requests. - 4th is APPROVED leave requests. | | 2 | HR Master choose a day to view all the events and leave request on that day. | View all the events and leave requests with each leave status in daily view. |   **Alternative:** N/A  **Exceptions:** N/A  **Relationships:**  Generalized from Manage Event List use case and Extended from View Yearly use case.   * **Business Rules:** N/A | | | |

Table 14. <HR Master> View events and leave request in monthly

* + - 1. <HR Master> Create leave request

**Use Case Diagram**

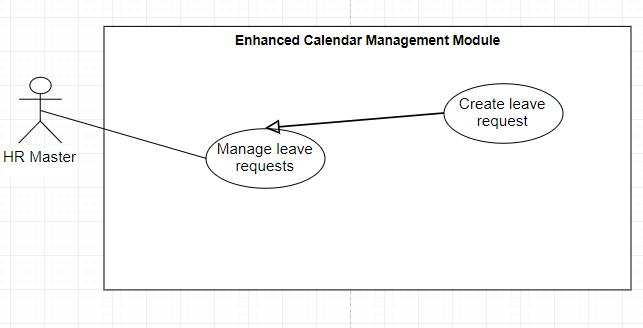


Figure 12. <HR Master> Create leave request

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ECMM\_09** | | | |
| **Use Case No.** | **09** | **Use Case Version** | 1.1 |
| **Use Case Name** | Create leave requests | | |
| **Author** | HuyNH | | |
| **Date** | 20/11/2018 | **Priority** | Normal |
| **Actor:**  HR Master  **Summary:**  This use case allows HR Master to create new leave request.  **Goal:**  HR Master can create new leave request and default status is PENDING.  **Triggers:**  Click on “New Apply Leave” in the top right corner  **Preconditions:**  **Post Conditions:**  **Success:** Leave requests is created and stored into database in PENDING status.  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on “New Apply Leave” in the top right corner. | Popup create leave request form will show. The popup form includes:  - Choose employee ID and employee name. - Choose start day and end date (the calculation for days of leave will be shown when choosing) - Choose type of leave. | | 2 | Fill in required information and click on “Submit” button. | [Exception 1]  [Exception 2]  Create new leave request with inputted information.  Redirect to current page and update count leave request. |   **Alternative:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Required fields are missing or fields are invalid. | Show instruction message below those input fields. | | 2 | Database occurs exception. | Show error message and ask user to try again later. |   **Relationships:**  Generalized from Manage Leave List use case.  **Business Rules:**  General validation:   * Choose employee ID and name to create leave. * Choose from date and to date (also calculate on total days of leave) * Choose type of leave requests – there are types of leave: MEDICAL, ANNUAL,COMPASSIONATE, MATERNITY, PROLONG ILLNESS,PILGRIMAGE . | | | |

Table 15. <HR Master> Create leave request

* + - 1. < HR Master> Update leave status

**Use Case Diagram**



Figure 13. <HR Master> Update leave status

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ECMM\_10** | | | |
| **Use Case No.** | 10 | **Use Case Version** | 1.1 |
| **Use Case Name** | Update leave status | | |
| **Author** | HuyNH | | |
| **Date** | 20/11/2018 | **Priority** | Normal |
| **Actor:**  Admin  **Summary:**  This use case allows HR Master to update leave status.  **Goal:**  Leave status is updated and change the color code of status in Calendar UI.  **Triggers:**  Click on "View Leave Request" tab and choose leave request that HR Master wants to update.  **Preconditions:**  The chosen leave request already exists in the system.  **Post Conditions:**  **Success:** Status of leave request is updated and change the color code in Calendar UI.  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Apply Leave in the menu slidebar. | Redirect to the Manage Leave Request page with lists of the leave requests. | | 2 | Choose leave request that HR Master wants to apply and choose Combo Box Status | Combo Box show 5 types of status: PENDING – ON HOLD – DECLINE – CANCEL – APPROVED. | | 3 | When HR Master choose 1 of 5 status | The popup confirm dialog appear. | | 4 | HR Master confirm to update the status | Return popup message success and redirect to current page. |   **Alternative:** N/A  **Exceptions:**  **Relationships:**  Generalized from Manage leave request List use case.  **Business Rules:**  There are 5 types of updating status: PENDING, ON HOLD, CANCEL, DECLINE and APPROVED. Each has some different status and meanings.   * PENDING: The default status when HR Master or Employee creating leave request. * ON HOLD: The HR Master has not decided yet. * CANCEL: The leave request will be deleted from database and Calendar UI. * DECLINE: The leave request will be deleted from Calendar UI but still keep in database. * APPROVED: The leave request is approved and change status into Calendar UI. | | | |

Table 16. <HR Master> Update leave status

* + - 1. < HR Master> Remove leave request

**Use Case Diagram**

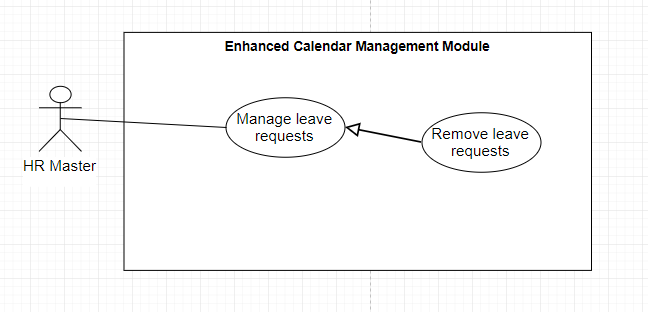


Figure 14. <HR Master> Remove leave request

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ECMM\_11** | | | |
| **Use Case No.** | **11** | **Use Case Version** | 1.0 |
| **Use Case Name** | Remove leave request | | |
| **Author** | HuyNH | | |
| **Date** | 20/11/2018 | **Priority** | Normal |
| **Actor:**  HR Master  **Summary:**  This use case allows HR Master to remove leave request.  **Goal:**  Leave requests is removed in that day in Calendar UI.  **Triggers:**  Click on “Delete” (recycle bin icon) button.  **Preconditions:**  The chosen leave request already exists in the system.  **Post Conditions:**  **Success:** Leave request of that day will be removed out of Calendar UI.  **Fail:** Show error message.  **Main Success Scenario: If HR Master remove the event in day view UI.**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Month view | The Calendar Month view UI appear with all the leave requests asking on that day of month. | | 2 | Choose the day that occur the event | The Day view UI appear to view all leave requests in that day with Delete(Recycle bin icon) button | | 3 | Choose the leave request that HR Master wants to remove | The popup show confirm delete message appear | | 4 | When HR Master confirm to delete the leave request. | [Exception 1]  Delete leave request successfully.  Popup message “Deleted successfully” and return to View Daily Calendar UI. |   **Alternative:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Database occurs exception. | Show error message and ask user to try again later. |   **Relationships:**  Generalized from Manage Leave request List use case. | | | |

Table 17. <HR Master> Remove leave request

* + - 1. <HR Master> View leave requests in yearly

**Use Case Diagram**

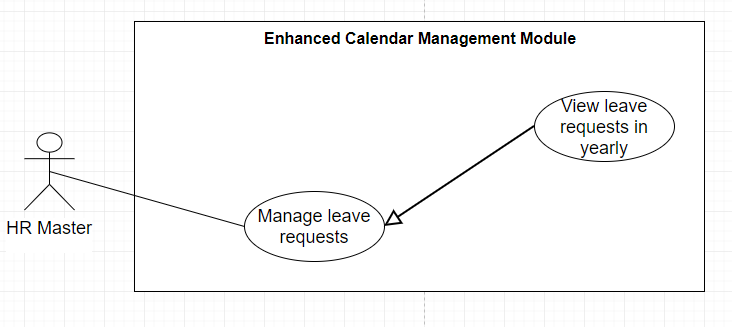
****

Figure 15. <HR Master> View leave requests in yearly

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ECMM\_12** | | | |
| **Use Case No.** | 12 | **Use Case Version** | 1.2 |
| **Use Case Name** | View leave requests in yearly | | |
| **Author** | HuyNH | | |
| **Date** | 20/11/2018 | **Priority** | Normal |
| **Actor:**  HR Master  **Summary:**  This use case allows HR Master to view the summary of all leave requests in yearly view.  **Goal:**  View all leave requests in Yearly view.  **Triggers:** Press on View Yearly in the Menu slidebar.  **Preconditions:**  The leave requests records already exists in the system.  **Post Conditions:**  **Success:** View leave requests (in orange color) in Calendar UI. (the red dot represents for more than 1 leave request on that day of month)  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Press on View Yearly in the menu slidebar. | View all the leave requests into Calendar UI. HR Master can hover to that day to show the leave requests and status of the leave request. |   **Alternative:** N/A  **Exceptions:**  **Relationships:**  Generalized from Manage Event List use case.  **Business Rules:** N/A | | | |

Table 18. <HR Master> View leave requests in yearly

* + - 1. <HR Master> View leave requests in monthly and daily.

**Use Case Diagram**



Figure 16. <HR Master> View leave requests in monthly and daily

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ECMM\_13** | | | |
| **Use Case No.** | 13 | **Use Case Version** | 1.2 |
| **Use Case Name** | View leave requests in monthly and daily | | |
| **Author** | HuyNH | | |
| **Date** | 21/11/2018 | **Priority** | Normal |
| **Actor:**  HR Master  **Summary:**  This use case allows HR Master to view the summary of all leave requests in monthly view and daily view.  **Goal:**  View all leave requests in Monthly view, choose a day in month to view all leave request in daily view.  **Triggers:** Click on Monthly to view all leave requests summary count.  **Preconditions:**  The leave requests record already exists in the system.  **Post Conditions:**  **Success:**  **Fail:**  Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Press on View Monthly on the menu slidebar. | View all the leave requests total count into Calendar UI in that month. The leave request is shown after the events title is shown:  - 1st is events.  - 2nd is PENDING leave requests. - 3rd is ON HOLD leave requests. - 4th is APPROVED leave requests. | | 2 | HR Master choose a day to view all the leave request on that day. | View all the leave requests with each leave status in daily view. |   **Alternative:** N/A  **Exceptions:** N/A  **Relationships:**  Generalized from Manage Event List use case and Extended from View Yearly use case.   * **Business Rules:** N/A | | | |

Table 19. <HR Master> View leave requests in monthly and yearly

* + - 1. < HR Master> Search leave request using filter

**Use Case Diagram**



Figure 17. <HR Master> Search leave request using filter

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ ECMM\_14** | | | |
| **Use Case No.** | **14** | **Use Case Version** | 1.1 |
| **Use Case Name** | Search leave request using filter | | |
| **Author** | HuyNH | | |
| **Date** | 20/11/2018 | **Priority** | Normal |
| **Actor:**  HR Master  **Summary:**  This use case allows HR Master to filter all the leave request of specific employee in monthly view.  **Goal:**  Levae request of that employee is filtered and view in Calendar Monthly UI.  **Triggers:**  Choose the monthly view and choose the employee the HR Master wants to filter.  **Preconditions:**  The chosen leave request of that employee exists in the system.  **Post Conditions:**  **Success:** Show all the record of the chosen employee in Calendar month UI.  **Fail:** Database error.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click on Monthly in the Menu slidebar. | Navigate to Monthly View Calendar UI. | | 2 | Choose the employee in the filter box. | [Exception 1]  Show all the record of the chosen employee and view into Calendar Monthly UI. |   **Alternative:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Database occurs exception. | Show error message and ask user to try again later. |   **Relationships:** Generalized from Manage Leave request List use case.  **Business Rules:** N/A | | | |

Table 20. <HR Master> Search leave request using filter.

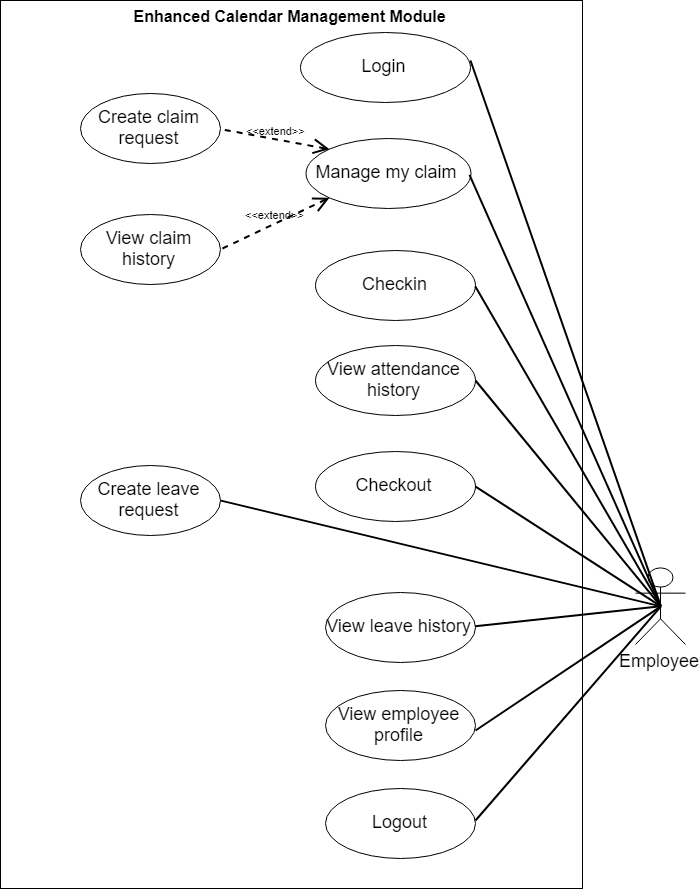
* + - 1. Employee Overview Use Cases

Figure 18. Employee overview use cases

* + - 1. <Employee> Login

**Use Case Diagram**

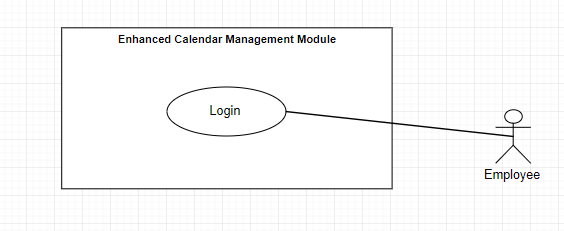


Figure 19. <Employee> Login

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_15** | | | |
| **Use Case No.** | 15 | **Use Case Version** | 1.0 |
| **Use Case Name** | Login | | |
| **Author** | HungTD | | |
| **Date** | 21/11/2018 | **Priority** | Normal |
| |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Actor:** Employee  **Summary:** This use case allows employee to login in Company Selection.  **Goal:** Employee can login in the system.  **Triggers:** Employee sends the login command  **Preconditions:**  **Post Conditions:**  **Success:** Navigate to choose company page.  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Employe goes to Login view | Check token in Storage [Alternative 1]  System requires indentity information form Employee:   * Email: free text input, required. * Password: free password input, required. | | 2 | Employee inputs information. |  | | 3 | Click button “Sign-in” | [Exception 1]  [Exception 2]  Navigate to Choose company page. |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Token already exist and not expires. | Navigate to Profile View |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Missing required fields. | Show instruction message bellow missing fields. | | 2 | Database occurs exception. | Navigate to Failed page. |   **Relationships:**  N/A  **Business Rules:** | | | | |

Table 21. <Employee> Login

* + - 1. <Employee> Log out

**Use Case Diagram**

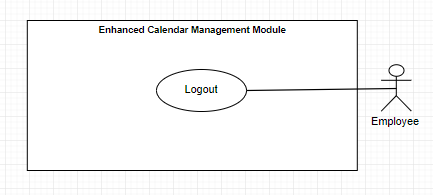
****

Figure 20. <Employee> My Profile

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_16** | | | |
| **Use Case No.** | **16** | **Use Case Version** | 2.0 |
| **Use Case Name** | Logout | | |
| **Author** | HUNGTD | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:** Employee  **Summary:** This use case allows employee logout system.  **Goal:** Employee can logout system.  **Triggers:** Click button “Logout” in the left menu.  **Preconditions:** Login or token in storage not expires.  **Post Conditions:**  **Success:** Show Login page.  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click button “Logout” in the left Menu  [Aternative 1] | Navigate to Login page |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | Token expired | Navigate to Login page |   **Exceptions: N/A**  **Relationships:**  **Business Rules:** | | | |

Table 22. <Employe> Logout

* + - 1. <Employee> View Employee Profile

**Use Case Diagram**

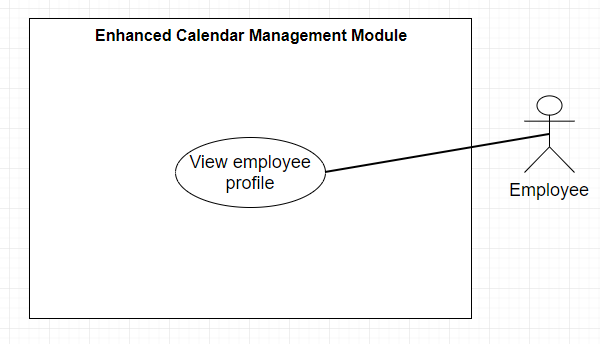


Figure 21. <Employee> View employee profile

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_16** | | | |
| **Use Case No.** | **16** | **Use Case Version** | 2.0 |
| **Use Case Name** | My Profile | | |
| **Author** | HUNGTD | | |
| **Date** | 19/11/2018 | **Priority** | Normal |
| **Actor:** Employee  **Summary:** This use case allows employee view their’s profile.  **Goal:** Employee can view their’s profile.  **Triggers:** Click button “My Profile / QR” in the left menu.  **Preconditions:** Login or token in storage not expires.  **Post Conditions:**  **Success:** Show Profile  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click button “My Profile / QR” in the left Menu | System show employee’s information:   * Employee’s avata * Employee Name * Employee ID * Employee Department * Employee position * Hr Master * Supervisor   [Exception 1] |   **Alternative:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Token expired | Navigate to Login page |   **Relationships:**  **Business Rules:** | | | |

Table 23. <Employee> View employee profile.

* + - 1. < Employee > Mobile Checkin attendance

**Use Case Diagram**

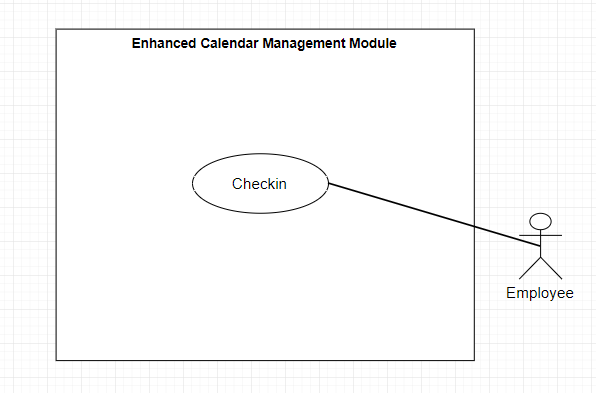


Figure 22. <Employee> Mobile checkin attendance

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_17** | | | |
| **Use Case No.** | 17 | **Use Case Version** | 1.0 |
| **Use Case Name** | Mobile Check-in | | |
| **Author** | HUNGTD | | |
| **Date** | 01/11/2018 | **Priority** | High |
| **Actor:** Employee  **Summary:** This use case allows employee check-in online.  **Goal:** Employee can check-in online.  **Triggers:** Click button “Check-in” in the left menu.  **Preconditions:** Login or Token not expires.  **Post Conditions:**  **Success:** Show message “Successful sign-in on”  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click button “Mobile Check-in” in the left Menu | [Alternative 1]  System show Check-in view and require some information:   * Reason for online check-in: free text-area input. | | 2 | Employee inputs information. |  | | 3 | Click button “mobile check-in” | System show confirm page:   * Time Check-in * Button “Confirm check-in” * Button “Cancel” * Location (lat,long) in the bottom   [Exception 1] | | 4 | Click button “confirm check-out”  [Alternative 2] | System show success page :   * Time check-in: * Reason online check-in * Location(lat,long) |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | If employee has checked-in | System navigate and show check-out page. | | 2 | If employee click button “Cancel” | System reload check-in page. |   **Exceptions: N/A**  **Relationships:**  **Business Rules:**  Validation: | | | |

Table 24. <Employee> Mobile checkin attendance

* + - 1. < Employee > Mobile Checkout attendance

**Use Case Diagram**

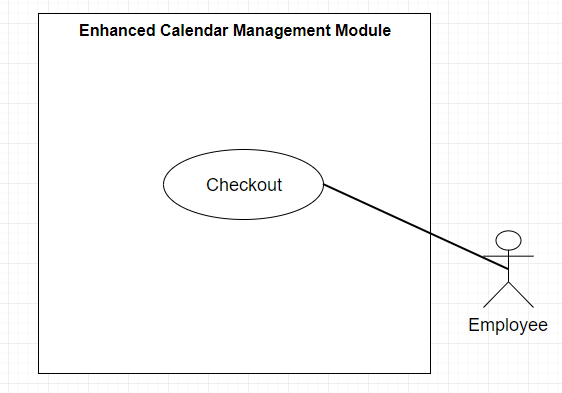


Figure 23. <Employee> Mobile checkout attendance

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_18** | | | |
| **Use Case No.** | 18 | **Use Case Version** | 2.0 |
| **Use Case Name** | Mobile Check-out | | |
| **Author** | HUNGTD | | |
| **Date** | 07/11/2018 | **Priority** | High |
| **Actor:** Employee  **Summary:** This use case allows employee check-out  **Goal:** Employee can check-out  **Triggers:** Click button “Check-in” in the left menu.  **Preconditions:** Already Check-in.  **Post Conditions:**  **Success:** Show message “Successful check-out on:  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click button “Check-in” in the left Menu | [Alternative 1]  [Alternative 2]  System show Check-out view and require some information:   * Reason for online check-out: free text-area input. | | 2 | Employee inputs information. |  | | 3 | Click button “mobile check-out” | System show confirm page:   * Time Check-out * Button “Confirm check-out” * Button “Cancel” * Location (lat,long) in the bottom   [Exception 1] | | 4 | Click button “confirm check-out”  [Alternative 3] | System show success page :   * Time check-out: * Reason online check-out * Location(lat,long) |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor action | System Response | | 1 | If employee has not check-in. | System navigate and show check-in page. | | 2 | If employee has checked-out | System show success page:   * Time check-out: * Reason online check-out * Location(lat,long) | | 3 | If Employee click button “Cancel” | System reload page check-in |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Database occurs exception. | Show error message and ask employee to try again. |   **Relationships:**  **Business Rules:** | | | |

Table 25. <Employee> Mobile checkout attendance.

* + - 1. < Employee > View attendance report

**Use Case Diagram**

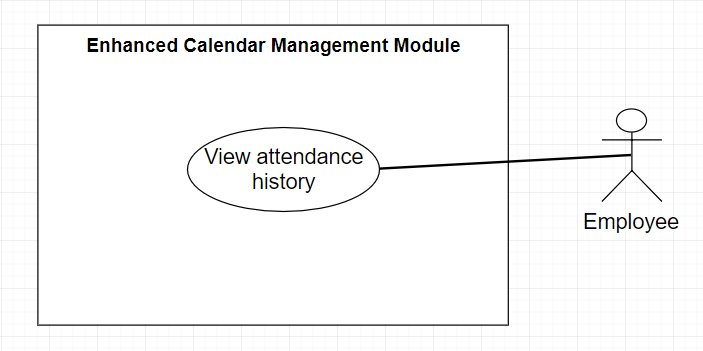


Figure 24. <Employee> View attendance report.

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_19** | | | |
| **Use Case No.** | 19 | **Use Case Version** | 2.0 |
| **Use Case Name** | Attendance Report | | |
| **Author** | HUNGTD | | |
| **Date** | 07/11/2018 | **Priority** | Normal |
| **Actor:** Employee  **Summary:** This use case allows employee can see check-in status statistics in month.  **Goal:** Employee can see check-in status statistics  **Triggers:** Click button “Attendance report” in the Attendance page.  **Preconditions:** Login or token in storage not expires.  **Post Conditions:**  **Success:** Show attendance report view.  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click button “Attendance report” in the Attendance page. | System show:   * The blue circle represents the percent of presence for working * Current month. * Display working day in month with 5 type: * On time. * On Leave * Late * Very Late * Absence |   **Alternative:** N/A  **Exceptions:** N/A  **Relationships:**  **Business Rules:**  Validation: | | | |

Table 26. <Employee> View attendance history.

* + - 1. < Employee > Create leave requests

**Use Case Diagram**

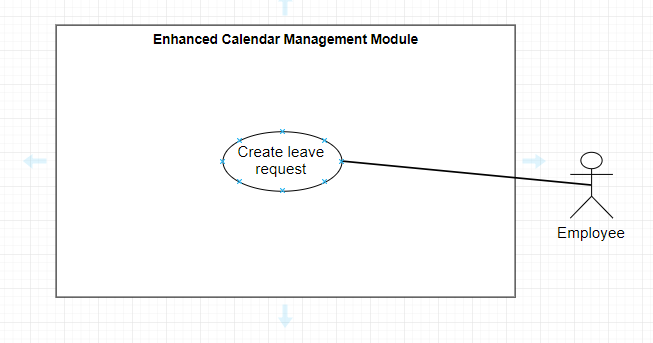


Figure 25. <Employee> Create leave request.

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_20** | | | |
| **Use Case No.** | **20** | **Use Case Version** | 2.0 |
| **Use Case Name** | My Leave Apply | | |
| **Author** | HUNGTD | | |
| **Date** | 06/11/2018 | **Priority** | Normal |
| **Actor:** Employee  **Summary:**  This use case allows employees create new apply leave report.  **Goal:** Employee can create new apply leave report.  **Triggers:** Click button “My Leave” in the left menu. Then the tab “Apply leave” display like default.  **Preconditions:** Login or Token in storage not expires.  **Post Conditions:**  **Success:** Create new apply leave report.  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click button “My Leave” in left menu. Then the tab “Apply Leave” display like default. | System require information:   * Type of leave: combobox of 5 type. * Medical * Annual * Compassionate * Materinity * Pilgrimage * Prolong lllness * Date: * Remark: free text-area input * Attachment (optional): * Submit now: button submit type. | | 2 | Employee inputs information. | [Exception 1] | | 3 | Click button “Submit now” | Message success appear.  [Exception 2] |   **Alternative:** N/A  **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Validation field | Show error message. | | 2 | Database occurs exception. | Show error message and ask employee to try again. |   **Relationships:**  **Business Rules:**  Validation: | | | |

Table 27. <Employee> Create leave request.

* + - 1. < Employee > View leave request history

**Use Case Diagram**

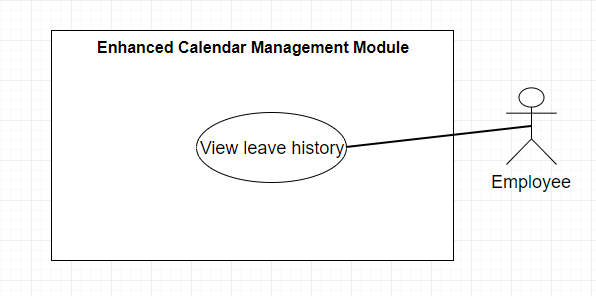


Figure 26. <Employee> View leave request history

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_21** | | | |
| **Use Case No.** | **21** | **Use Case Version** | 2.0 |
| **Use Case Name** | View leave request history | | |
| **Author** | HUNGTD | | |
| **Date** | 06/11/2018 | **Priority** | Normal |
| **Actor:** Employee  **Summary:** Show all of employee’s leave report  **Goal:** Employee see their’s leave report  **Triggers:** Click tab “Status” in Page My Leaves  **Preconditions:** Login or Token not expires.  **Post Conditions:**  **Success:**  Show all of list employee’s leave report  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click tab “Status” in My leaves Page | Show all of list employee’s leave report.  [Alternative 1] |   **Alternative:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | If employee no had any leave report | There are no leave report. |   **Exceptions:** N/A  **Relationships:**  **Business Rules:** | | | |

Table 28. <Employee> View leave request history.

* + - 1. < Employee > Manage my claim

**Use Case Diagram**

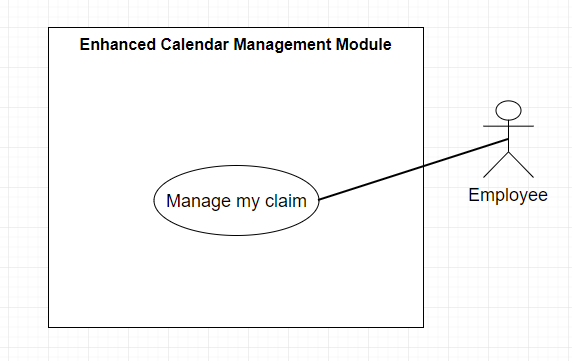


Figure 27. <Employee> Manage my claim.

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_22** | | | |
| **Use Case No.** | **22** | **Use Case Version** | 2.0 |
| **Use Case Name** | Overview My Claims | | |
| **Author** | HUNGTD | | |
| **Date** | 06/11/2018 | **Priority** | High |
| **Actor:** Employee  **Summary:** Employee can create new claims report and view all of employee’s claims report.  **Goal:** Employee create new and view claims report.  **Triggers:** Click button “My Claims” in the left menu.  **Preconditions:** Login or token in storage not expires.  **Post Conditions:**  **Success:** Show My claims view.  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click button “My Claims” in the left Menu | Check token in storage [Alternative 1]  Show:   * Approver name: * Total success claims this year: * Apply claims tab: * Status claims tab: |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Check token expires | Navigate to Login Page |   **Exceptions: N/A**  **Relationships:**  **Business Rules:**  Validation: | | | |

Table 29. <Employee> Manage my claim.

* + - 1. < Employee > Create my claim

**Use Case Diagram**

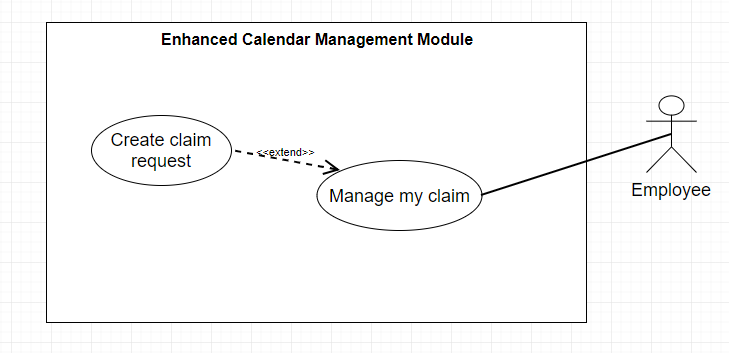


Figure 28. <Employee> Create claim request.

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_23** | | | |
| **Use Case No.** | **23** | **Use Case Version** | 2.0 |
| **Use Case Name** | Create my claim. | | |
| **Author** | HUNGTD | | |
| **Date** | 07/11/2018 | **Priority** | Normal |
| **Actor:** Employee  **Summary:** Employee create their’s claims report.  **Goal:** Employee create claims report  **Triggers:** Click button “My Claims” in the left menu  **Preconditions:** Login or Token in storage not expires.  **Post Conditions:**  **Success:** Show My Claims view.  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click button “My Claims” in the left Menu | Check token is expires [Alternative 1]  Show all of Attendence view like UC: 15 | | 2 | Default when access to My Claims page is “Apply” tab. | Show create Apply Claims form:   * Type of Claims: * Amout: free number input. * Remarks: free text-area input * Attachment (optional): * Submit now | | 3 | Employee input information |  |  | | 4 | Click button “Submit now” | Message “Success” appear.  New record added in Database and display in Claims tab.  [Exception 1]  [Excepion 2] |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Token Expires | Navigate to Login Page |   **Exceptions:**   |  |  |  | | --- | --- | --- | | No | Cause | System Response | | 1 | Validation Feild | Show error message. | | 2 | Database occurs exception. | Show error message and ask employee to try again. |   **Relationships:**  **Business Rules:**  Validation: | | | |

Table 30. <Employee> Create claim requests.

* + - 1. < Employee > View claim history

**Use Case Diagram**

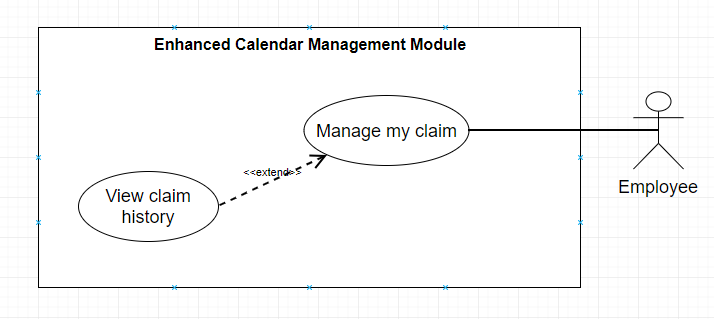


Figure 29. <Employee> View claim history

**Use Case Specification**

|  |  |  |  |
| --- | --- | --- | --- |
| **USE CASE – UC\_ECMM\_24** | | | |
| **Use Case No.** | **24** | **Use Case Version** | 2.0 |
| **Use Case Name** | My Claims Status | | |
| **Author** | HUNGTD | | |
| **Date** | 06/11/2018 | **Priority** | Normal |
| **Actor:** Employee  **Summary:** Show all of employee’s claims report  **Goal:** Employee see all of their’s claims report  **Triggers:** Click “My Claims” in left menu  **Preconditions:**  **Post Conditions:**  **Success:** Show My Claims view  **Fail:** Show error message.  **Main Success Scenario:**   |  |  |  | | --- | --- | --- | | Step | Actor Action | System Response | | 1 | Click button “My Claims” in the left Menu | Check token in Storage [Alternative 1]  Show all of My Claims view like UC: 15 | | 2 | Click tab “Status” | Not exsit any report [Alternative 2]  Show list of employee’s claims report with 3 properties:   * Applied on: * Type: * Status: * Image about claims (optional) |   **Alternative:**   |  |  |  | | --- | --- | --- | | No | Actor Action | System Response | | 1 | Token Expires | Navigate to Login Page | | 2 | Not exist any employee’s claims report | There are no claims report display in tab status. |   **Exceptions: N/A**  **Relationships:**  **Business Rules:**  Validation: | | | |

Table 31. <Employee> View claim history.

1. **Software System Attributes** 
   1. Usability
      1. Graphic User Interface

* All textboxes, labels, alerts and messages are written in English. GUI for web application is designed based on the requirements given by Track&Roll Company.
* Mobile application is based on existed design app on the requirements of T&R Company.
  + 1. Usability
* The web application is assumed to be the small module and combine to synchronized with the current HR Management System of Track&Roll company.
* The ESS Mobile application with friendly view mobile and simple use.
  1. Reliability
* Data should be backed up once a day.
  1. Availability
* System must be available 24/7.
* System can handle a large number of requests.
  1. **Security**
* User password is encrypted in database.
* All input data should be validated before saving to database.
  1. **Maintainability**
* The system is divided into separated modules.
* When a module is down, it is easy to take it down to fix without impacting other functions.
* Code is easy to maintain and upgrade.
  1. Portability
* Website runs on Chrome version 50 or above, or other common browsers.
  1. Performance
* All requests are handled in maximum time of 5 seconds.

1. Conceptual Diagram

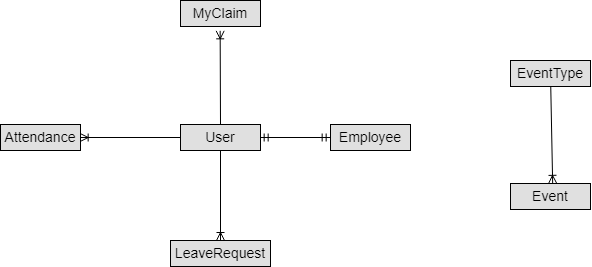


Figure 30. Conceptual diagram

# **D. Software Design Description**

**1. Design Overview**

This document describes the technical and user interface design of HiEIS system. It includes architectural design, detail design of common functions, business functions and design of database model. Document overview:

Section 2: Overall description of system architecture design.

Section 3: Component diagrams that describe the connection and integration of the system.

Section 4: Detail design description including class diagram, class explanation, and sequence diagram to application function details.

Section 5: Screen designs.

Section 6: Attributed ERD.

**2. System Architectural Design**

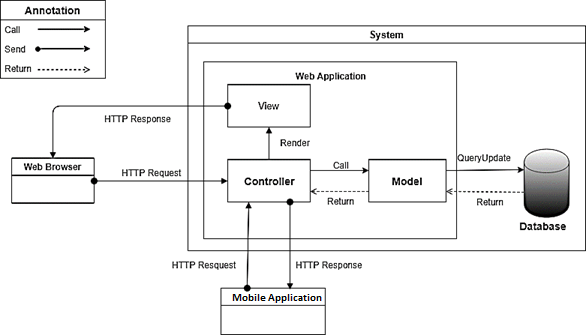
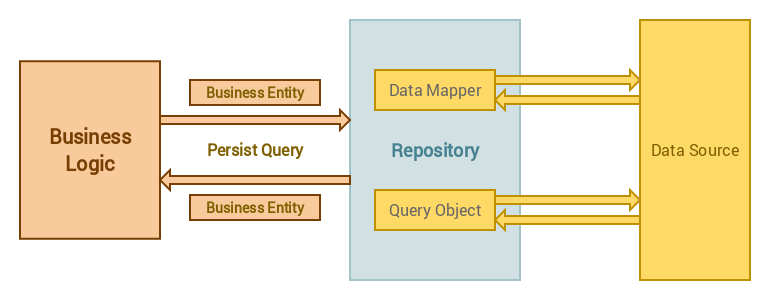
**

Figure 31. System architectural design

In Web Application, the system is developed under MVC architecture style. This project follows MVC architecture with following components:

* Controller handles user interaction. Typically, controller reads data from a request, calls appropriate business method, selects and renders view for user.
* View is a user interface. It is responsible for rendering the model data and generating HTML output that client's browser can interpret.
* Model is responsible for managing data of the application. It responds to requests from view and also to instructions from controller to update itself. Model objects store and retrieve model state in a database.  
  

But not only using the old MVC Model, the maintenance code will have lots of difficult because the LOC in Controller would be too many, defragged code is very hard so in the MVC Architecture, our team apply a design pattern that can reduce the code in Comtroller so that it is not affective to the current system, and also the maintenance will be easy, especially for new feature.

**3. Component Diagram**

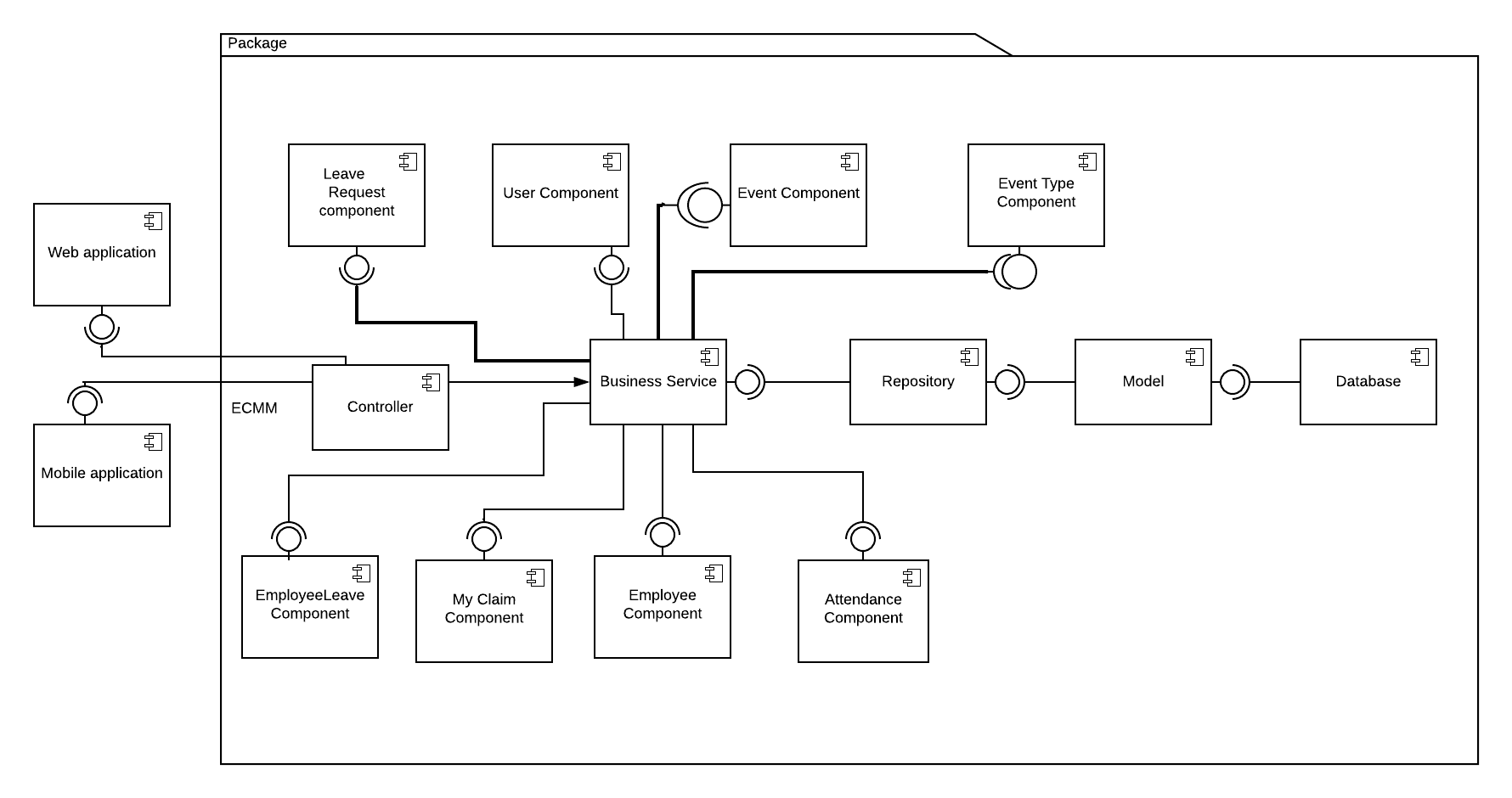


Figure 32. Component diagram

|  |  |
| --- | --- |
| COMPONENT DICTIONARY: DESCRIBES COMPONENTS | |
| HRMM | Corporate Human Resources Management Module |
| Web Application | Web application package contains operation of ECMM on web |
| Mobile Application | Mobile application package contains operation of ESS on mobile. |
| Controller | Handle requests and responses; accept input and convert it to commands for model and view. |
| Business Service | Handle system’s business operations |
| Data Model | Present the interaction between the system and database. |
| DB Storage | Store data of system. |
| Employee Component | Handle employee information in the system |
| Event Component | Handle event information in the system |
| User Component | Handle user account in the system |
| Event Type Component | Handle types of events information in the system |
| MyClaim Component | Handle claim of the employee information in the system |
| EmployeeLeave Component | Handle EmployeeLeave in the system |
| Attendance Component | Handle attendance information in the system |

Table 32. Component diagram dictionary

**4. Detail Description**

**4.1 Class Diagram**

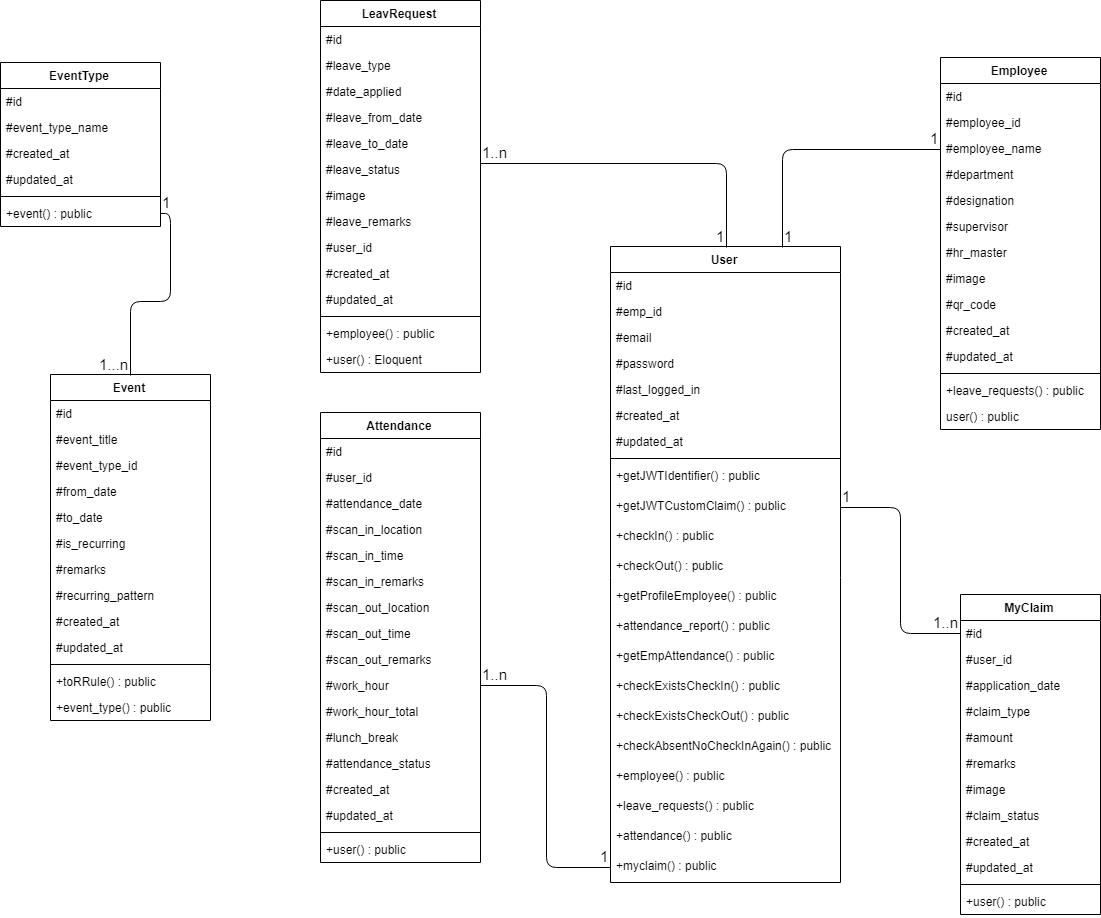
****

Figure 33. Class diagram

|  |  |  |
| --- | --- | --- |
| **CLASS DICTIONARY: DESCRIBE CLASS** | | |
| **Class Name** | **Mapping column with Conceptual diagram** | **Description** |
| **User** | User | Contain account information |
| **Employee** | Employee | Contain employee information of that user |
| **Attendance** | Attendance | Contain attendance record of the employee. |
| **LeaveRequest** | LeaveRequest | Contain leave request information |
| **Event** | Event | Contain event information |
| **EventType** | EventType | Contain types of events information |
| **MyClaim** | MyClaim | Contain my claim requests. |

Table 33. Class Diagram Dictionary

**4.2 Class Diagram Explanation**

1. **User**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Id | Integer | Protected | Mark the number of accounts. |
| emp\_id | String | Protected | Contains employee id |
| Email | String | Protected | Employee’s email. |
| Password | String | Protected | Employee’s password. |
| last\_logged\_in | Datetime | Protected | User’s phone number |
| Created\_at | Datetime | Protected | Timestamp |
| Updated\_at | Datetime | Protected | Timestamp |

Table 34. <Class Diagram Attributes> User

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| getJWTIdentifier | mixed | Public | Get the identifier that will be stored in the subject claim of the JWT |
| getJWTCustomClaims | array | Public | Return a key value array, containing any custom claims to be added to the JWT |
| checkIn | Array | Public | Return the array with checkIn information |
| CheckOut | Array | Public | Return the array with checkOut information |
| getProfileEmployee | Array | Public | Return the employee information |
| attendance\_report | Collection | Public | Return all the attendance record of that employeebased on attendance\_status |
| getEmpAttendance | Collection | Public | Return all record using the current datetime of system. |
| checkExistsCheckIn | Boolean | Public | Check if the employee has already checked in or not |
| checkExistsCheckOut | Boolean | Public | Check if the employee has already checked out or not |
| checkAbsentNoCheckInAgain | Boolean | Public | Return Boolean if user is absent, not allowed to check in. |
| employee | Eloquent | Public | One user is only have one employee profile. |
| leave\_request | Eloquent | Public | One user can have many leave requests |
| attendance | Eloquent | Public | One user can have many attendance records |
| myclaim | Eloquent | Public | One user can have many claim requests. |

Table 35. Class Diagram Methods> User

1. **Attendance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Id | Integer | Protected | Unique identifier of a role |
| User\_id | String | Protected | User id |
| Attendane\_date | Date | Protected | Current date for checking attendance |
| Scan\_in\_location | String | Protected | Place for check in |
| Scan\_in\_time | Time | Protected | Time for check in |
| Scan\_in\_remarks | String | Protected | Information about check in |
| Scan\_out\_location | String | Protected | Place for check out |
| Scan\_out\_time | Time | Protected | Time for check out |
| Scan\_out\_remarks | String | Protected | Information about check out |
| Work\_hour | Time | Protected | Total work hour include lunch break |
| Work\_hour\_total | Time | Protected | Total work hour deduct lunch break |
| Lunch\_break | Time | Protected | Default is 1 hour |
| Attendance\_status | Integer | Protected | The status of the check in |
| Created\_at | Datetime | Protected | Timestamp |
| Updated\_at | Datetime | Protected | Timestamp |

Table 36. <Class Diagram Attributes> Attendance

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| User() | Eloquent | Public | Eloquent with User |

Table 37. <Class Diagram Methods> Attendance

1. **Employee**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Id | Integer | Protected | Auto increment for number of records. |
| Employee\_id | String | Protected | Identifier of an employee id |
| Employee\_name | String | Protected | Identifier of an employee name |
| Department | String | Protected | Identifier of a department |
| Designation | String | Protected | Identifier of the current position in the department. |
| Supervisor | String | Protected | Identifier of the supervisor |
| HR\_Master | String | Protected | Identifier of the HR Master |
| Image | String | Protected | Identifier of the employee image profile |
| QR\_code | String | Protected | Identifier of the QR code of that employee |
| Created\_at | Datetime | Protected | Timestamp |
| Updated\_at | Datetime | Protected | Timestamp |

Table 38. < Class Diagram Attributes> Employee

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| leave\_requests() | Eloquent | Public | Employee own their many leave requests |
| user | Eloquent | Public | One employee belongs to one user account |

Table 39. <Class Diagram Methods> Employee

1. **EventType**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Id | Integer | Protected | Auto increase number of record |
| Event\_type\_name | String | Protected | Identifier of event type name |
| Created\_at | Datetime | Protected | Timestamp |
| Updated\_at | Datetime | Protected | Timestamp |

Table 40. <Class Diagram Attributes> EventType

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| Event | Eloquent | Public | One event type can have many events |

Table 41. <Class Diagram Methods> EventType

1. **Events**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Id | Integer | Protected | Auto increase number of records |
| Event\_title | String | Protected | Event’s title |
| Event\_type\_id | Integer | Protected | Event’s type id |
| From\_date | Datetime | Protected | Event‘s start date |
| To\_date | Datetime | Protected | Event’s end date |
| Remarks | int | Protected | Event’s alternative information |
| Is\_recurring | Boolean | Protected | Check if the event is recurring or not |
| Recurring\_pattern | String | Protected | Event recurring information |
| Created\_at | Datetime | Protected | Timestamp |
| Updated\_at | Datetime | Protected | Timestamp |

Table 42. <Class Diagram Attributes> Events

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| Event\_type | Eloquent | Public | One event belongs to one type of event. |
| toRRule | Array | Public | RRULE to get the data in json\_decode |

Table 43. <Class Diagram Methods> Events

1. **LeaveRequest**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Id | Integer | Protected | Auto increase number of records |
| Leave\_type | Integer | Protected | Type of leave requests |
| Date\_applied | Float | Protected | Count numbers of days for apply leave |
| Leave\_from\_date | Datetime | Protected | The start day for leave |
| Leave\_to\_date | Datetime | Protected | The end day for leave |
| Leave\_status | Integer | Protected | Leave status whether it is approved or still on hold. |
| Image | String | Protected | Optional |
| Leave\_remarks | String | Protected | Note for leave requests |
| User\_id | Integer | Protected | Leave requests belongs to 1 user |
| Created\_at | Datetime | Protected | Timestamp |
| Updated\_at | Datetime | Protected | Timestamp |

Table 44. <Class Diagram Attributes> LeaveRequest

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| Employee | Eloquent | Public | One employee can have many leave requests |
| User | Eloquent | Public | One leave request belongs to one user. |

Table 45. <Class Diagram Methods> LeaveRequest

1. **MyClaim**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute** | **Type** | **Visibility** | **Description** |
| Id | Integer | Protected | Auto increase number of records |
| User\_id | Int | Protected | Claim request of that user |
| Application\_date | Datetime | Protected | Date for applied the claim request |
| Claim\_type | Integer | Protected | Type of claim fr request |
| Amount | Float | Protected | The amount of budget/money. |
| Remarks | String | Protected | Information of the claim request |
| Image | String | Protected | Picture of the receipt |
| Claim\_status | Integer | Protected | The status whether the claim is approved or on hold |
| Created\_at | Datetime | Protected | Timestamp |
| Updated\_at | Datetime | Protected | Timestamp |

Table 46. <Class Diagram Attributes> My Claim

|  |  |  |  |
| --- | --- | --- | --- |
| **Method** | **Return Type** | **Visibility** | **Description** |
| User | Eloquent | Public | One Claim requests belongs to one user only. |

Table 47. <Class Diagram Methods> My Claim

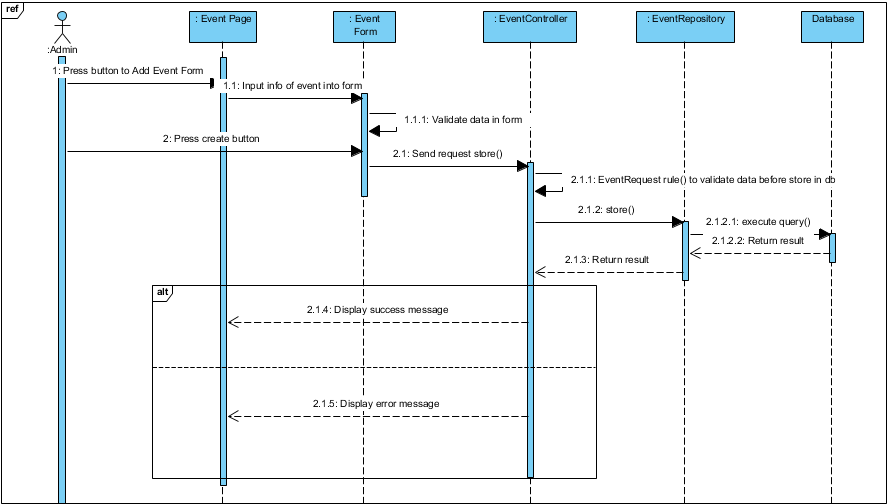
* 1. Interaction Diagram
     1. <HR Master> Create events 

Figure 34. <Sequence Diagram><HR Master> Create events

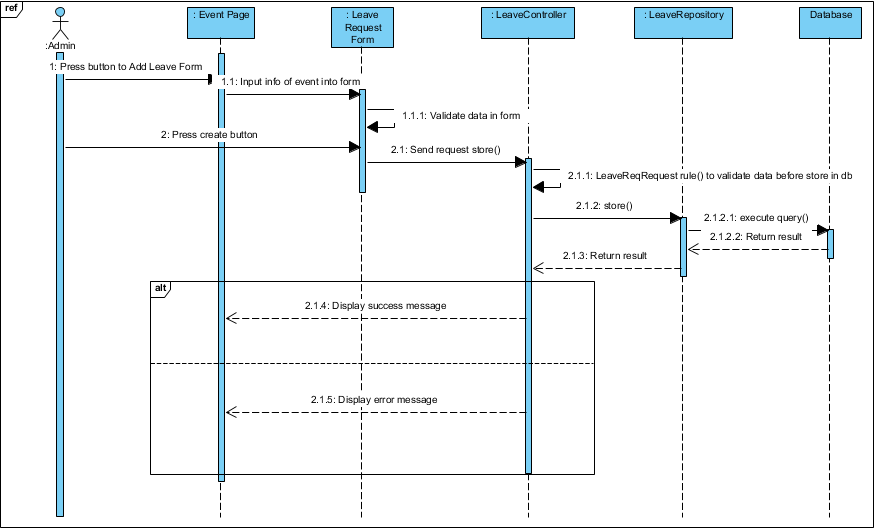
* + 1. <HR Master, Employee > Create leave requests

Figure 35. <Sequence Diagram><HR Master > Create leave requests

* + 1. <HR Master> View all events

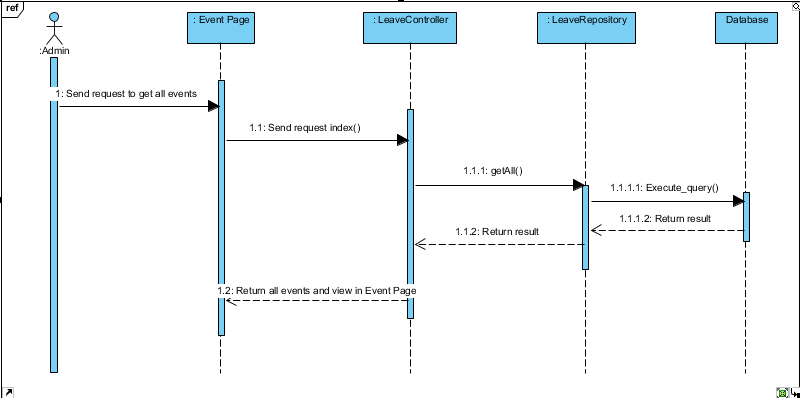


Figure 36. <Sequence Diagram><HR Master> View all events

* + 1. <HR Master> Update event information

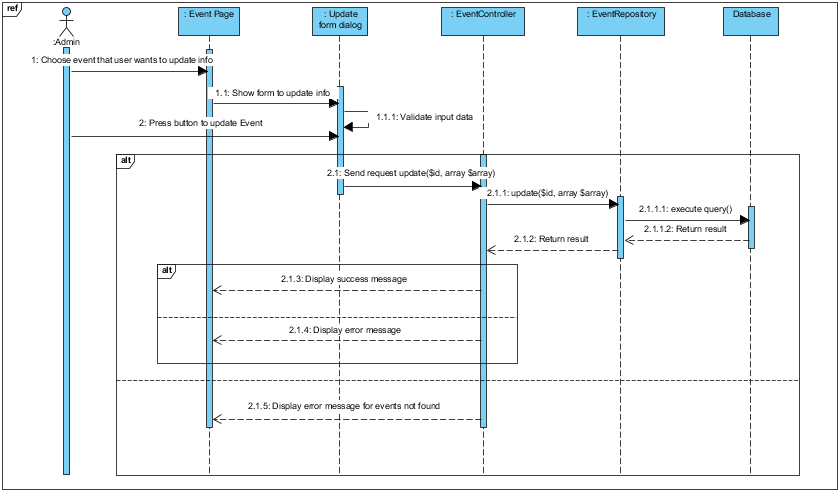


Figure 37. <Sequence Diagram><HR Master> Update event information

* + 1. <HR Master> Delete events

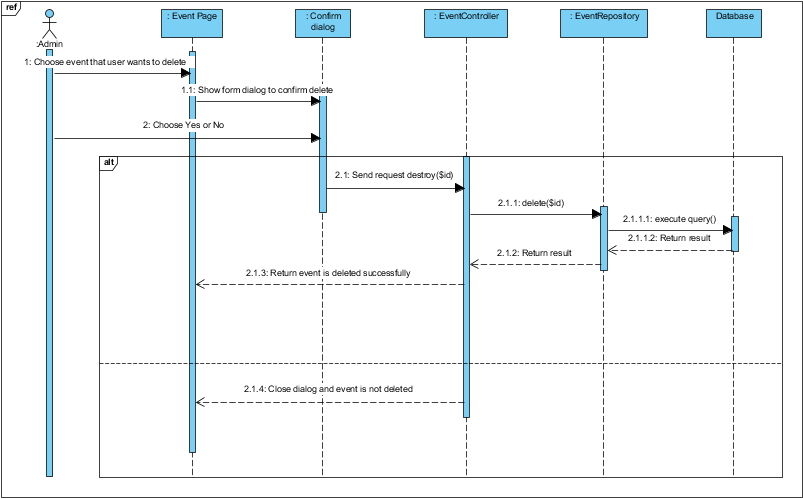


Figure 38. <Sequence Diagram><HR Master> Delete events

* + 1. <HR Master> Update status of leave request

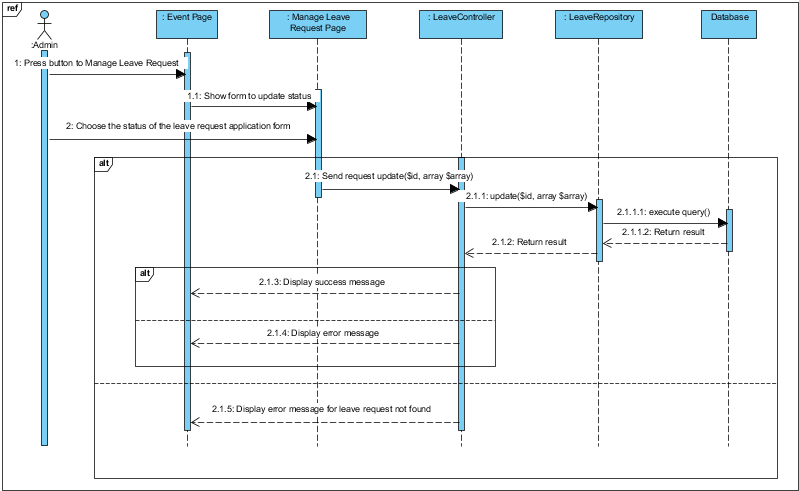


Figure 39. <Sequence Diagram><HR Master> Update status of leave request

* + 1. <HR Master> Delete leave request

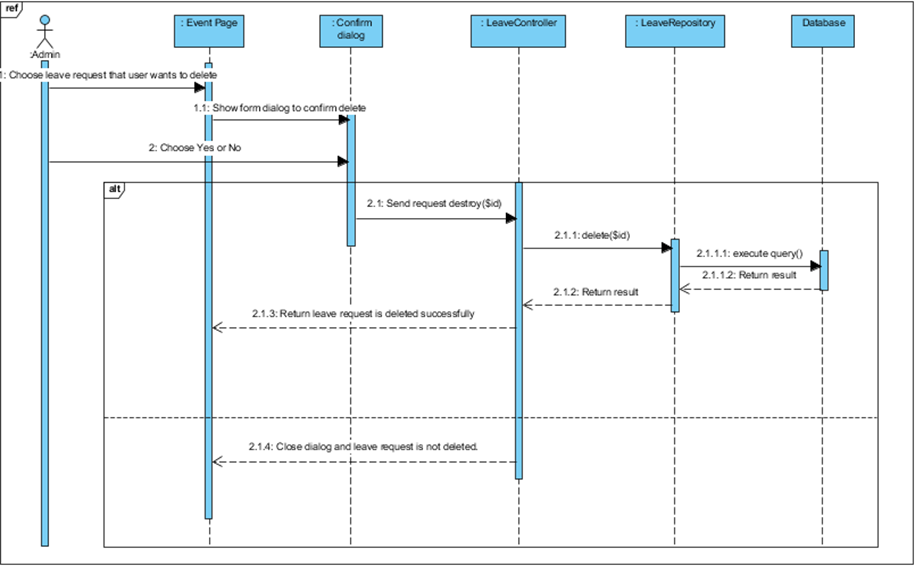


Figure 40. <Sequence Diagram><HR Master> Delete leave request

* + 1. <Employee > View profile

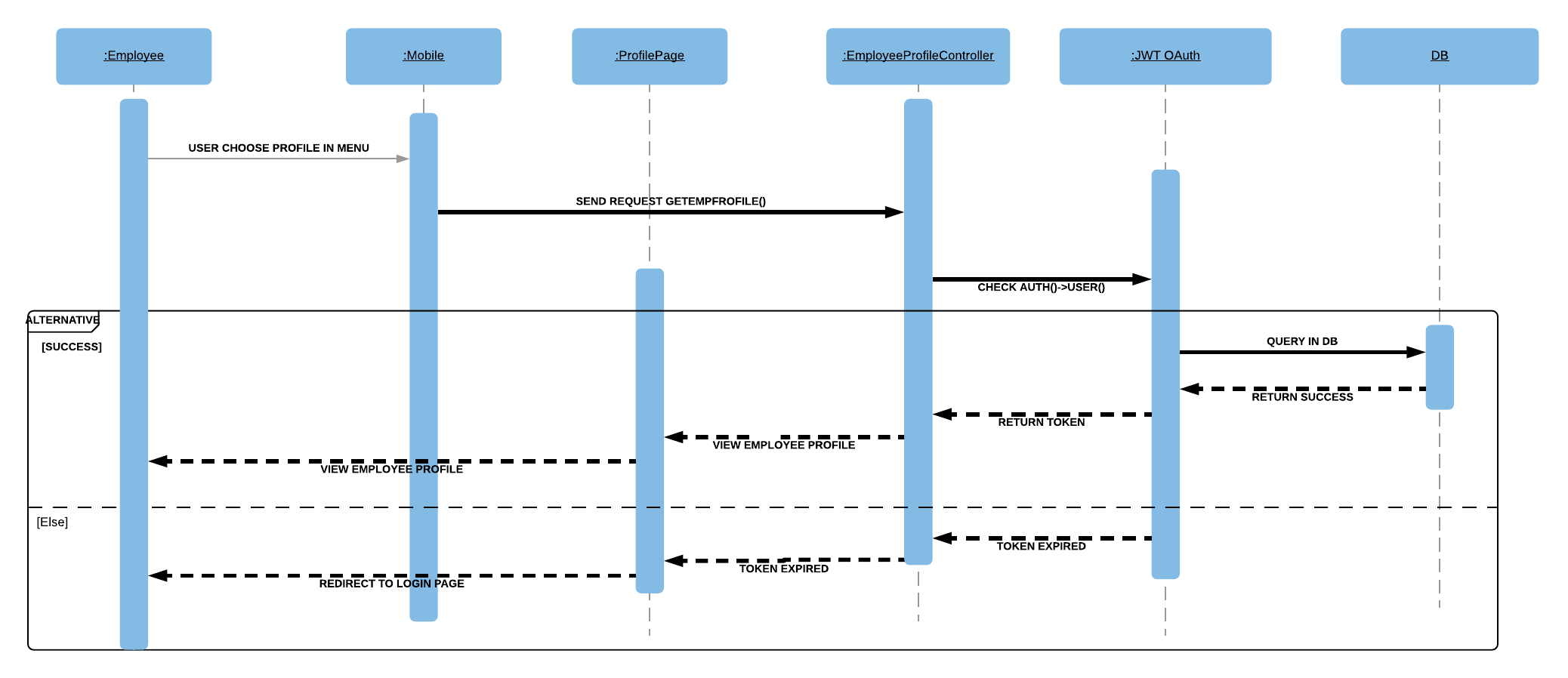


Figure 41. <Sequence Diagram><Employee> View profile employee.

* + 1. <Employee > Check in

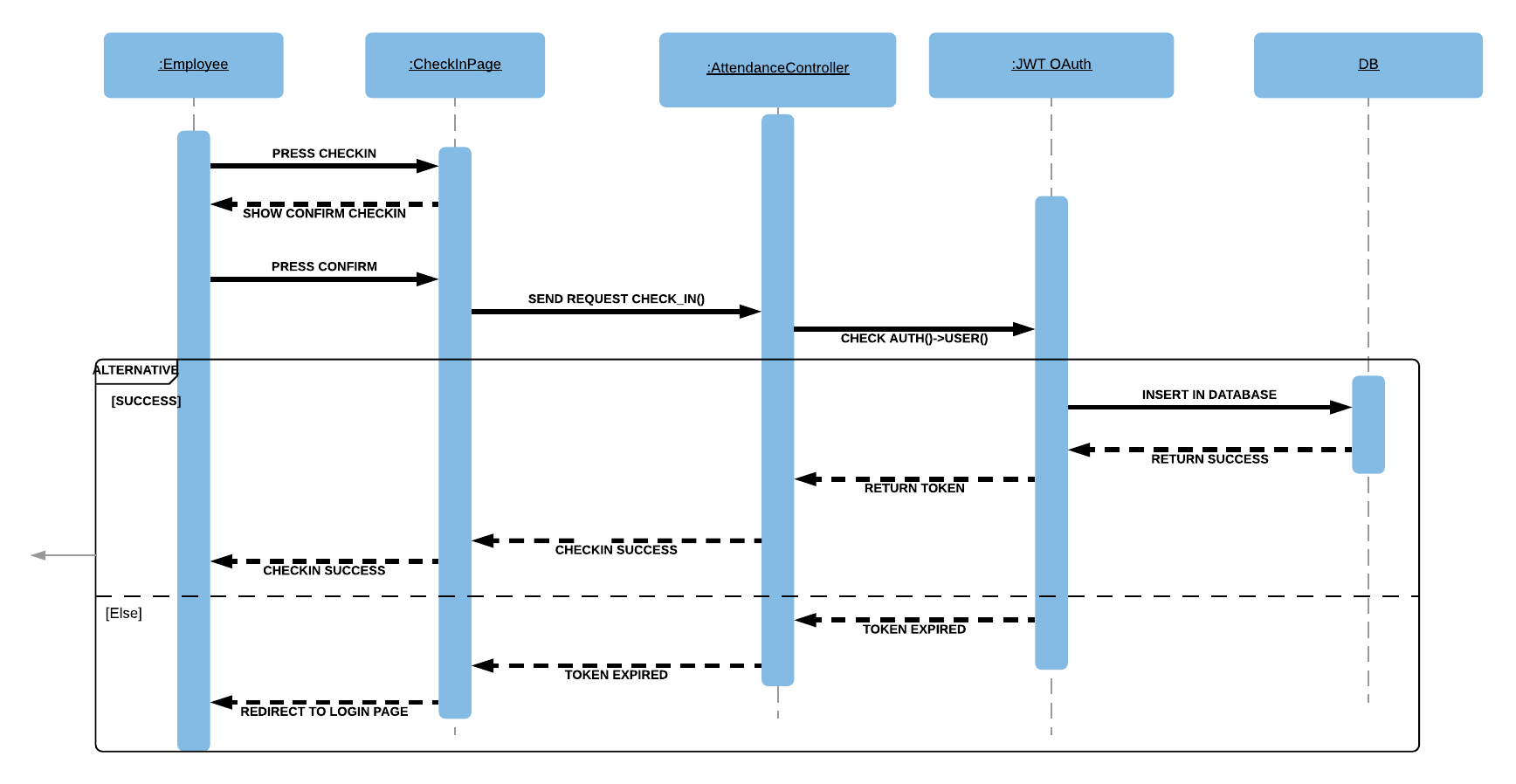


Figure 42. <Sequence Diagram><Employee> Checkin.

* + 1. <Employee > Check out

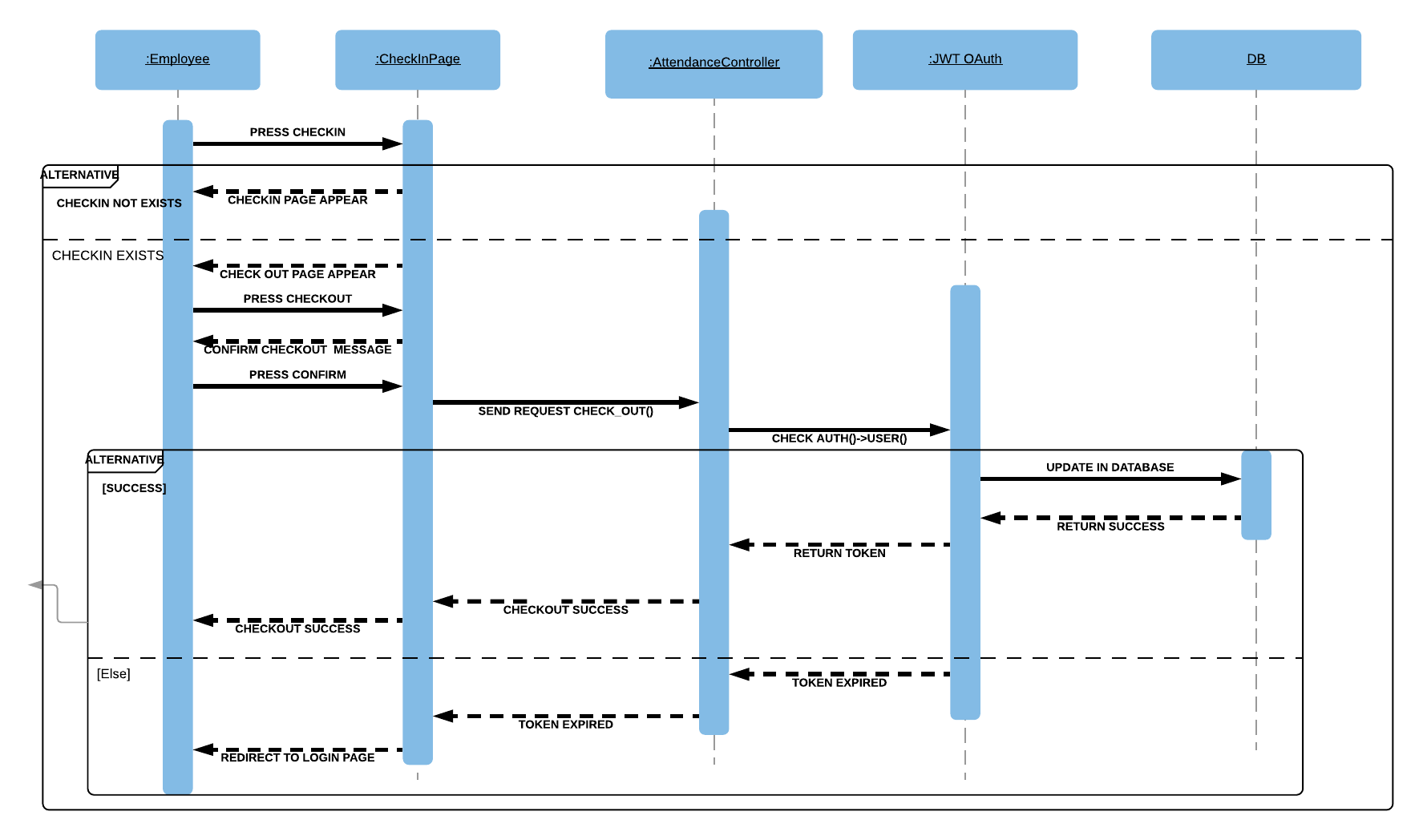


Figure 43. <Sequence Diagram><Employee> Checkout.

* + 1. <Employee > Create leave request

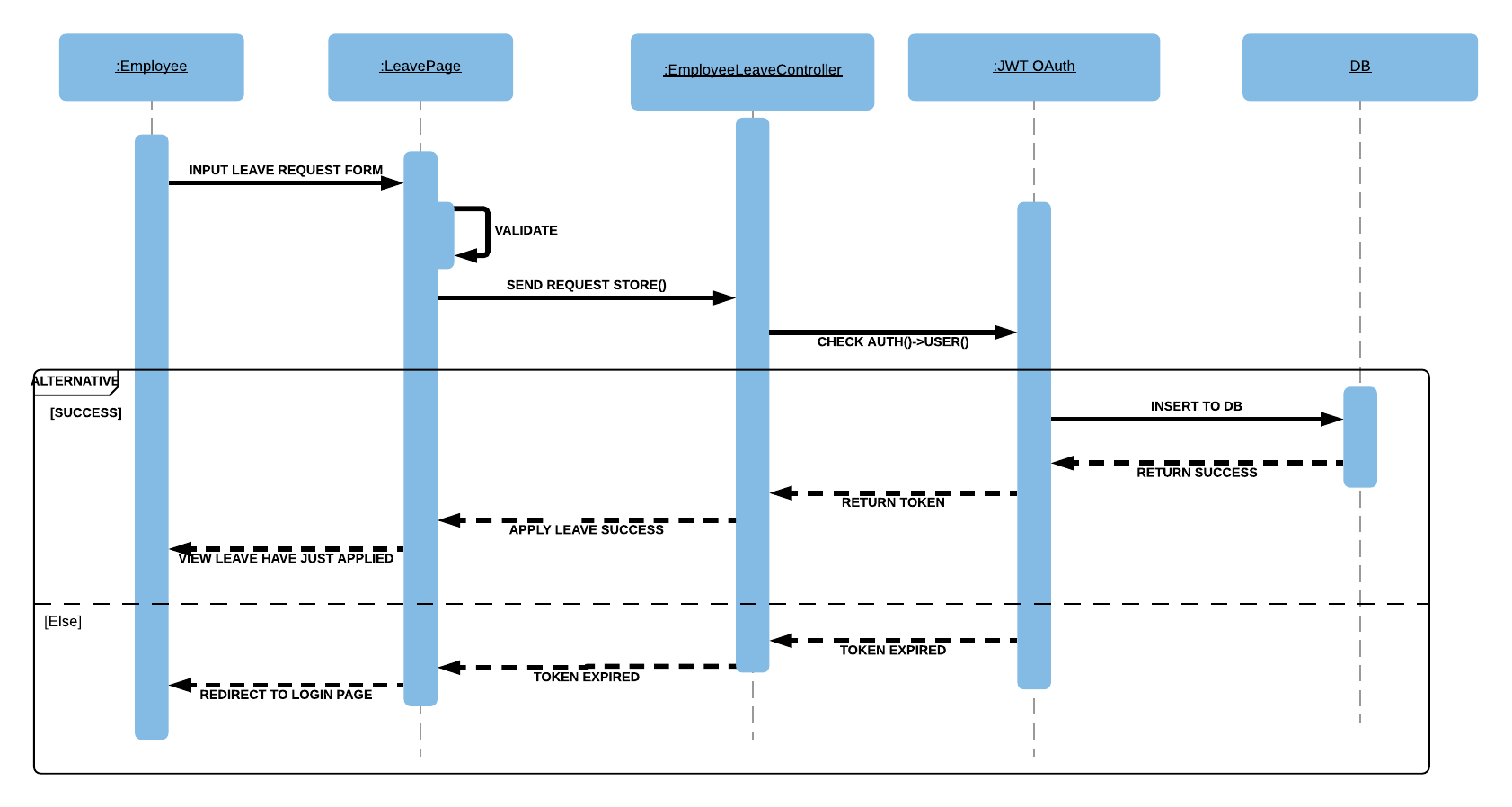


Figure 44. <Sequence Diagram><Employee> Create leave request.

* + 1. <Employee > Create claim request

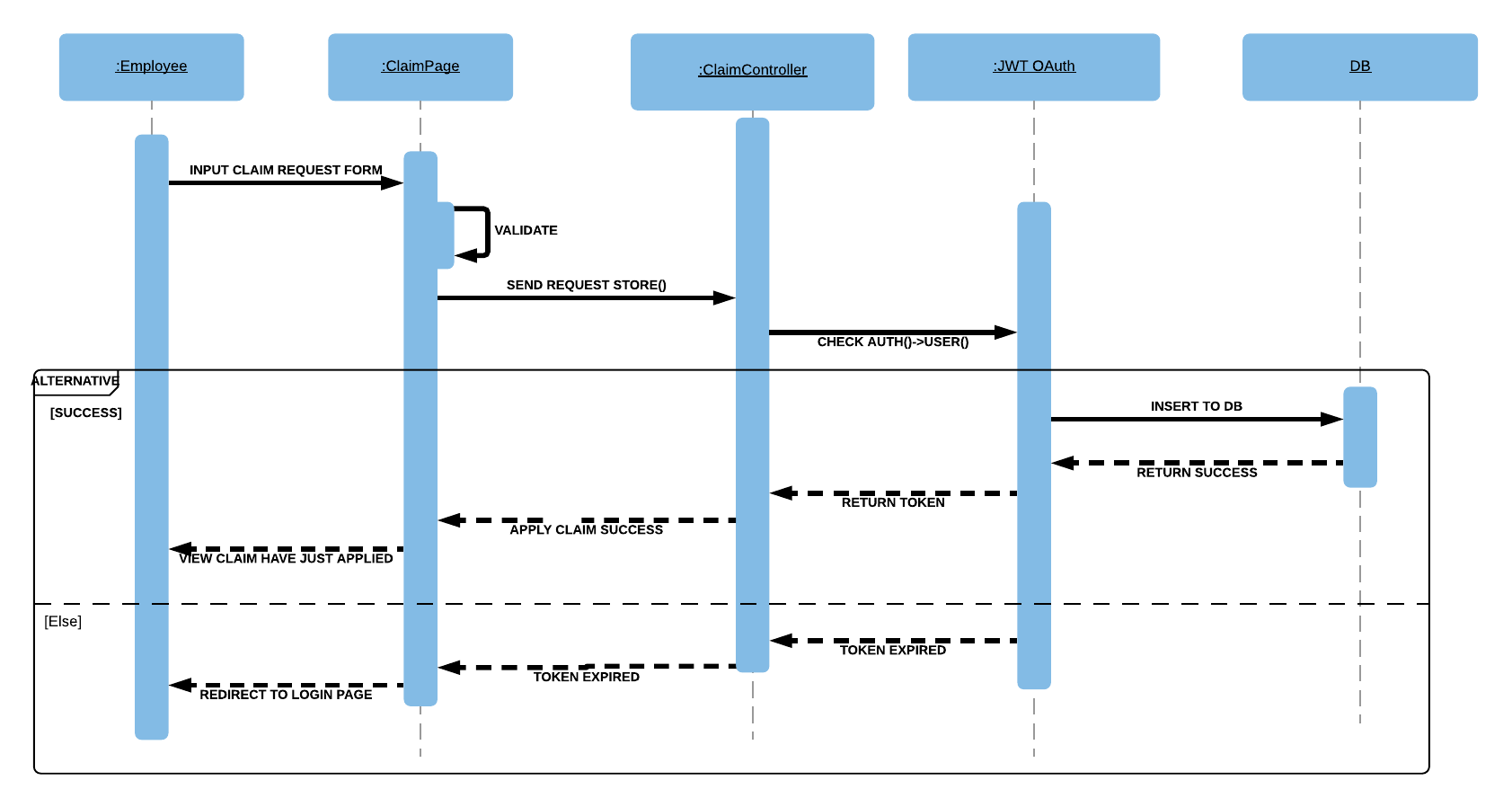


Figure 45. <Sequence Diagram><Employee> Create claim request.

* + 1. <Employee > View leave report

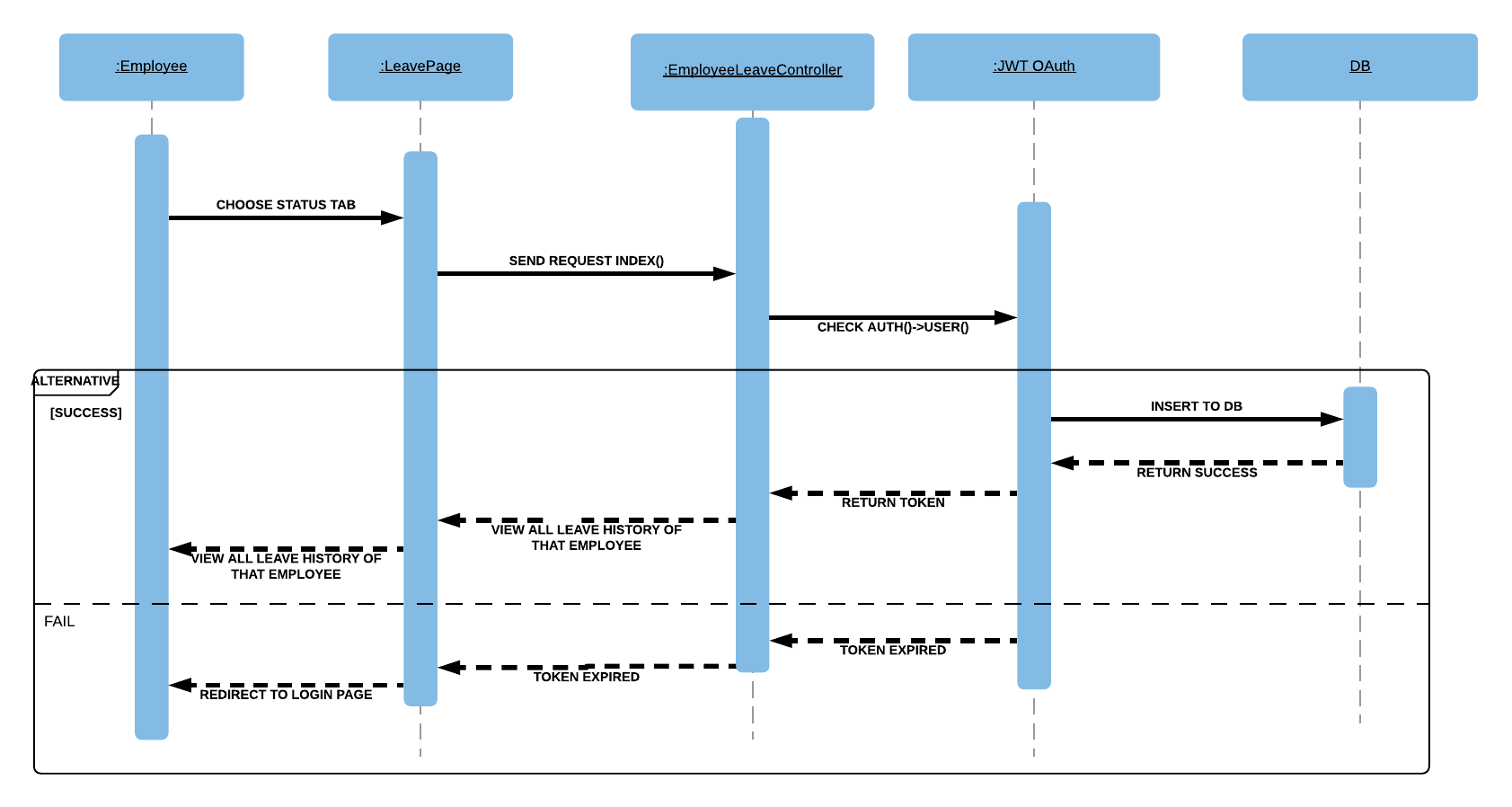


Figure 46. <Sequence Diagram><Employee> View leave report.

* + 1. <Employee > View claim report

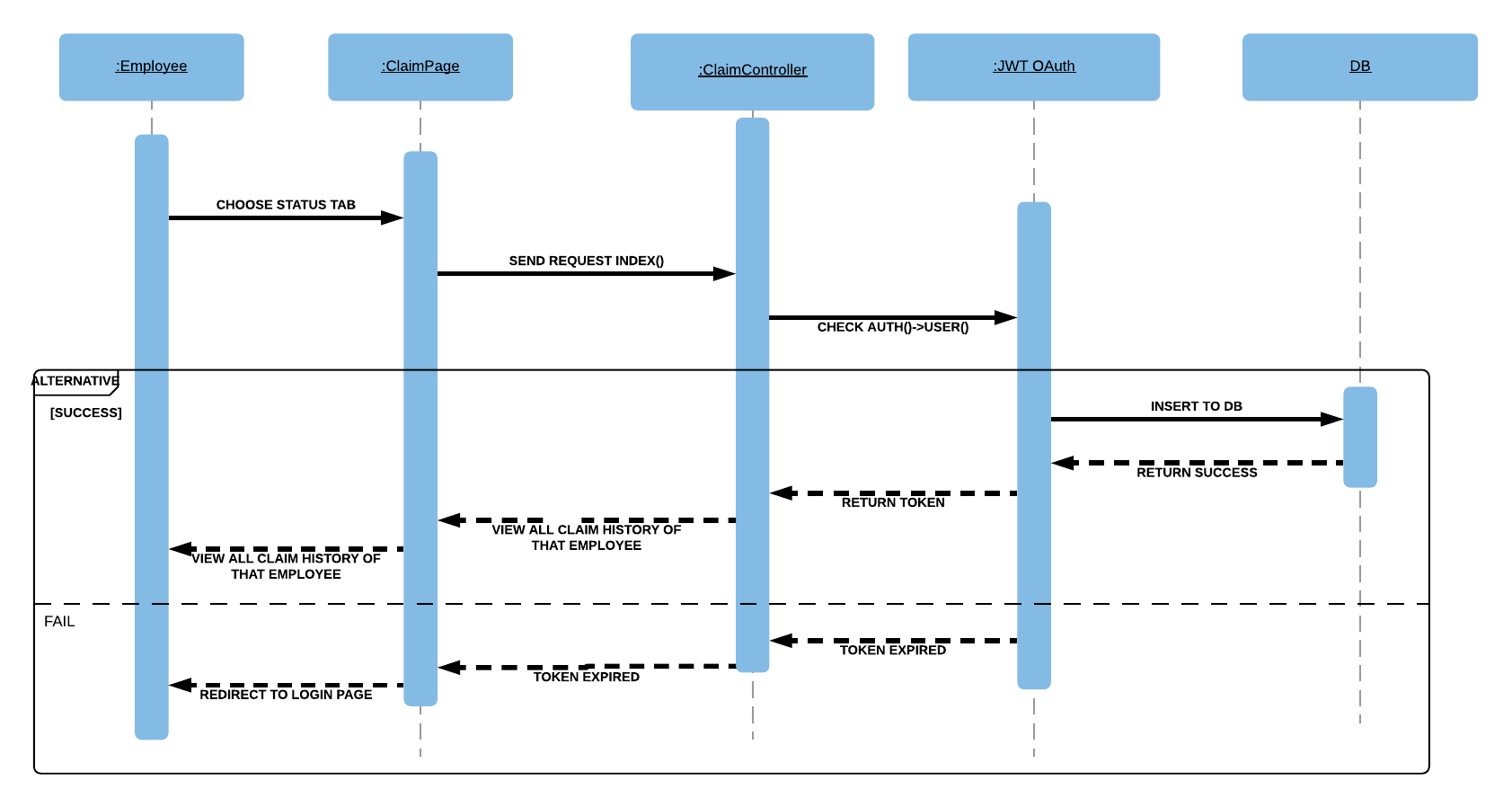


Figure 47. <Sequence Diagram><Employee> View claim report.

* + 1. <Employee > View attendance

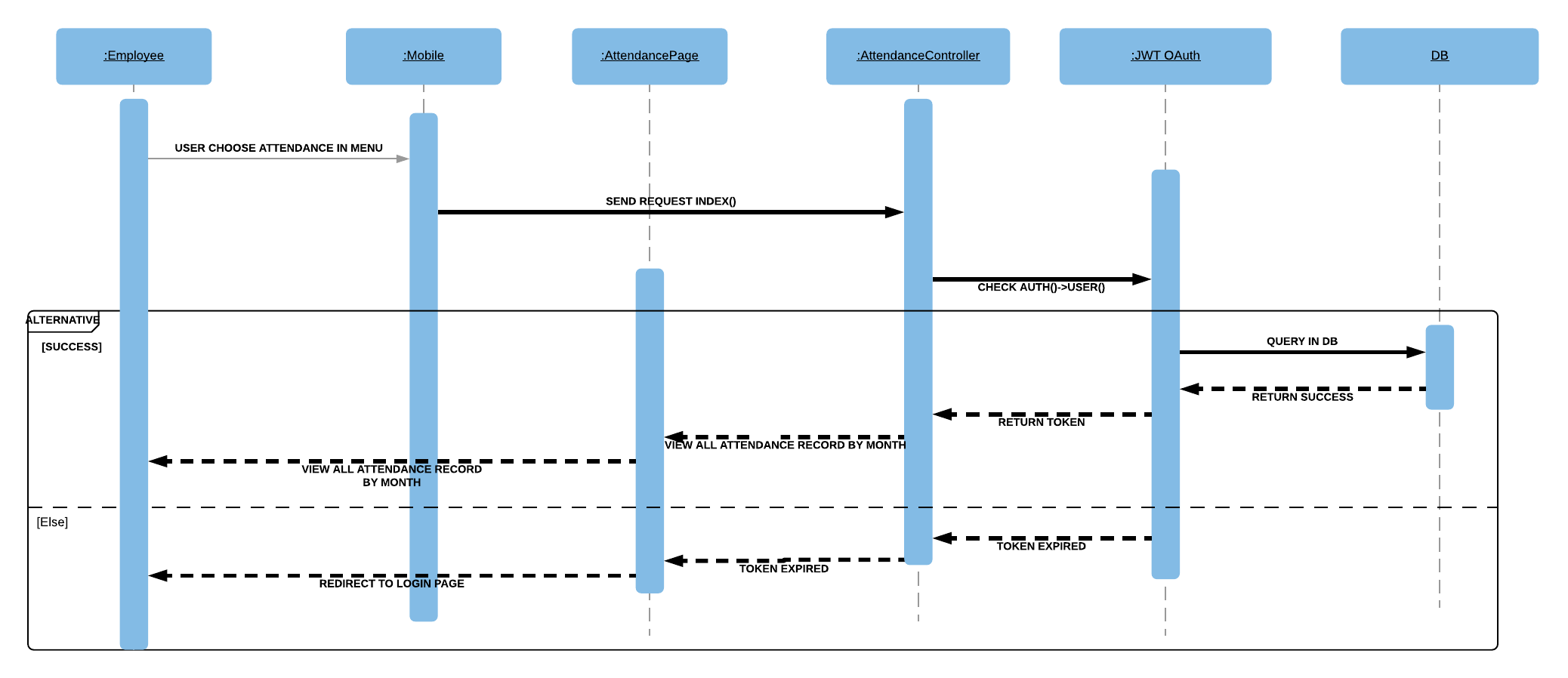


Figure 48. <Sequence Diagram><Employee> View attendance.

* + 1. <Employee > View my attendance

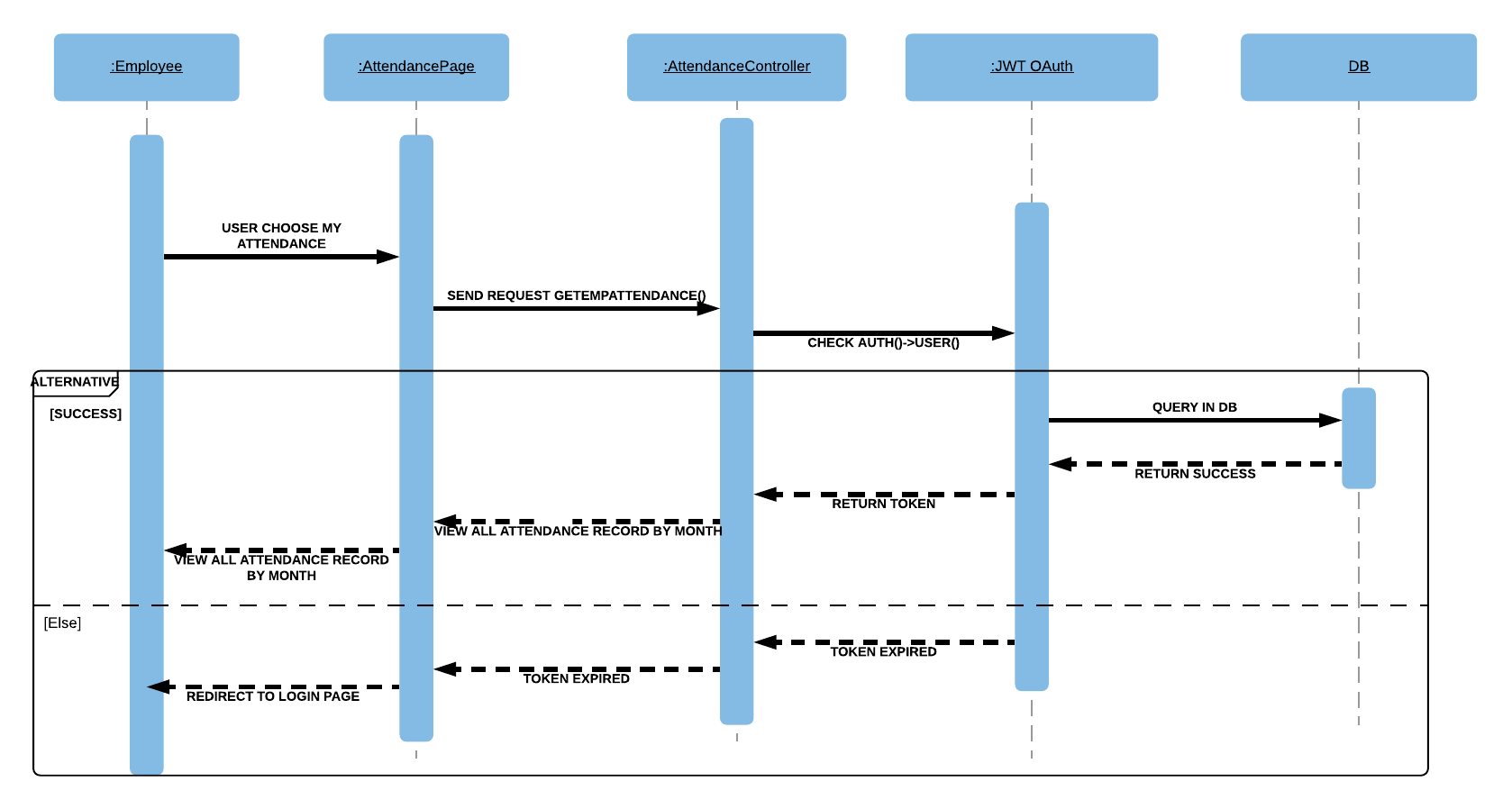


Figure 49. <Sequence Diagram><Employee> View my attendance.

1. User Interface Design
   1. Web Application Interface Design
      1. Home Page (Year view)

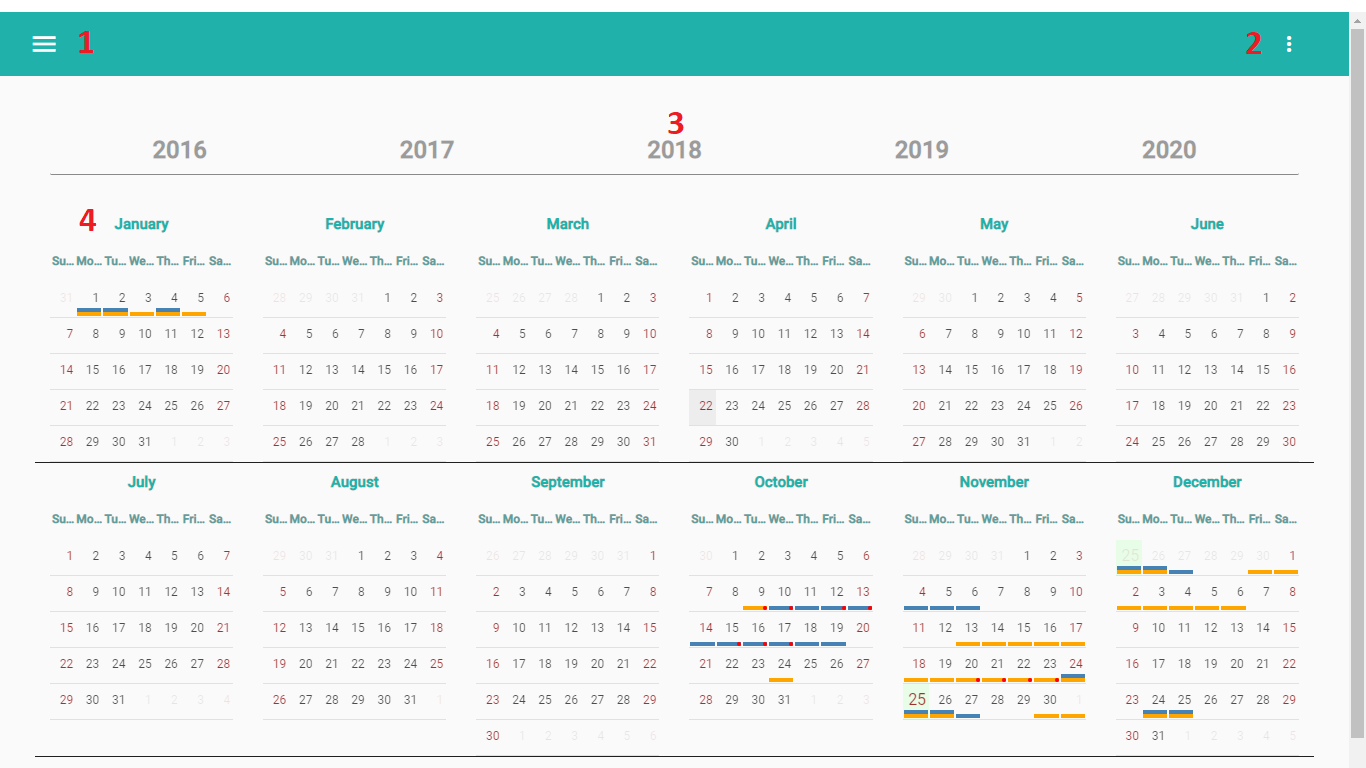


Figure 50. Home Page (Year view)

**Fields :** N/A

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 1 | Open Menu | Open slidebar menu | N/A | Slidebar meu Open. |
| 2 | Open Menu Right Top | Open menu right top | N/A | Menu right top open. |
| 3 | Current Year | Choose year for calendar below | N/A | Set year for yearly calendar bellow. |
| 4 | Detail month | Go to month view | N/A | Navigate to month view. |

Table 48. <Buttons/Hyperlinks> Home page (Year View)

* + 1. Event Management Interface.
       1. View list event.



Figure 51. View event list

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Filter | Search record in list event | No | Yes | Textbox | String |
| 2 | From Date | Start Date of Event | Yes | Yes | Textbox | Date |
| 3 | To Date | End Date of Event | Yes | Yes | Textbox | Date |
| 4 | Event type | Type of Event | Yes | Yes | Textbox | String |
| 5 | Event title | Title of event | Yes | Yes | Textbox | String |
| 6 | Added by | Name of added person. | Yes | Yes | Textbox | String |
| 7 | Date Added | Date when record added. | Yes | Yes | Textbox | String |
| 8 | Remarks | Remarks of event | Yes | Yes | Textbox | String |

Table 49. <Fields> View event list

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 10 | Edit | Edit Event | N/A | From Event |
| 11 | Delete | Delete Event | N/A | Event Deleted |

Table 50. <Buttons/Hyperlinks> View event list

* + - 1. Create event.
         1. Create Event One Time

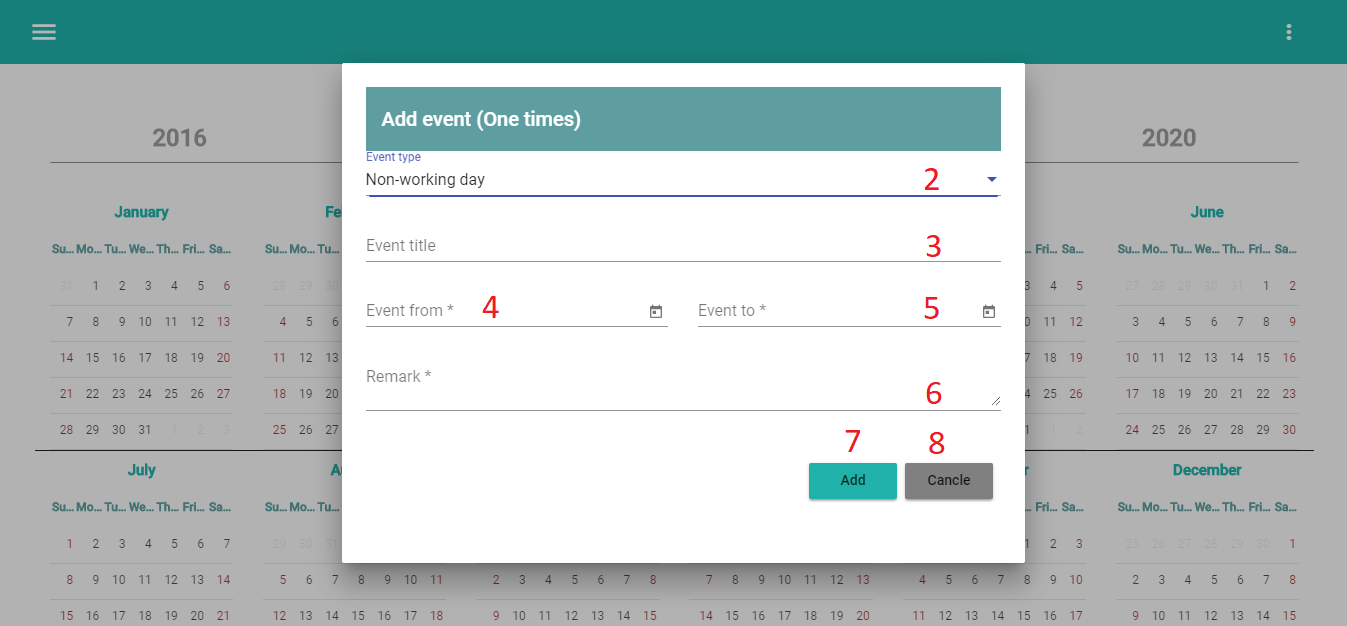


Figure 52. Create Event one time.

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 2 | Event type | Event type | No | Yes | Combobox | String |
| 3 | Event title | Event title | No | Yes | Textbox | String |
| 4 | Event from | Start date of event | No | Yes | Date | Date |
| 5 | Event to | End date of event | No | Yes | Date | Date |
| 6 | Remarks | Remarks of event | No | Yes | TextArea | String |

Table 51. <Fields> Create Event one time.

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 7 | Add | Button add | Input all information. | Added event. |
| 8 | Cancel | Cancel dialog popup | N/A | Dialog closed |

Table 52. <Buttons/Hyperlinks> Create Event one time.

* + - * 1. Create Event Recurring.

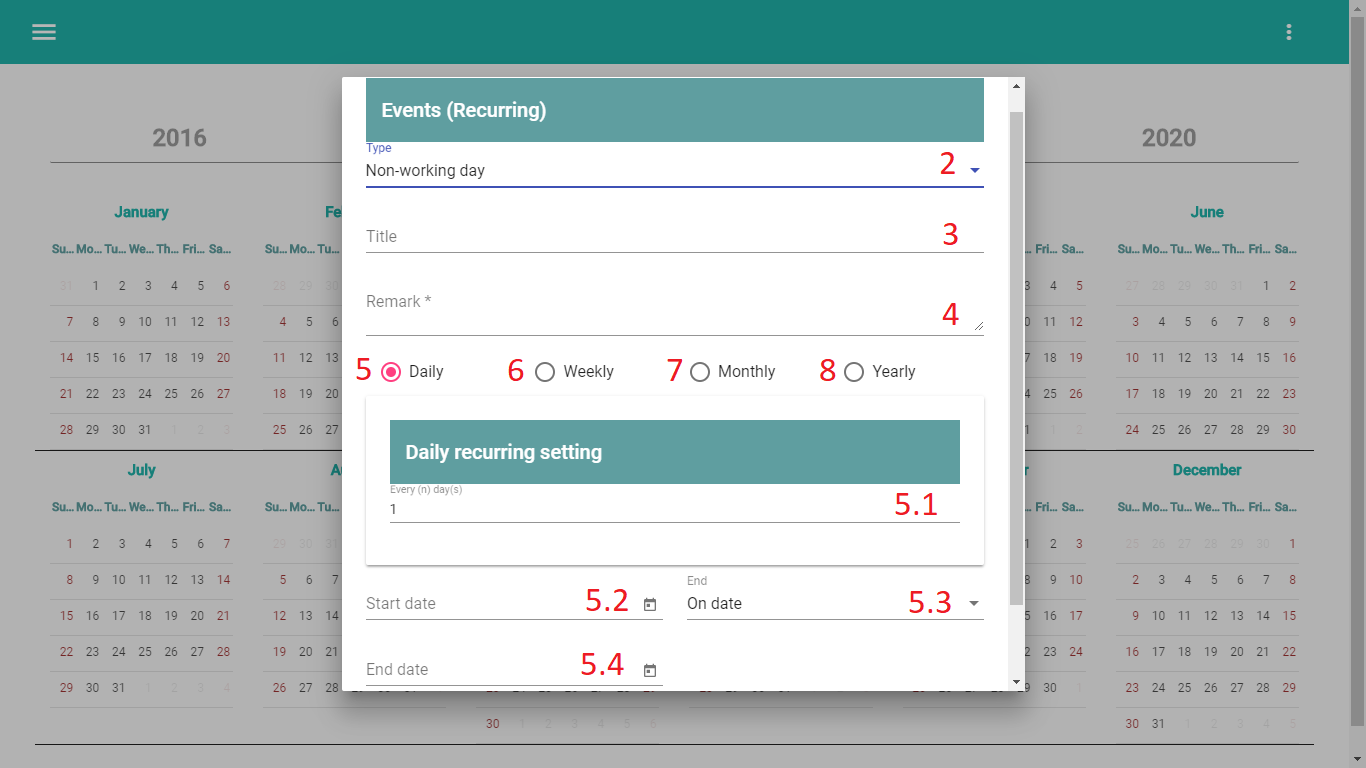


Figure 53. Create Event Recurring

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 2 | Event type | Type of Event | No | Yes | Combo box | String |
| 3 | Event title | Title of Event | No | Yes | Text | String |
| 4 | Remarks | Remarks of Even | No | Yes | Text | String |
| 5.1 | Event(n)day(s) | Day recurring | No | Yes | Text | Number |
| 5.2 | Start Date | Start date of event | No | Yes | Date | Date |
| 5.3 | On Date | On Date | No | Yes | Text | String |
| 5.4 | End Date | End date of event | No | Yes | Date | Date |

Table 53. <Fields> Create Event Recurring

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | Daily | Type Daily of Recurring | N/A | Show form daily recurring. |

Table 54. <Buttons/Hyperlinks> Create Event Recurring

* + 1. Apply Leave Management Interface
       1. View list Apply Leave.

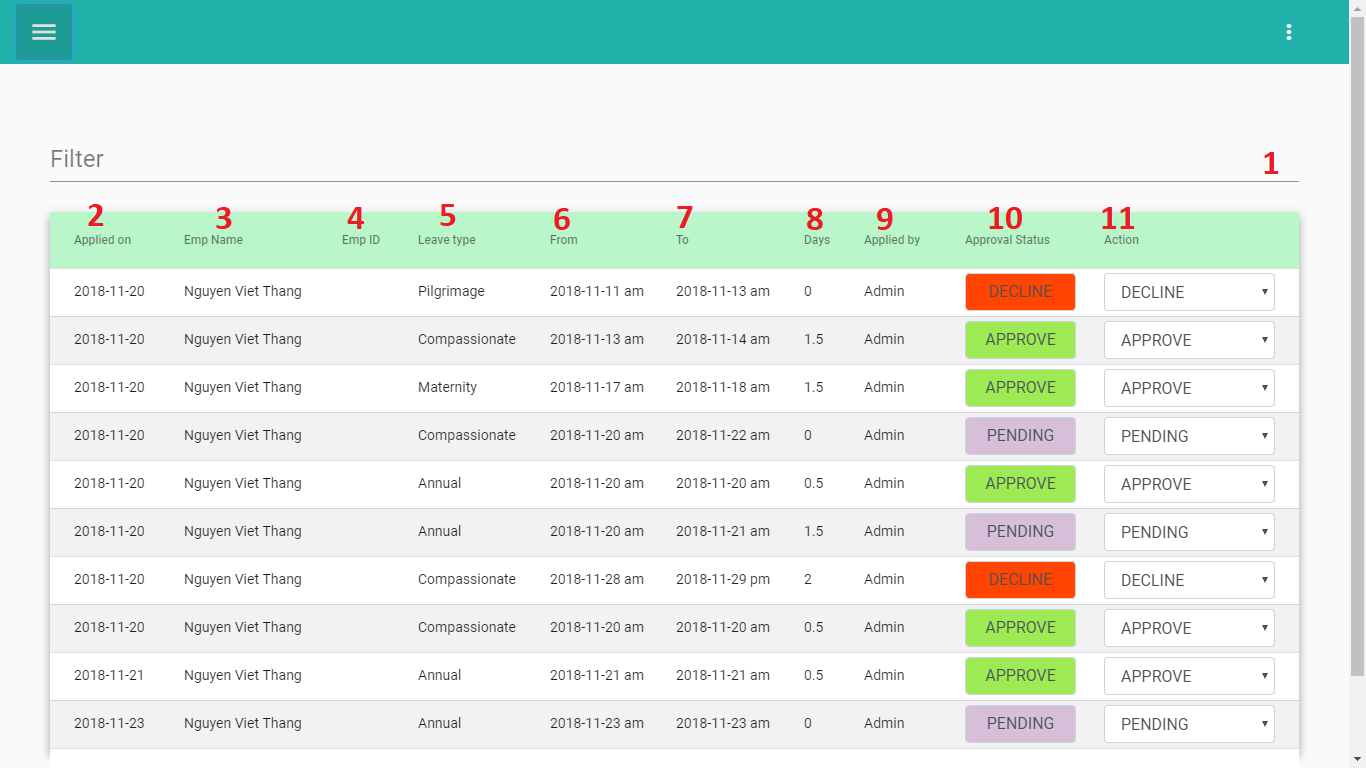


Figure 54. View list Apply Leave

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Filter | Filter record in table | No | Yes | Text | String |
| 2 | Applied on | Date apply record in Database. | Yes | Yes | Text | Date |
| 3 | Emp Name | Name of Employee | Yes | Yes | Text | String |
| 4 | Emp ID | Employee ID | Yes | Yes | Text | String |
| 5 | Leave type | Type of Apply Leave | Yes | Yes | Text | String |
| 6 | From | Start of Apply Leave | Yes | Yes | Date | String |
| 7 | To | End of Apply Leave | Yes | Yes | Date | String |
| 8 | Days | Total days of apply leave | Yes | Yes | Text | Number |
| 9 | Applied by | Name of applied person | Yes | Yes | Text | String |
| 10 | Approval Status | Status of apply leave request | Yes | Yes | Text | String |

Table 55. <Fields> View list Apply Leave.

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 11 | Action | Change status of apply leave request | N/A | Approval Status changed. |

Table 56. <Buttons/Hyperlinks> View list Apply Leave.

* + - 1. Create Apply Leave

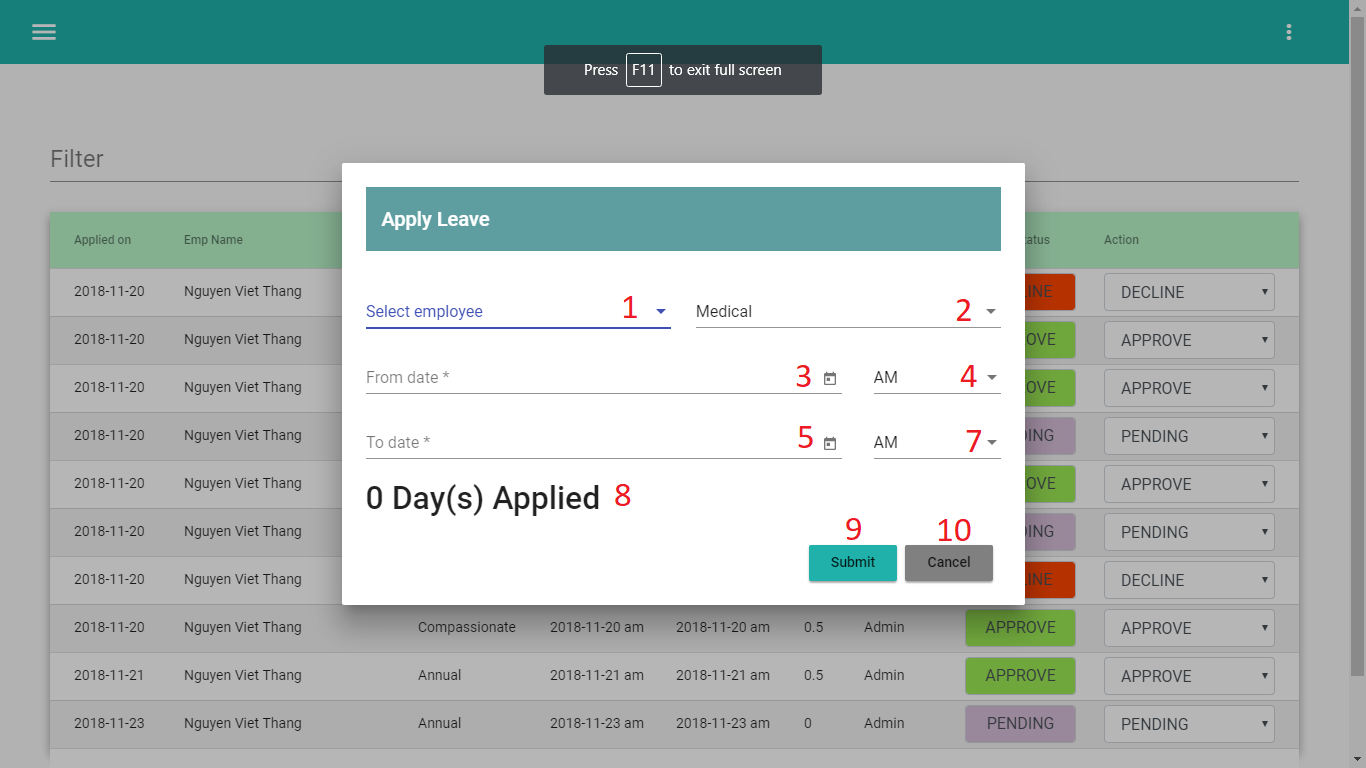


Figure 55. Create Apply Leave

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Select Employee | Select Employee | Yes | Yes | Combo-box | String |
| 2 | Type leave | Select type of apply leave | Yes | Yes | Combo-box | String |
| 3 | From Date | Start date of apply leave | No | Yes | Date picker | Date |
| 4 | AM/PM | Time Apply leave | Yes | Yes | Combo-box | String |
| 5 | To Date | End date of apply leave | No | Yes | Text | String |
| 7 | AM/PM | Time Apply Leave | No | Yes | Text | String |
| 8 | Day applied | Count of day apply | Yes | Yes | Text | String |

Table 57. <Fields> Create Apply Leave

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 9 | Submit | Submit | N/A | Form Apply closed. And show successful message |
| 10 | Cancel | Closed dialog popup | N/A | Dialog popup closed. |

Table 58. <Buttons/Hyperlinks> Create Apply Leave

* 1. Mobile Application Interface Design
     1. Manager’s menu slidebar

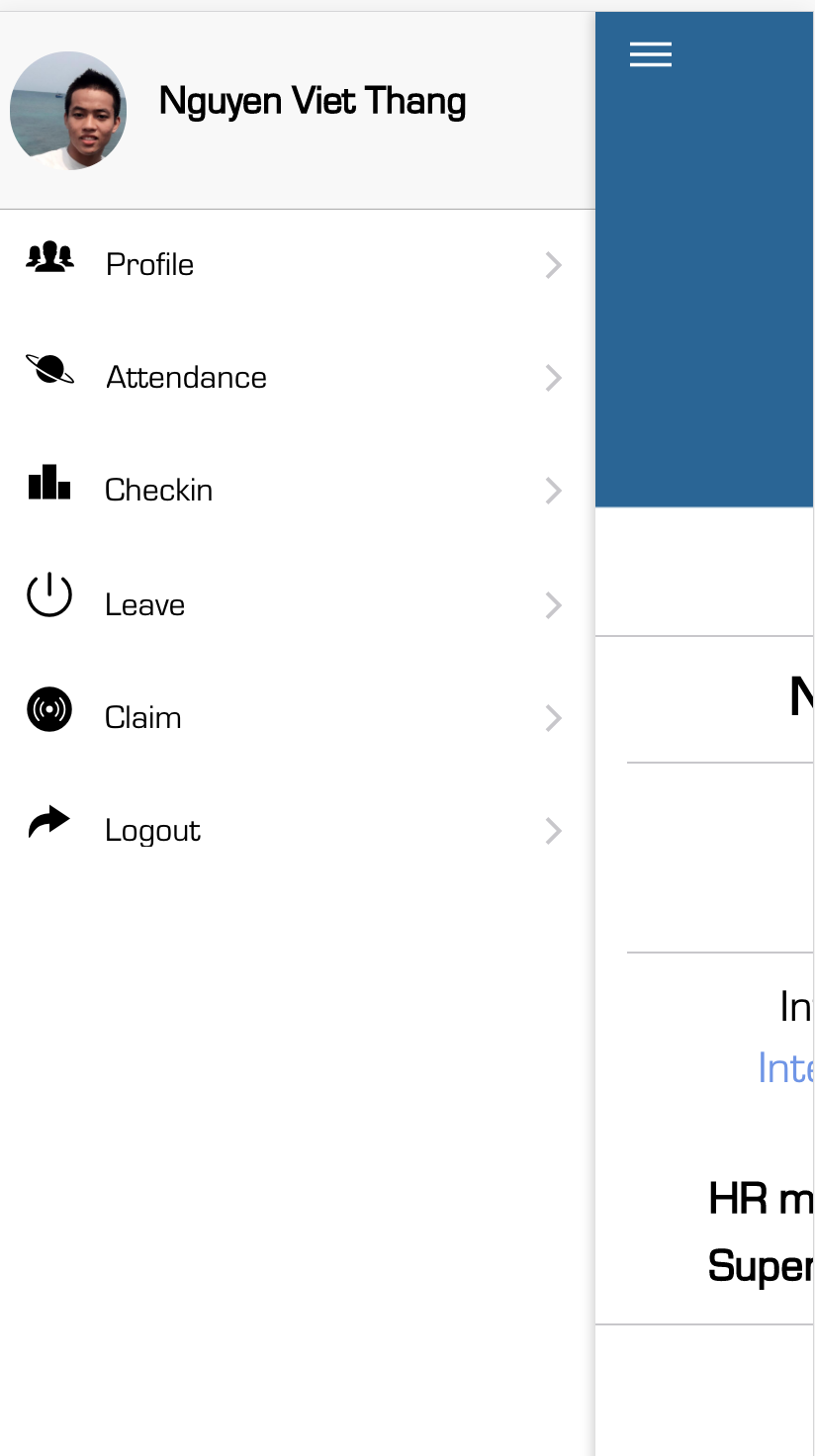


Figure 56. Employee’s menu slidebar

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | AccountInformation | Display user’s full name and role. | Yes | Yes | Text | String |

Table 59. <Fields> Employee’s menu slidebar

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Profile | View Profile | N/A | Navigate to invoice list page |
| 3 | Attendance | View Attendance | N/A | Navigate to proforma invoice list page |
| 4 | Check in / Check out | Checkin / Check out | N/A | Navigate to liabilities page |
| 5 | Leave | Manage leave | N/A | Navigate to liabilities page |
| 6 | Claim | Manage claim | N/A | Navigate to liabilities page |

Table 60. <Buttons/Hyperlinks> Employee’s menu slidebar

* + 1. Attendance

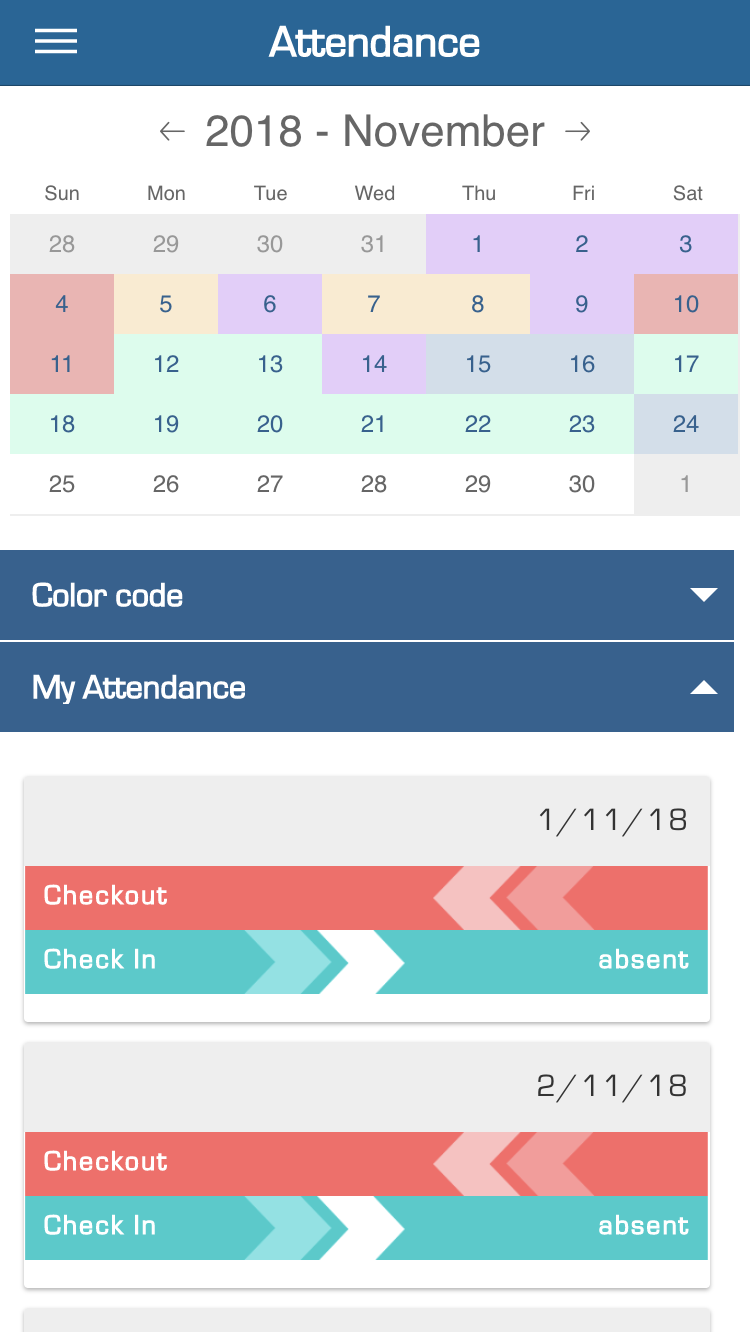


Figure 57. View Attendance

**Fields:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Color Code | Color code | Yes | Yes | Drop down | String |
| 2 | My attendance | View detail attendence | Yes | Yes | Drop down | String |
| 3 | My attendance Report | View report attendance | Yes | Yes | Drop down | String |

Table 61. <Fields> Attendance interface

* + 1. Check in/Check out
       1. Check in

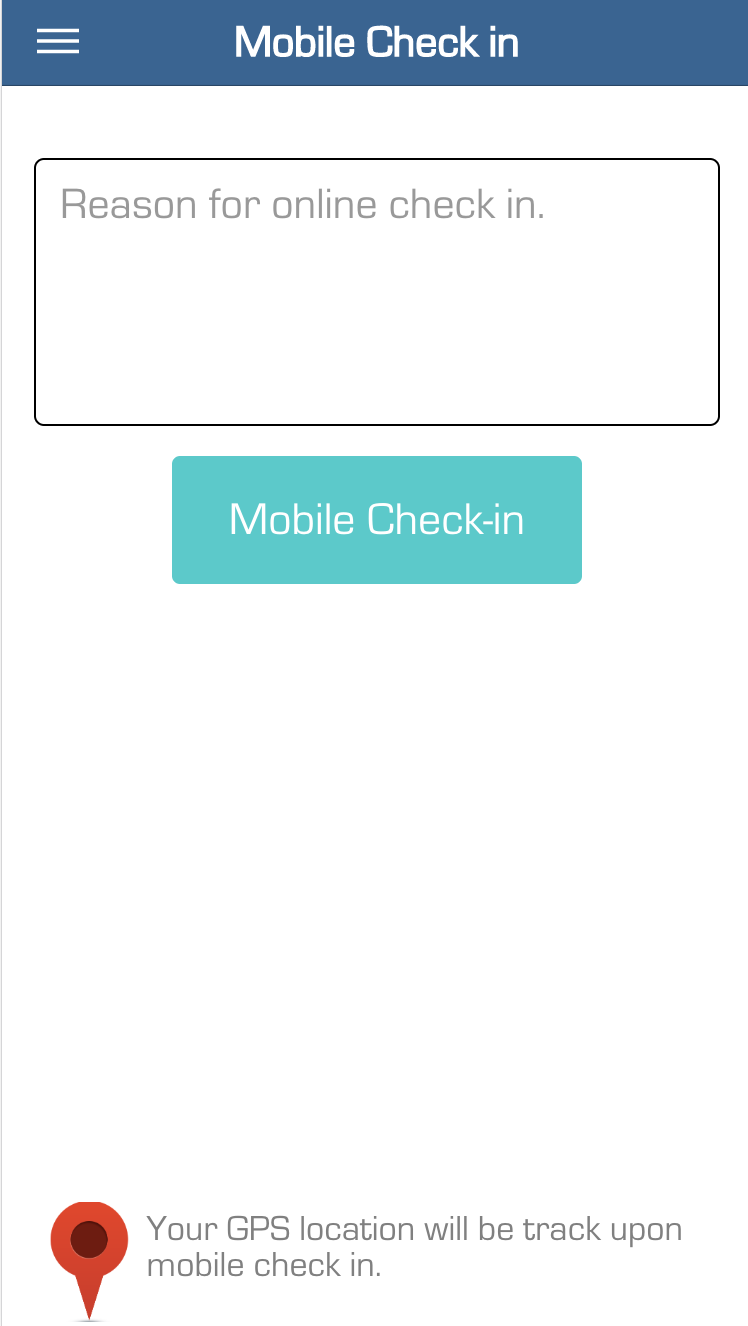


Figure 58. Mobile checkin

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Reasion check in | Reasion for check in | No | Yes | Text ara | String |

Table 62. <Fields> Check in interface

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Check in | Check in | N/A | Check in |

Table 63. <Buttons/Hyperlinks> Check in

* + - 1. Check out

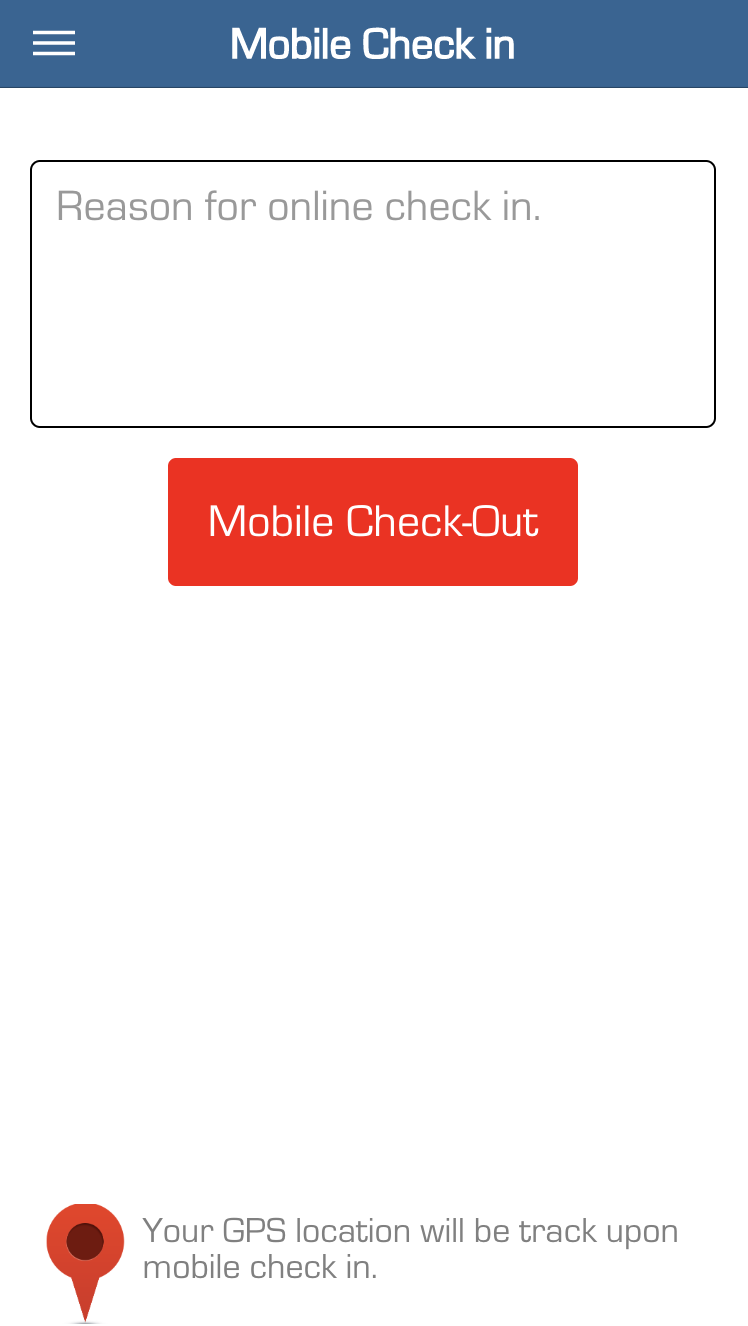


Figure 59. Mobile check out

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Reasion check out | Reasion for check out | No | Yes | Text ara | String |

Table 64. <Fields> Check out interface

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 2 | Check out | Check out | N/A | Check out |

Table 65. <Buttons/Hyperlinks> Check out

* + 1. Leave Manage
       - 1. View Leave

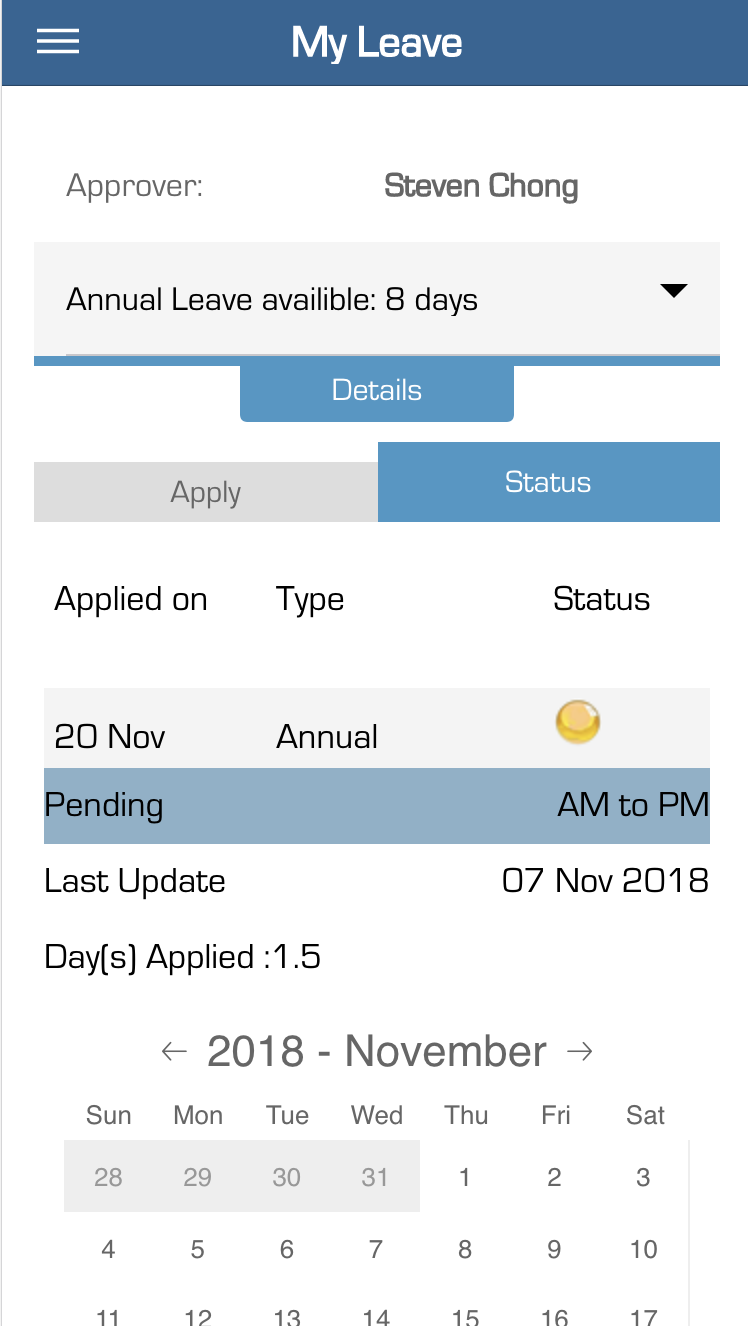


Figure 60. View leave history

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Approver | Name of approver | Yes | Yes | TextBox | String |
| 2 | Availible | Availible for leave | Yes | Yes | TextBox | String |
| 3 | Leave type | Type of leave | Yes | Yes | TextBox | String |
| 4 | Leave status | Status of leave | Yes | Yes | TextBox | String |
| 5 | Last update | Last update of leave | Yes | Yes | Text | String |
| 6 | Day applied | Total day applied | Yes | Yes | Text | String |
| 7 | Calendar | Calendar for leave | Yes | Yes | Text | String |

Table 66. View leave history

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 8 | Detail | Details available leave | N/A | N/A |

Table 67. <Buttons/Hyperlinks> View leave history

* + - * 1. Create leave

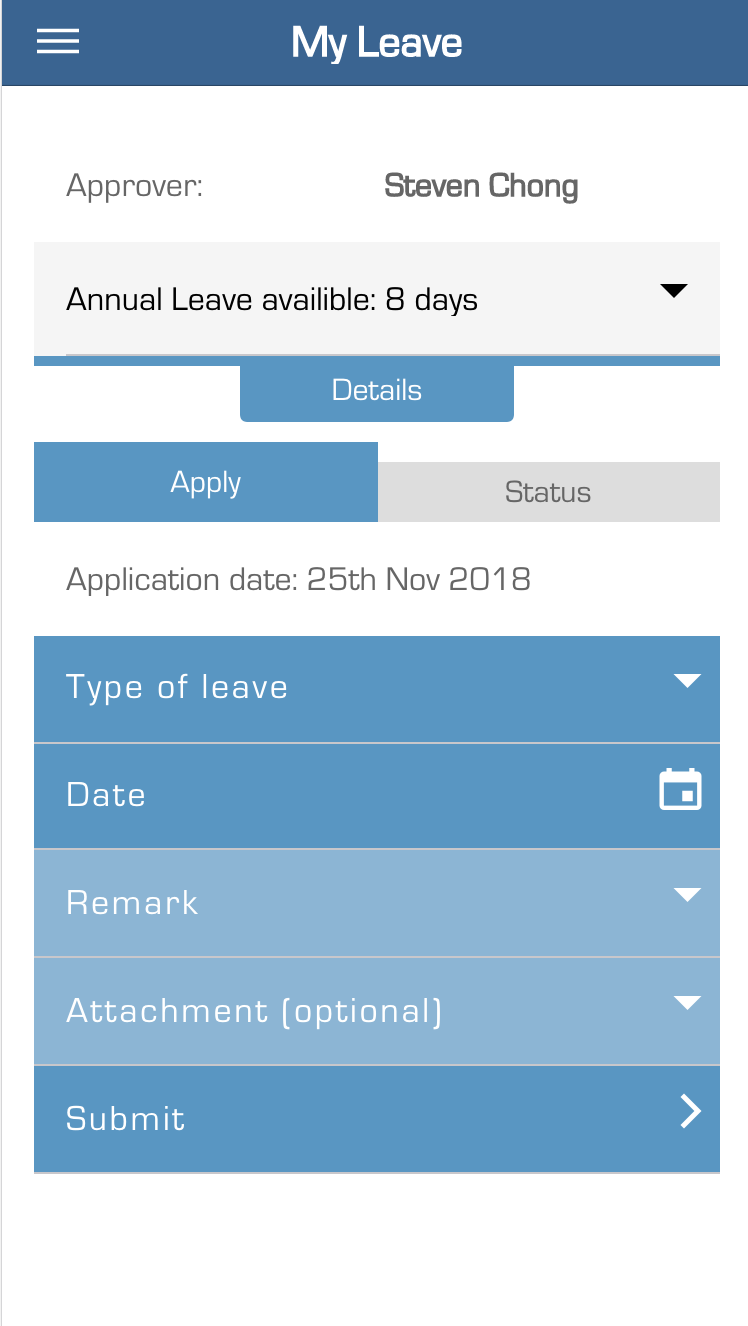


Figure 61. Create leave request

**Fields**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Field Name** | **Description** | **Read only** | **Mandatory** | **Control Type** | **Data Type** |
| 1 | Type of leave | Select type of leave | No | No | Drop down | String |
| 2 | Date | Date of leave | No | Yes | Date picker | String |
| 3 | Remark | Remark of leave | No | No | Textbox | String |
| 4 | Attachment | Image for leave | No | Yes | Textbox | String |

Table 68. <Fields> Create leave request

**Buttons/Hyperlinks:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Function** | **Description** | **Validation** | **Outcome** |
| 5 | Submit | Create leave | Validate required fields | Create leave.  Navigate to product list page. |

Table 69. <Buttons/Hyperlinks> Create leave request

1. Database Design
2. Entity Relationship Diagram (ERD)

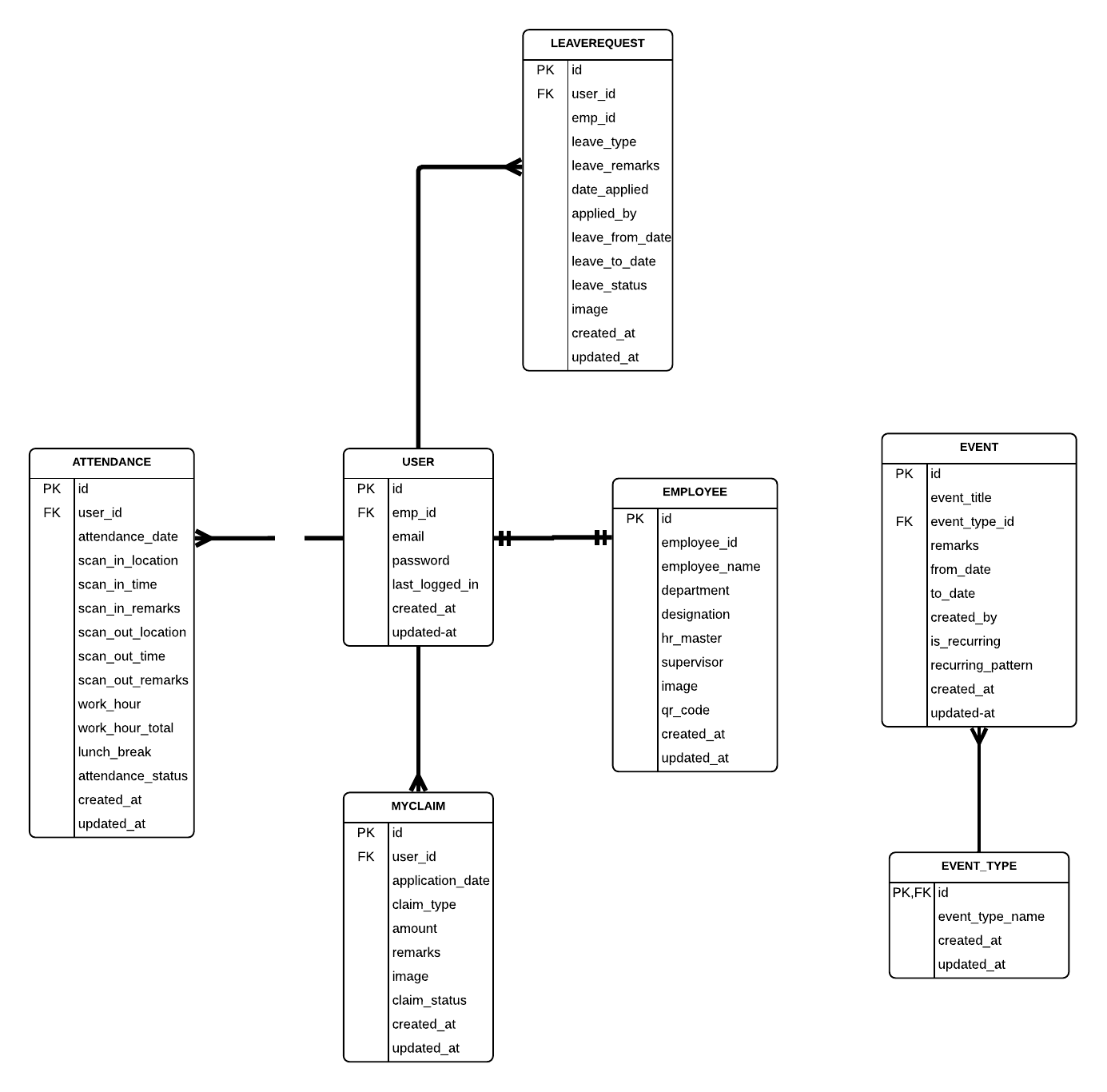


Figure 62. Entity relationship diagram

1. Entity Dictionary

|  |  |
| --- | --- |
| **ENTITY DATA DICTIONARY: DESCRIBE CONTENT OF ALL ENTITIES** | |
| **Entity Name** | **Description** |
| **User** | Contain user information. |
| **Employee** | Contain employee information. |
| **Attendance** | Contain attendance of all employees. |
| **LeaveRequests** | Contain leave requests of all employees. |
| **MyClaim** | Contain claim requests of all employees. |
| **Event** | Contain information of events. |
| **EventType** | Contain information of event type. |

Table 70. Entity dictionary

1. System Implementation & Test
   1. Introduction
      1. Overview

This section describes approach and methodologies used by our team to plan, organize and manage testing process of ECMM and ESS Mobile App before doing the final delivery documents to Track&Roll Company.

It provides all necessary information about implementation and testing procedure of the system including test plans, test cases, test results and pass/fail criteria as well as testing flow to cover all possible cases.  
To go through all the test cases, test plans, test results, the testing procedure will be passed and directly by Jason Ngo – the current project manager of ECMM and ESS project, User experience test is Simon Soo, the demonstration tester is ThangNV, HungTD and testing document is written by HuyNH.

* + 1. Test Approach
* **Goal:** Test main features in the whole ECMM and ESS Mobile based on core flow.
* **Method:** black-box testing.
* **Technique:** check list, pair testing.

The testing process for this project consists of Integration System test level.

System testing is focused on assessing the system’s reliability and the process of testing an integrated system to verify if it meets specified requirements.

Testing will cover functionalities through the use of test interface. This process is concerned with finding errors that result from unanticipated interactions between components and component interface problems.

* 1. Database Relationship Diagram
     1. Physical Diagram

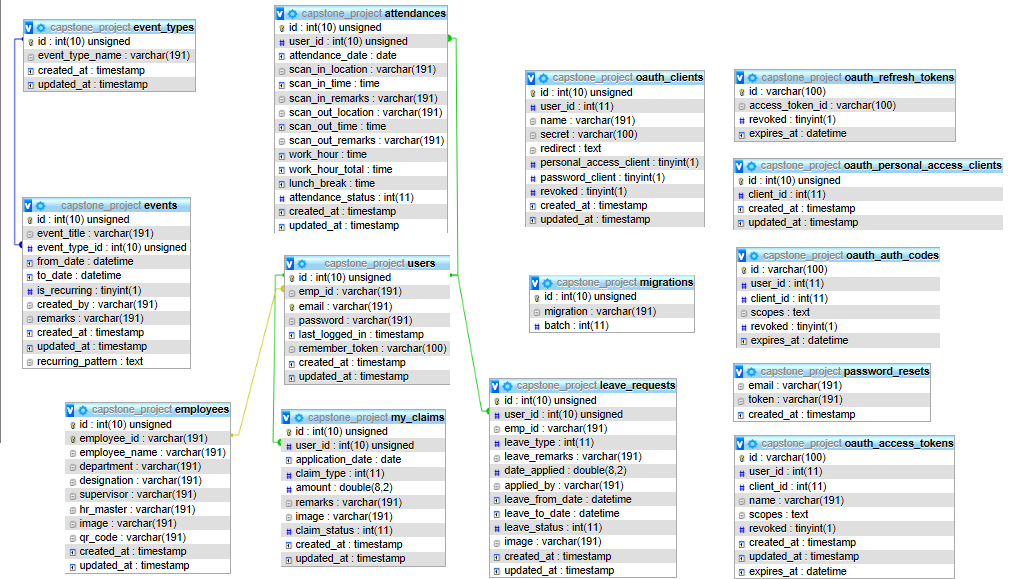


Figure 63. Physical database

* + 1. Data Dictionary

|  |  |  |
| --- | --- | --- |
| **Entity Data Dictionary: Describe content of all tables** | | |
| **No.** | **Table Name** | **Description** |
| 1 | oauth\_refresh\_token | Refresh tokens are usually subject to strict storage requirements to ensure they are not leaked. They can also be blacklisted by the authorization server |
| 2 | oauth\_personal\_access\_client | users may want to issue access tokens to themselves without going through the typical authorization code redirect flow |
| 3 | oauth\_auth\_codes |  |
| 4 | oauth\_access\_token | information contained in the token to decide whether the client is authorized or not. Access tokens usually have an expiration date and are short-lived |
| 5 | oauth\_clients | creates the private and public key needed for encryption |
| 6 | migrations | Contains all the tables using Code first. |
| 7 | events | Cointain information about events |
| 8 | event\_types | Contain types of events |
| 9 | users | Contain information about users. |
| 10 | employees | Contain information of the employees |
| 11 | attendances | Contain information about items of each proforma invoice |
| 12 | my\_claims | Contain template information |
| 13 | leave\_requests | Contain invoice information |

Table 71. Database dictionary

* 1. Test Plan

The purpose of this section is to verify and ensure that ECMM and ESS Mobile Application meets its entire technical, functional and business requirements.The test case is responsible by the Project Manager Jason Ngo, demonstration tester is Nguyen Viet Thang and Tran Dang Hung, finally is testing documentation is written by Nguyen Hoc Huy.

The following parts describe which features will be tested and which will be not.

* + 1. Features To Be Tested

For the Calendar web application:

* Features of creating events.
* Feature of creating leave request.
* Calculating days of leave
* Applying leave request on blocked day.
* Update leave status which each status given.
  + 1. Features Not To Be Tested
* Log in, log out, update profile,…
  1. System Testing Test Case
     1. Test Case
        + 1. View events

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Acceptance Testing (UAT) : Test Cases for Section 2.1 <View Events>** | | | | | | |
| **Test Number** | **Functional Area** | **Test Description** | **Date for Testing** | **Resource for Testing** | **Expected Results** | **Status** |
| 1 | Calendar Events (ADD) - Database 'events' | Users when create new new event One time/Recurring (Type 1 - 2 - 3) without inputted anything, leave blank | 29th October, 2018 | HuyNH | The popup should return red field - means you have to input the information into popup form | PASS |
| 2 | Calendar Events (ADD) - Database 'events' | Users when create a new event One time/Recurring (Type 1 - 2 - 3) with inputted data, return Created event successfully and shown in Calendar | 29th October, 2018 | HuyNH | The event is created successfully and shown in Calendar | PASS |
| 3 | View all event (One time/Recurring) | Users click on View all events to view all the events created | 1st November 2018 | HuyNH | The events are shown successfully | PASS |
| 4 | Delete event (One time/Recurring) | Users choose 1 event and click delete button | 1st November 2018 | HuyNH | The events deleted successfully and not shown in calendar | PASS |
| 5 | Edit event | User choose 1 event to edit the information | 5th November | HuyNH | The events is edited successfully and updated in Calendar | PASS |

Table 72. <Test Case> View events

* + - * 1. View lists of events

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| User Acceptance Testing (UAT) : Test Cases for Section 2.2 <View lists of events> | | | | | | |
| Test Number | Functional Area | Test Description | Resource for Testing | Expected Results | Status |
| 1 | View events report in yearly | User click on yearly view and hower to 1 day with dot button | HuyNH | The hover box shows all of the events and leave request | PASS |
| 2 | View events report in monthly | User click on specific month and view all events with type of event | HuyNH | The calendar in month view all events with event type | PASS |
| 3 | View events report in daily | User wants to view all events in daily | HuyNH | All the events occur in daily is shown | PASS |
| 4 | Color code | View in yearly: events (blue) - Leave request (orange)  With more than 1 event and leave request: small red dot at the end | HuyNH | View in yearly: events (blue) - Leave request (orange)  With more than 1 event and leave request: small red dot at the end | PASS |
| 5 | Color code | Leave request have 4 color with 4 status | HuyNH | View all 4 color when changing the status and update color in Calendar | PASS |
| 6 | Filter leave request by employeeID | Choose employeeID that HR Manager wants to filter into report and view status of the employee leave request | HuyNH | All the leave request of that employee will be shown with status of the leave application form | PASS |
| 7 | Optional: count events | When user choose month to view, they will show total of events in that day | HuyNH, HungTD | Count all the events in each day | PASS |
| 8 | Filter events by event type | When user choose type of event to filter, it will show all the events in the same type | HuyNH, HungTD | Filter all the events by type | PASS |
| 9 | Filter by event type and leave request together | Choose types of events that HR Master wants to filter | HuyNH, HungTD | Both filter events and leave request | PASS |

Table 73. View lists of events

* + - * 1. Manage leave request

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Acceptance Testing (UAT) : Test Cases for Section 2.3 <Manage Leave Request>** | | | | | | |
| **Test Number** | **Functional Area** | **Test Description** | **Resource for Testing** | **Expected Results** | **Status** |
| 1 | Apply leave request of an employee | Users when choose to create new leave request and dont fill in the form | HuyNH | The pop up with red line appear and require input data into form | PASS |
| 2 | Apply leave request of an employee | Users when choose to create new leave request and fill all the required data in the form | HuyNH | The message create leave request successfully and status is pending | PASS |
| 3 | View leave request in yearly | When user hover on the specific day, the hover box show all the leave request of that day | HuyNH | Hover box show all the leave request with status in that day | PASS |
| 4 | View leave request in monthly | When user choose to view in month, there will be 3 color code with each status of the leave request | HuyNH | View the 3 color of the leave request (Pending, Approve, On Hold) | PASS |
| 5 | View leave request in daily | When user choose one day, view all the leave request of that day | HuyNH | View all the leave request in that day | PASS |
| 6 | Update status of leave request | When user wants to update the status of leave request, the color code will change with each status | HuyNH | When user update the status of leave request, the leave request will change the code of the color | PASS |
| 7 | Delete the leave request | When user view leave request in day and choose one leave request to delete, the message box is shown to confirm delete the event | HuyNH | Return message delete events successfull and back to list of leave request | PASS |
| 8 | Apply leave request from Afternoon to morning | When HR Master apply leave request from afternoon to morning in same day | HuyNH, HungTD | Cannot apply leave | PASS |
| 9 | Update the status to Decline | When the leave request change to decline, will remove the application | HuyNH, HungTD | Keep on DB and remove from calendar | PASS |
| 10 | Update the status to Cancel | When the leave request is cancelled, the leave request will be removed from calendar | HuyNH, HungTD | Removed on calendar and database | PASS |

Table 74. <Test Case> Manage leave request

* + - * 1. Apply leave request for type 1 and type 2 events

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Acceptance Testing (UAT) : Test Cases for Section 2.3.1 <Apply leave request for type 1 and 2 events>** | | | | | | |
| **Test Number** | **Functional Area** | **Test Description** | **Resource for Testing** | **Expected Results** | **Remarks** |
| 1 | Apply leave request for type 1 | From 23-Jan-18 To 24-Jan-18 | HuyNH | Calculate should be 0 days | PASS |
| 2 |  | From 1-Jan to 4-Jan-18 | HuyNH | Calculate should be 0 days | PASS |
| 3 |  | From 06-Dec-18 AM to 11-Dec-18 AM is 4.5 days | HuyNH | From 06-Dec-18 AM to 11-Dec-18 AM is 4.5 days | PASS |
| 4 |  | From 06-Dec-18 AM to 11-Dec-18 PM is 5.0 days | HuyNH | From 06-Dec-18 AM to 11-Dec-18 PM is 5.0 days | PASS |
| 5 |  | From 07-Dec-18 AM to 07-Dec-18 AM is 0 days | HuyNH | From 07-Dec-18 AM to 07-Dec-18 AM is 0 days | PASS |
| 6 |  | From 07-Dec-18 PM to 07-Dec-18 PM is 0 days | HuyNH | From 07-Dec-18 PM to 07-Dec-18 PM is 0 days | PASS |
| 7 |  | From 07-Dec-18 AM to 08-Dec-18 AM is 0.5 days | HuyNH | From 07-Dec-18 AM to 08-Dec-18 AM is 0.5 days | PASS |
| 8 | Apply leave request for type 2 | From 2 Jan to 4 Jan | HuyNH | Blocked day so cannot apply leave | PASS |
| 9 |  | From 1-Dec To 1-Dec 2018 | HuyNH | Blocked day so cannot apply leave | PASS |
| 10 |  | From 4-Dec To 6-Dec-2018 | HuyNH | Blocked day so cannot apply leave | PASS |
| 11 |  | From 10-Dec to 12-Dec-2018 | HuyNH | Blocked day so cannot apply leave | PASS |
| 12 |  | From 16-Dec to 16-Dec 2018 | HuyNH | Blocked day so cannot apply leave | PASS |
| 13 | Apply leave in no public holiday/off day | From 06-Dec-18 AM to 11-Dec-18 AM is 5.5 days | HuyNH | From 06-Dec-18 AM to 11-Dec-18 AM is 5.5 days | PASS |
| 14 |  | From 06-Dec-18 AM to 06-Dec-18 AM is 0.5 days | HuyNH | From 06-Dec-18 AM to 06-Dec-18 AM is 0.5 days | PASS |
| 15 |  | From 06-Dec-18 AM to 11-Dec-18 PM is 6.0 days | HuyNH | From 06-Dec-18 AM to 11-Dec-18 PM is 6.0 days | PASS |
| 16 | Apply leave request for type 1 | From 25th Dec AM To 25th Dec PM | HungTD, HuyNH | Calculate should be 0 days | PASS |
| 17 | Apply leave request type 2 recurring | From 25th March to 27th March | HungTD, HuyNH | Blocked day so cannot apply leave | PASS |

Table 75. <Test Case> Apply leave request for type 1 and type 2 events

* + - 1. Mobile Test Case

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Number** | **Functional Area** | **Test Description** | **Date for Testing** | **Resource for Testing** | **Expected Results** | **Actual Results** | **Status** |
| 1 | View Profile | When user view profile | <Date tested> | <Performed By> | User can see employee had 2 avata image. | Employee had two avata image | PASS |
| 2 | Check in | When user Check in | 29th October, 2018 | HungTD | Check in successful | Check in successful | PASS |
| 3 | Check out | When user Check out | 29th October, 2018 | HungTD | Can't check out if not checkin | Can't check out because not check in. | PASS |
| 4 | Send Apply Leave | When user send apply leave with no data | 1st November 2018 | HungTD | Can't send apply leave | Can't send apply leave | PASS |
| 5 | Send Apply Leave | When user fill all data and send apply leave | 1st November 2018 | HungTD | Send apply successful | Send apply leave successful | PASS |
| 6 | Check out | When user exist check-out | 5th November | HungTD | User can't check out if had been check out | Can't checkout | PASS |
| 7 | Login | When user login | 5th November | HungTD | User can't login with wrong account | Can't login | PASS |
| 8 | All Function | When token is expires | 5th November | HungTD | Navigate to Login page when token is expires | Navigate to login page | PASS |
| 9 | Apply Leave | User can see event when press in day had token on monthly calendar in apply leave pages | 10th November | HungTD | Event appear below the calenar | Event appear below calendar | PASS |
| 10 | Check out | User can see total work hour when check out | 12th November | HungTD | Appear toral work hour when check out | Total work hour appear when check out | PASS |

**F. Software User’s Manual**

**1. Installation Guide**

The specifications are based on dependencies requirements and performance test results from previous section of this document.

### **1.1. Server Requirement**

|  |  |
| --- | --- |
| Criteria | Minimum Requirements |
| Internet Connection | Cable (4 Mbps) |
| Computer Processor | 64-bit CPU 1.4GHz with 2 cores |
| Computer Memory | 4 GB RAM |

Table 76. Server requirement

### **1.2. Backend Requirement**

|  |  |
| --- | --- |
| Name | Name/Version |
| Operating System | Windows 8 or higher |
| IDE | Sublime Text 3, Visual Studio Code,… |
| DBMS | XAMPP Server with MySQL |
| Web browser | Firefox (v52 or higher), Chromes (v50 or higher) |

Table 77. Backend requirement

**2. User Manual**

### **2.1. Web Application**

1. **View Event in Yearly.**



Figure 64. <HR Master> View event and apply leave in yearly calendar.

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Click on day of month with underlight. |

Table 78. <User Manual> View event and apply leave in yearly calendar.

1. **Create New Event (One time)**

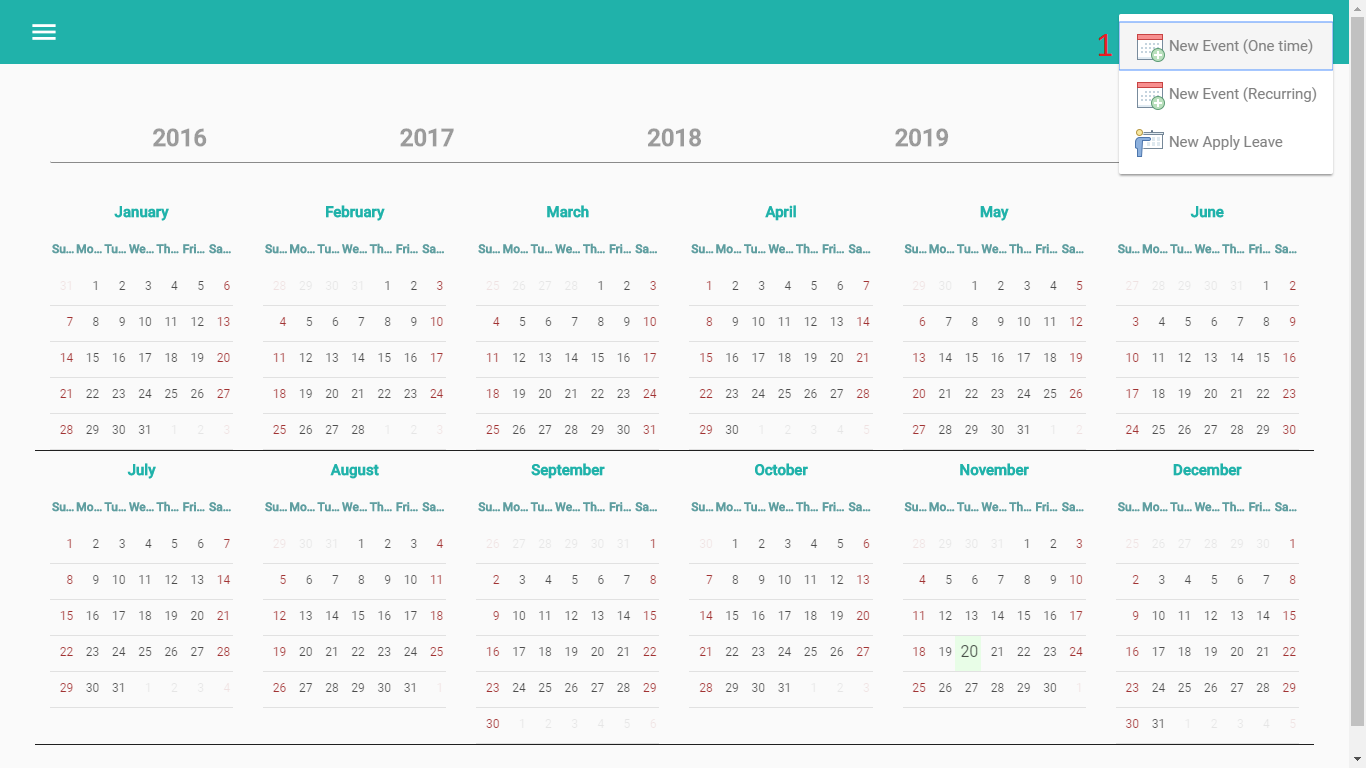


Figure 65. <HR Master> Create event one time (1)

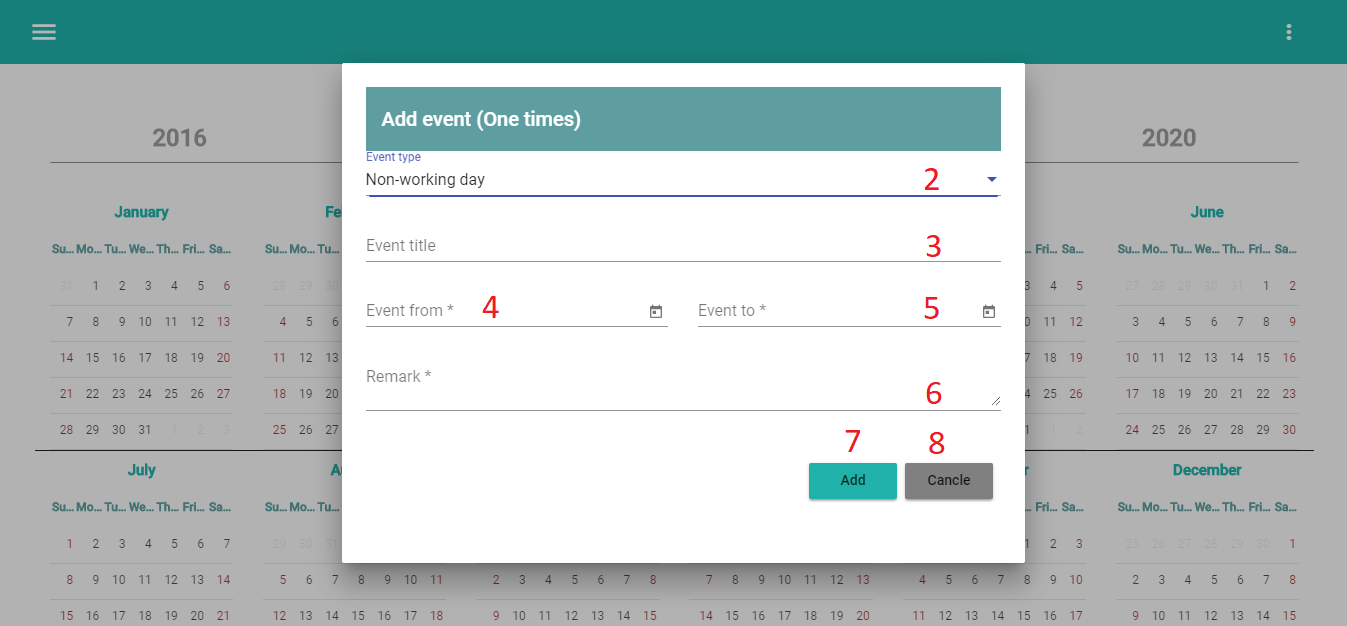


Figure 66. < HR Master > Create event one time (2)

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Click on “New event (one time) button. |
| 2 | Select “Event type” |
| 3 | Fill in Event title |
| 4 | Select Event from |
| 5 | Select Event to |
| 6 | Fill in Remarks |
| 7 | Click button “Add” |

Table 79. <HR Master> Create new one time event.

1. **Create new recurring event.**

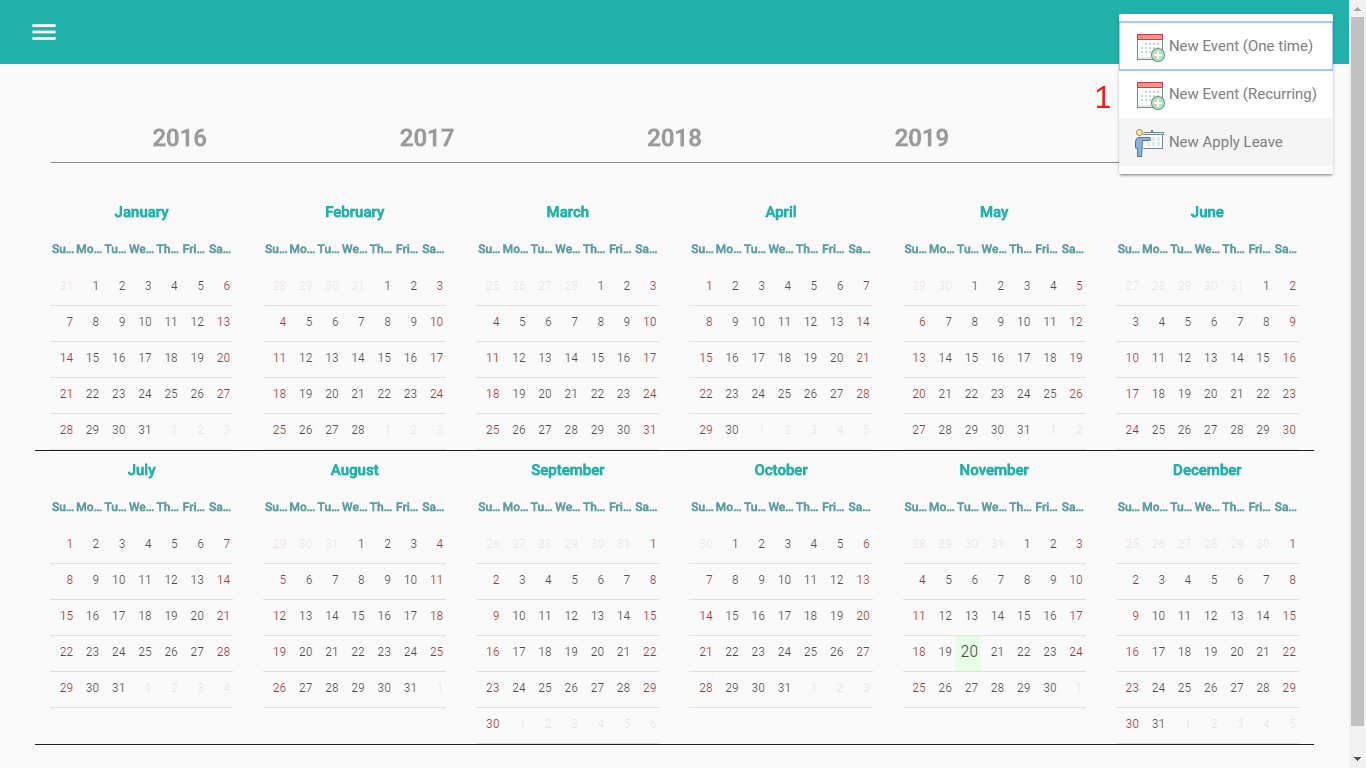


Figure 67. < HR Master> Create recurring event (1)

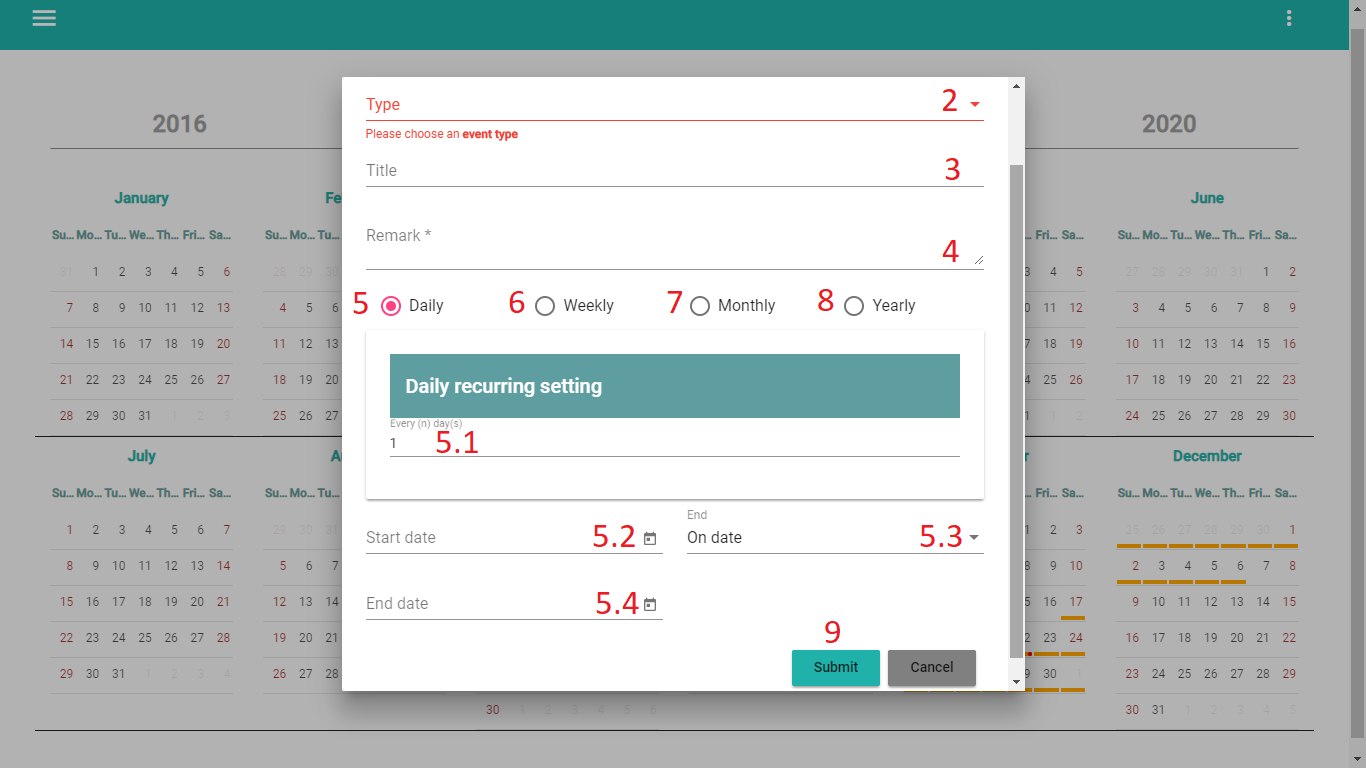


Figure 68. <HR Master> Create recurring event (2)

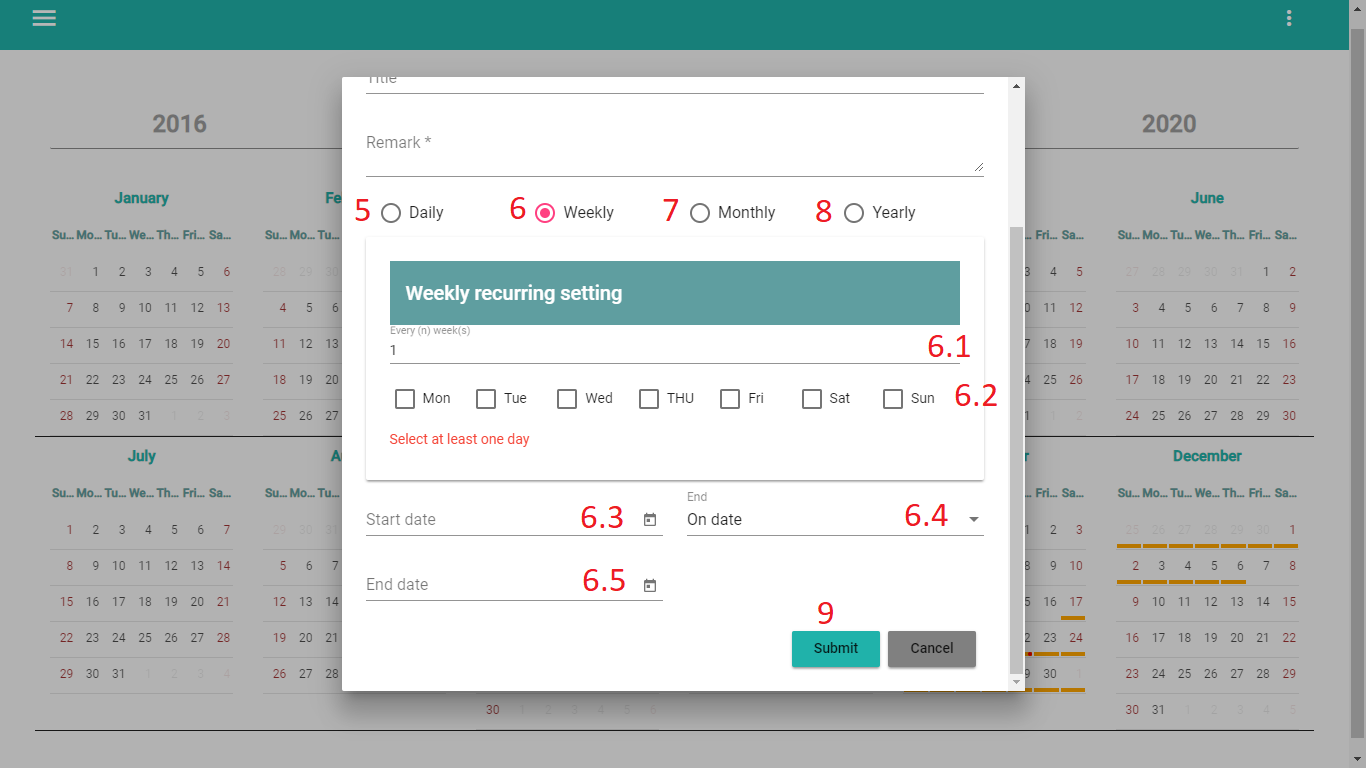
****

Figure 69. <HR Master > Create recurring event (3)

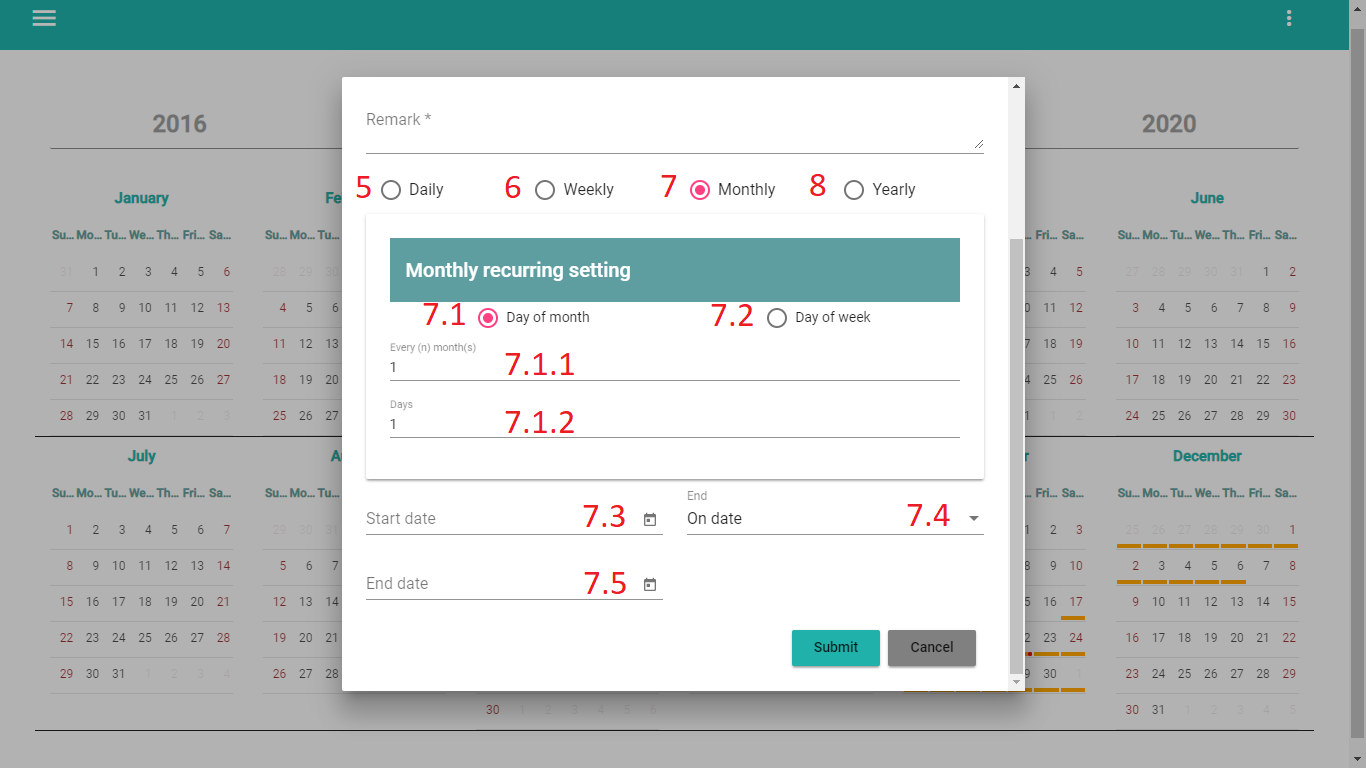


Figure 70. <HR Master> Create recurring event (4)

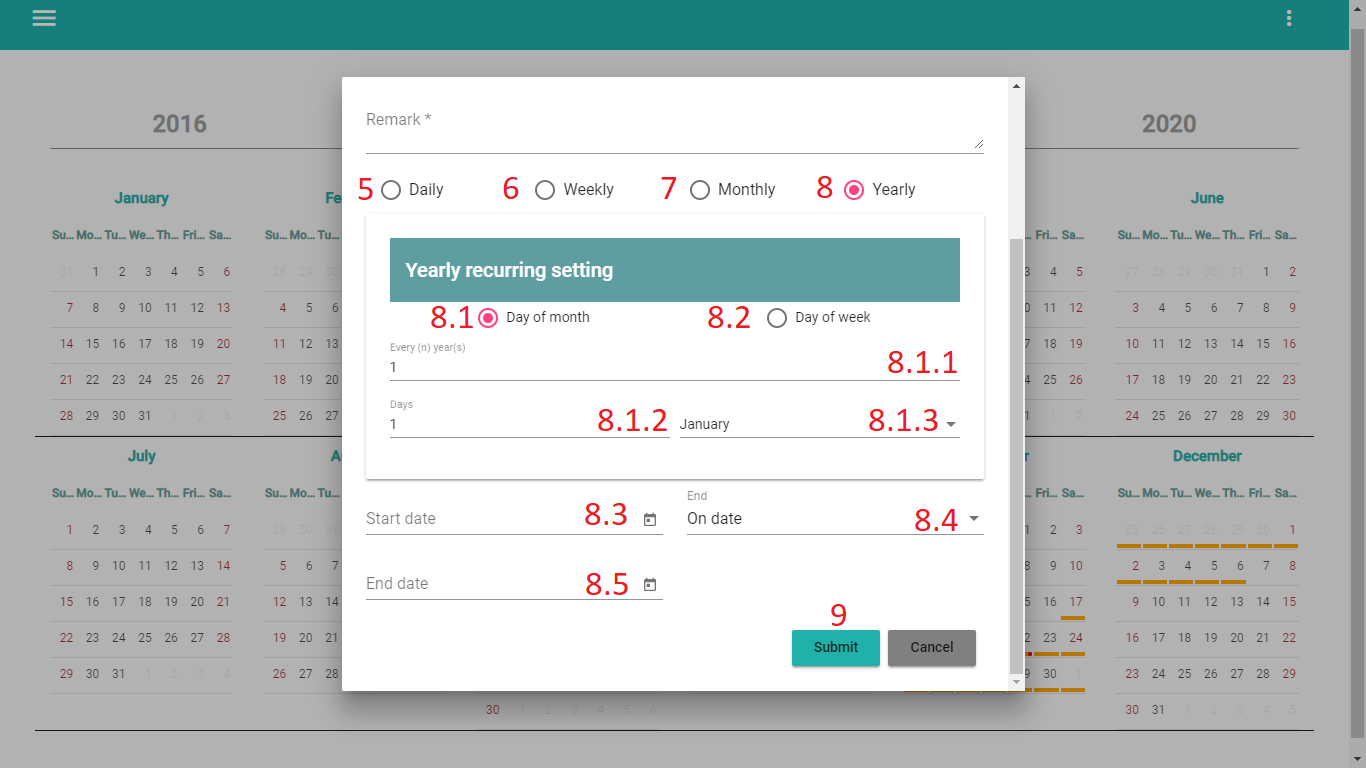


Figure 71. <HR Master> Create recurring event (5)

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Click on “New Event (Recurring)” button |
| 2 | Select type of event. |
| 3 | Fill in Title Event. |
| 4 | Fill in Remarks. |
| 5 | Select type of Recurring Daily (Optional) |
| 5.1 | Fill in Every(n) day(s) |
| 5.2 | Select start date |
| 5.3 | Select type on date |
| 5.4 | Select end date |
| 6 | Select type of Recurring Weekly (Optinal) |
| 6.1 | Fill in Every(n) week(s) |
| 6.2 | Select day of week |
| 6.3 | Select start date |
| 6.4 | Select type on date |
| 6.5 | Select end date |
| 7 | Select type of Recurring Monthly (Optinal) |
| 7.1 | Select type of recurring |
| 7.1.1 | Fill in Every(n) month(s) |
| 7.1.2 | Fill in Days |
| 7.3 | Select start date |
| 7.4 | Select type on date |
| 7.5 | Select end date |
| 8 | Select type of Recurring Yearly (Optinal) |
| 8.1 | Select type of recurring |
| 8.1.1 | Fill in Every(n) month(s) |
| 8.1.2 | Fill in Days |
| 8.1.3 | Select month of year. |
| 8.3 | Select start date |
| 8.4 | Select type on date |
| 8.5 | Select end date |
| 9 | Click button “Submit” |

Table 80. <User manual> Create recurring event.

1. **Select Year**

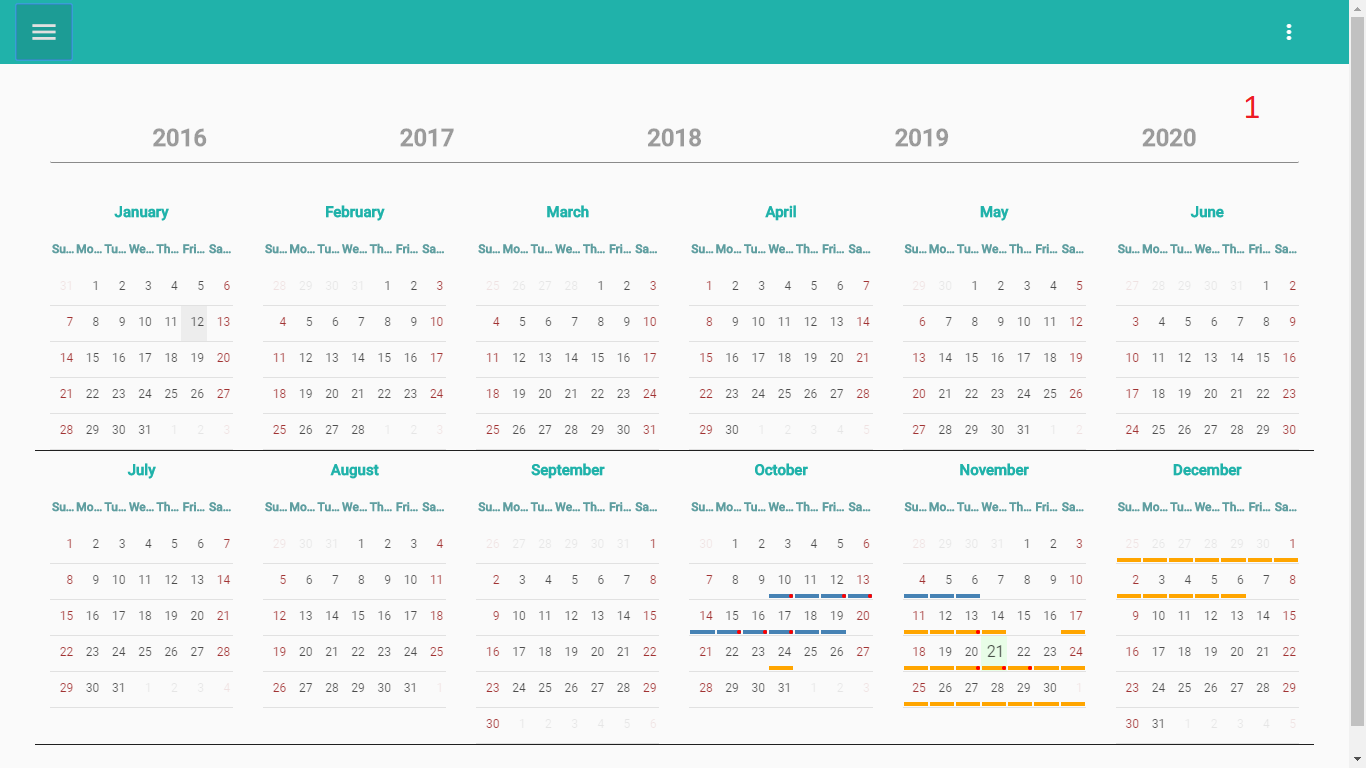


Figure 72. <HR Master> Select Year

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Click on Year number in taskbar year. |

Table 81. <User manual> Select Year

1. **Select Month**

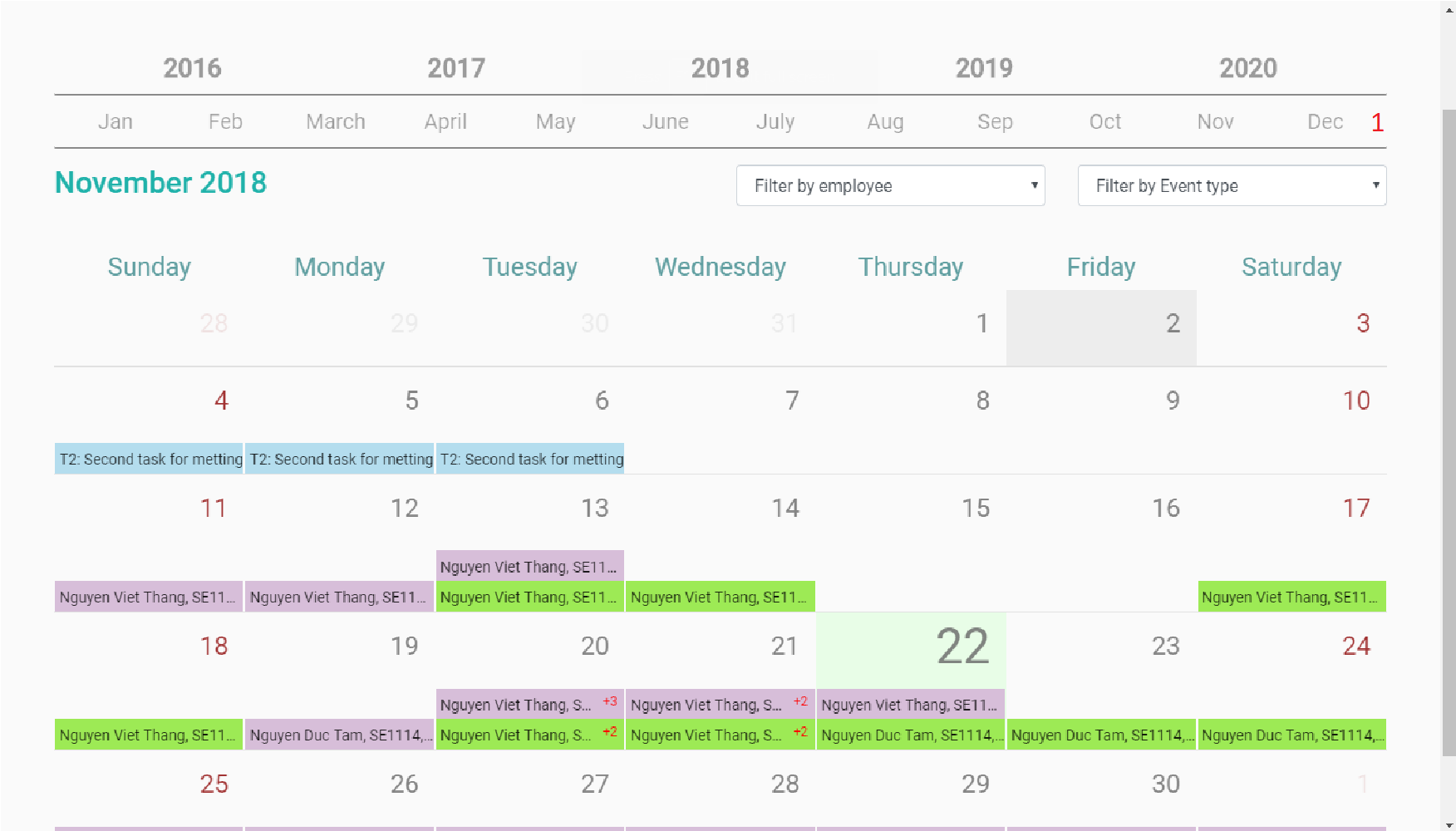


Figure 73. <HR Master> Select Month

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Click name month in month taskbar. |

Table 82. <User manual> Select Month

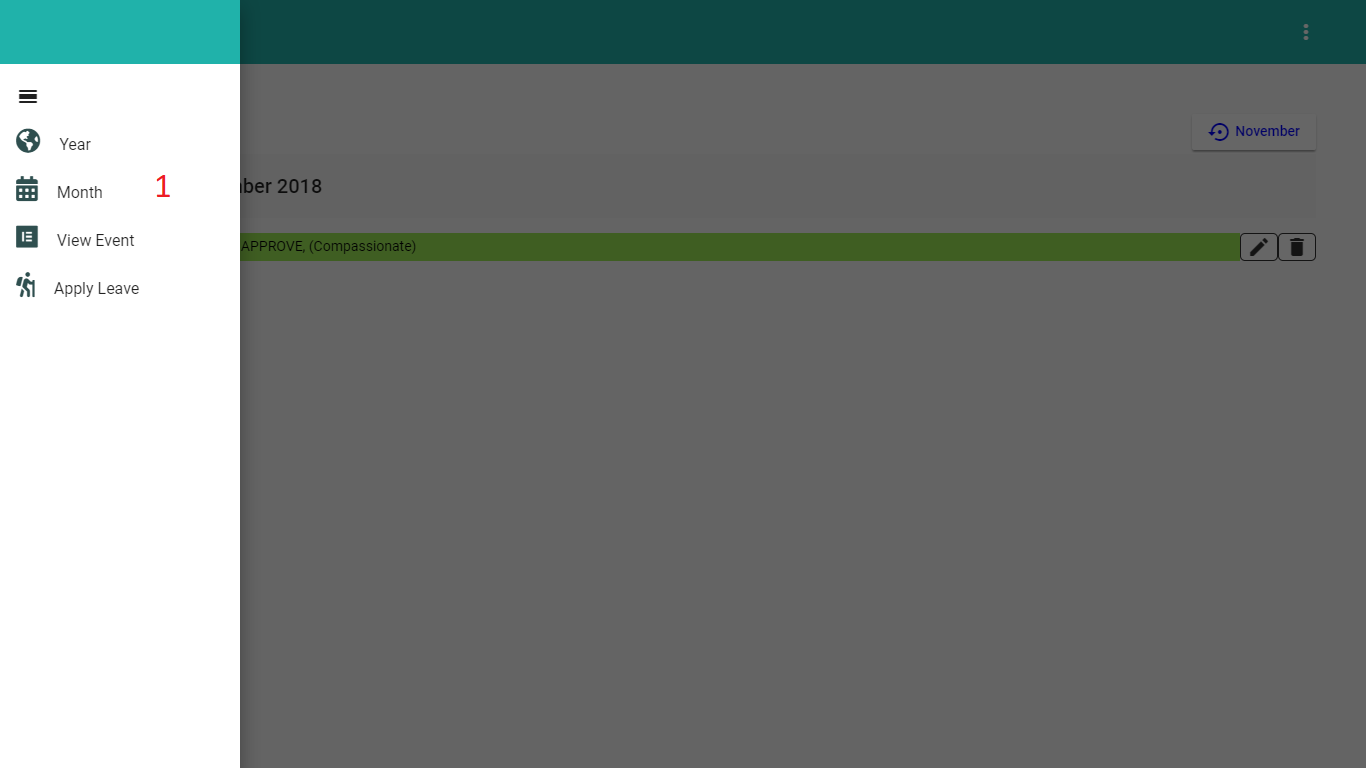
1. **Delete event and leave request.**

Figure 74. . <HR Master> Delete event and leave request (1)

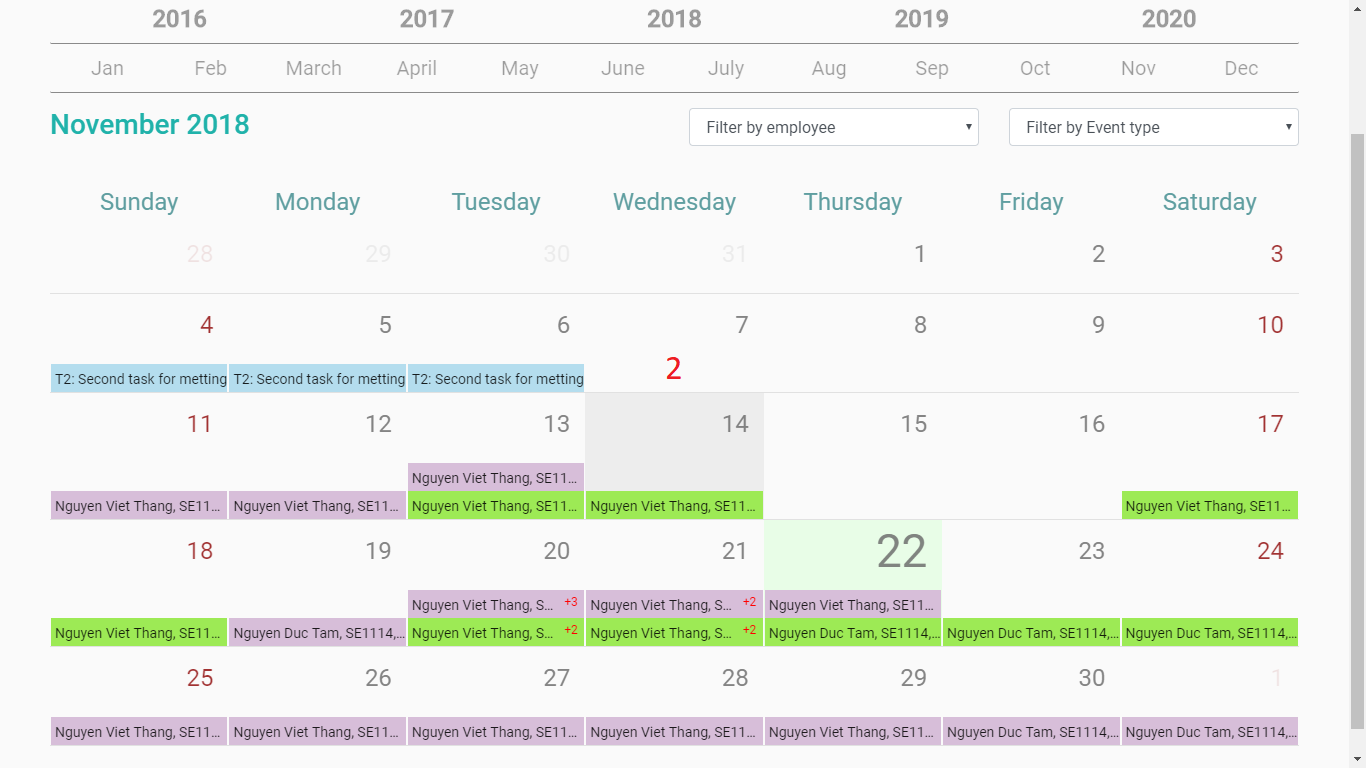
****

Figure 75. <HR Master> Delete event and leave request (2)

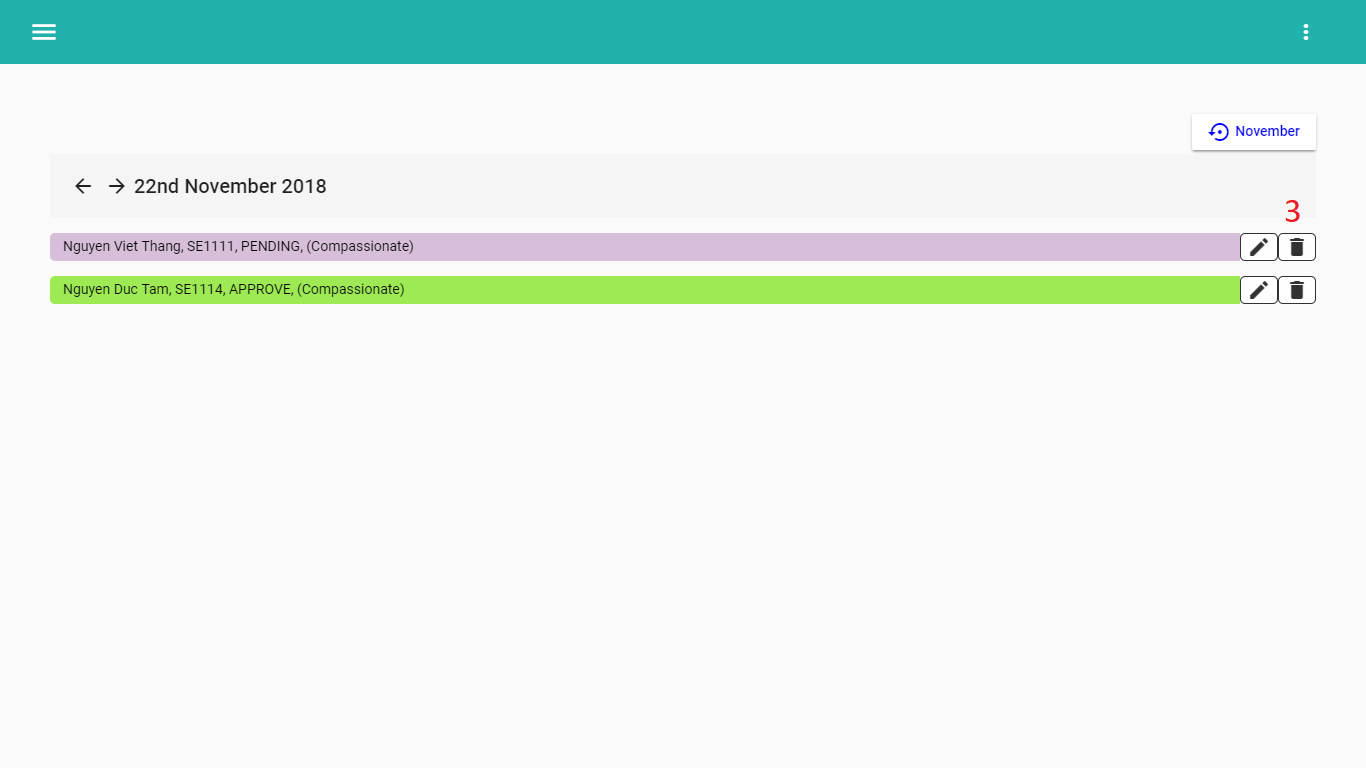


Figure 76. <HR Master> Delete event and leave request (3)

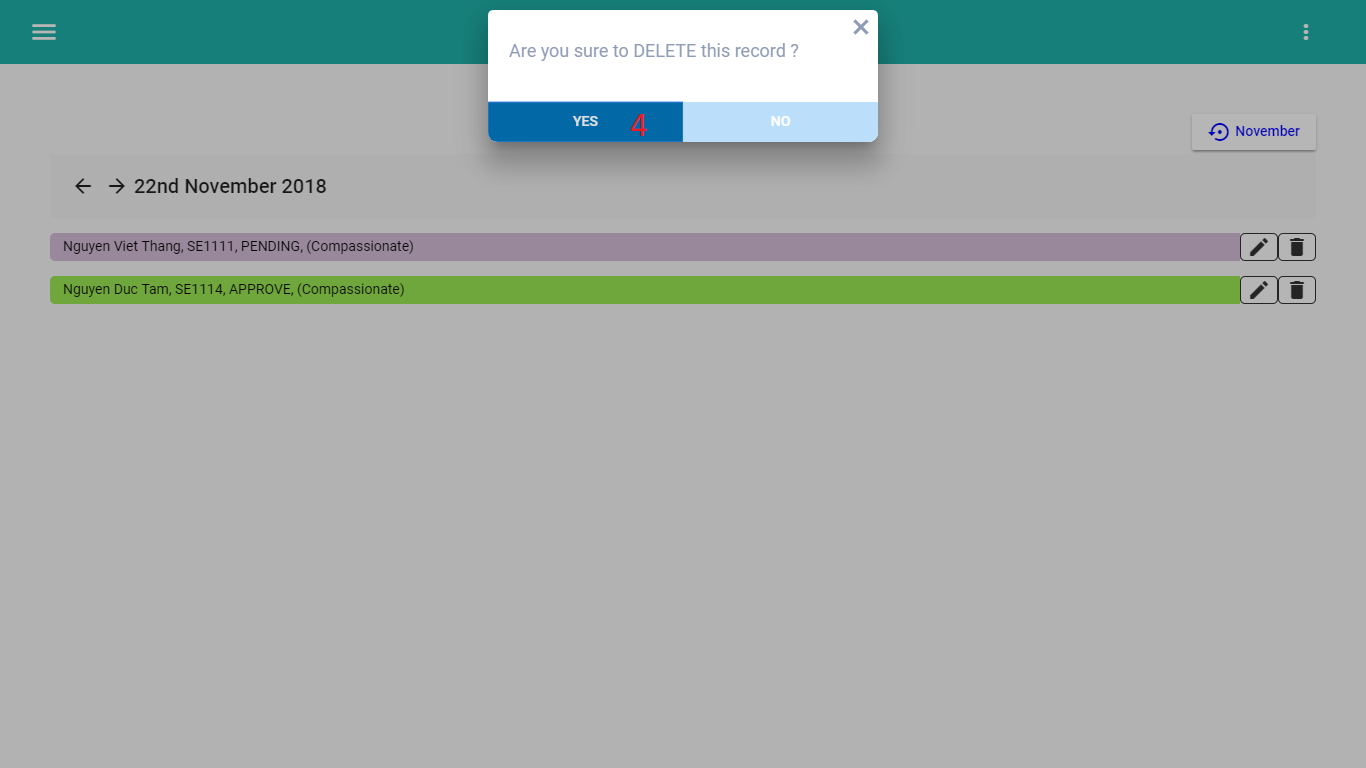
****

Figure 77. <HR Master> Delete event and leave request (4)

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Select in button Month or View Event |
| 2 | Select day had event wana Delete. |
| 3 | Click button “Delete” (recycle bin icon). |
| 4 | Click button “Yes” |

Table 83. <User manual> Filter in month view.

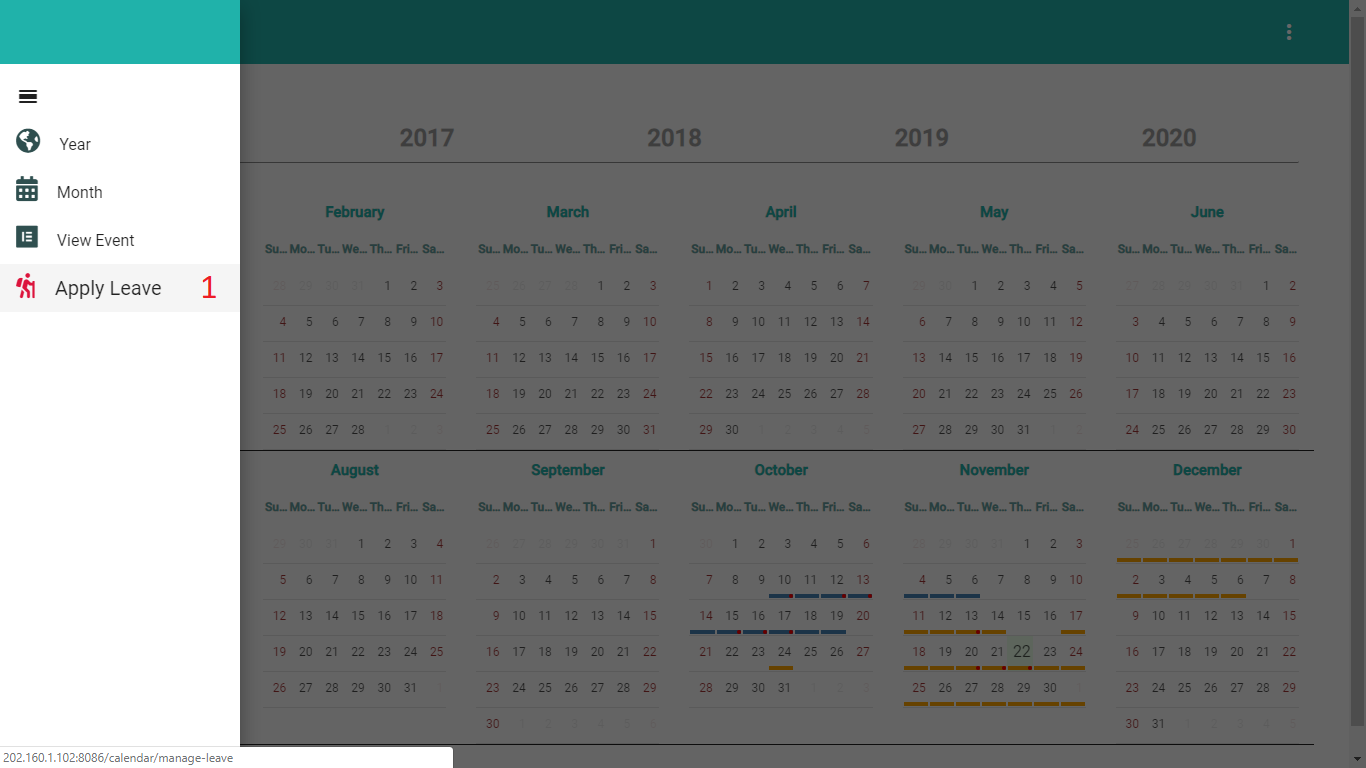
1. **Change Apply Leave status.**

Figure 78. <HR Master> Change Apply Leave Status (1)

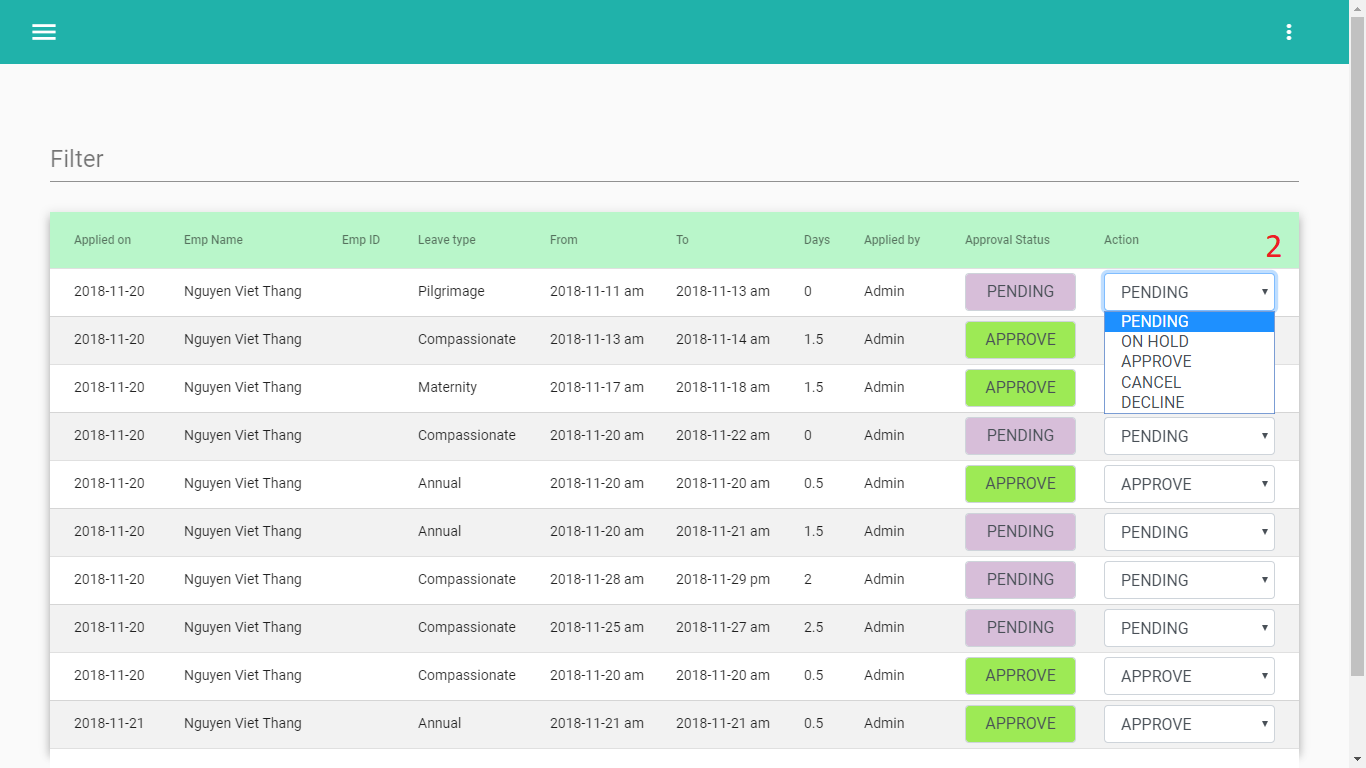
****

Figure 79. <HR Master> Change Apply Leave Status (2)

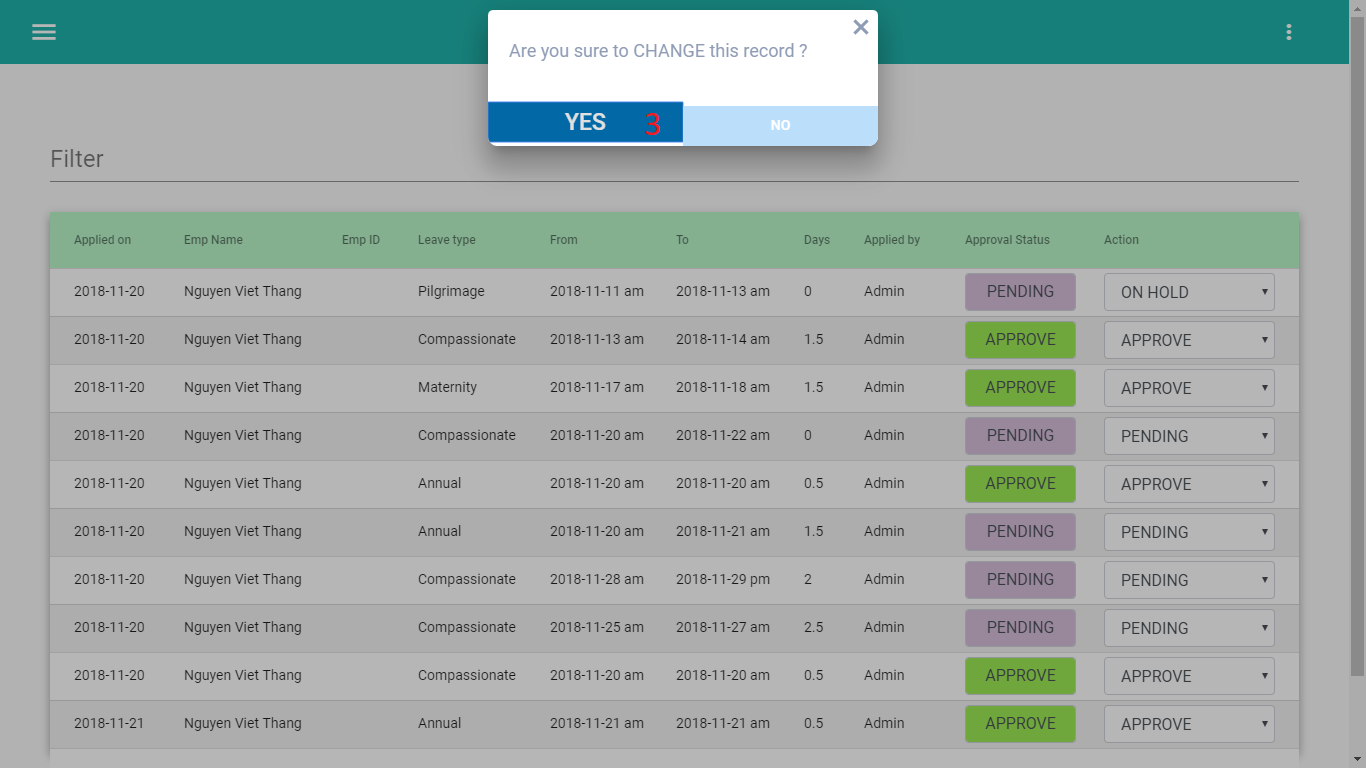
****

Figure 80. <HR Master> Change Apply Leave Status (3)

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Click button “Apply Leave” |
| 2 | Select type to Change in Combo-box list type |
| 3 | Click button “Yes” |

Table 84. <User manual> Change Apply Leave Status.

1. **Filter in month view.**

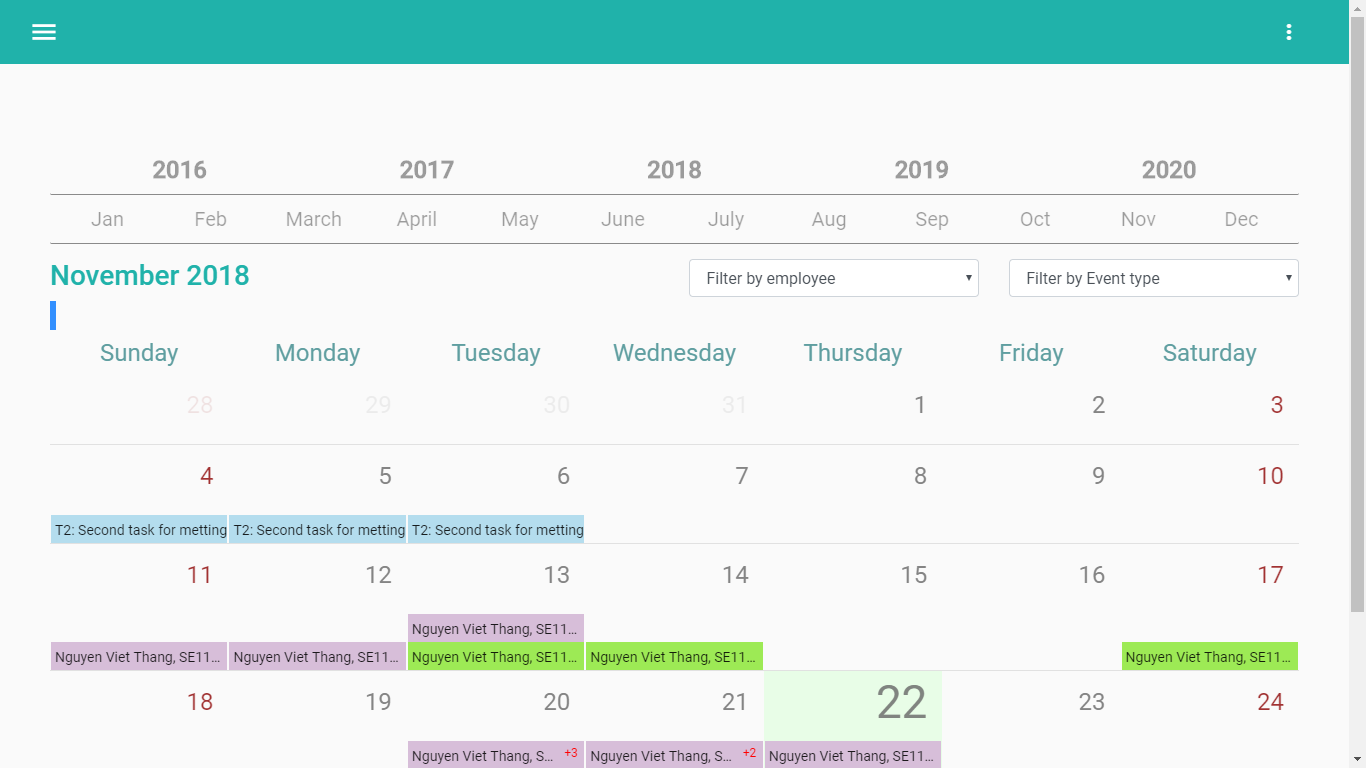


Figure 81. <HR Master> Filter in month view.

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Select in Filter by Employee |
| 2 | Select in Filter by Event type |

Table 85. <User manual> Filter in month view.

### **2.2. Mobile Application**

1. **Login**

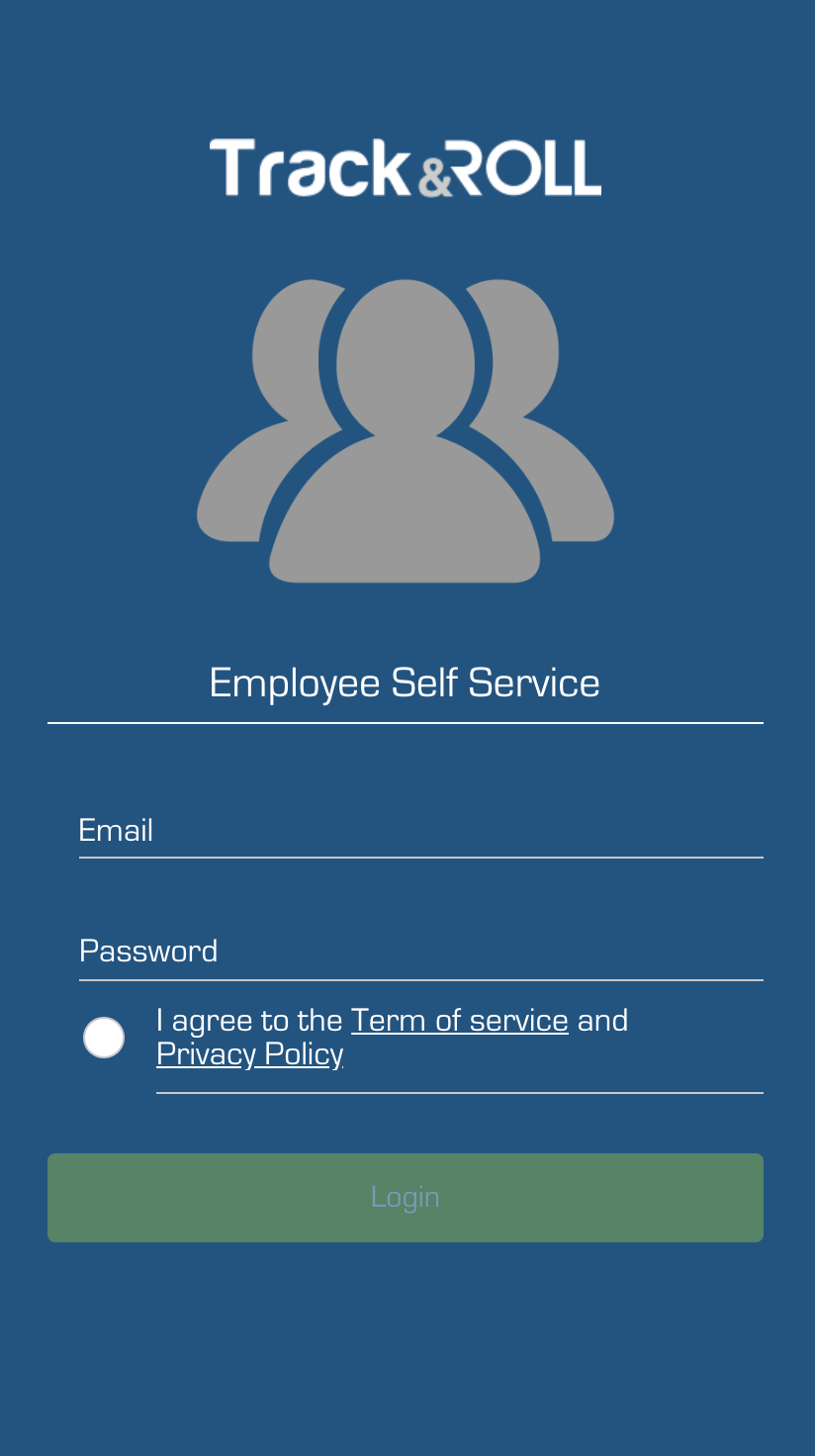
****

Figure 82. <User Manual> Mobile application - Login

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Enter username. |
| 2 | Enter password. |
| 3 | Click on “I agree to the Term of service and Privacy Policy” |
| 4 | Click on “Login” button |

Table 86. <User Manual> Mobile application - Login

1. **Navbar Menu**

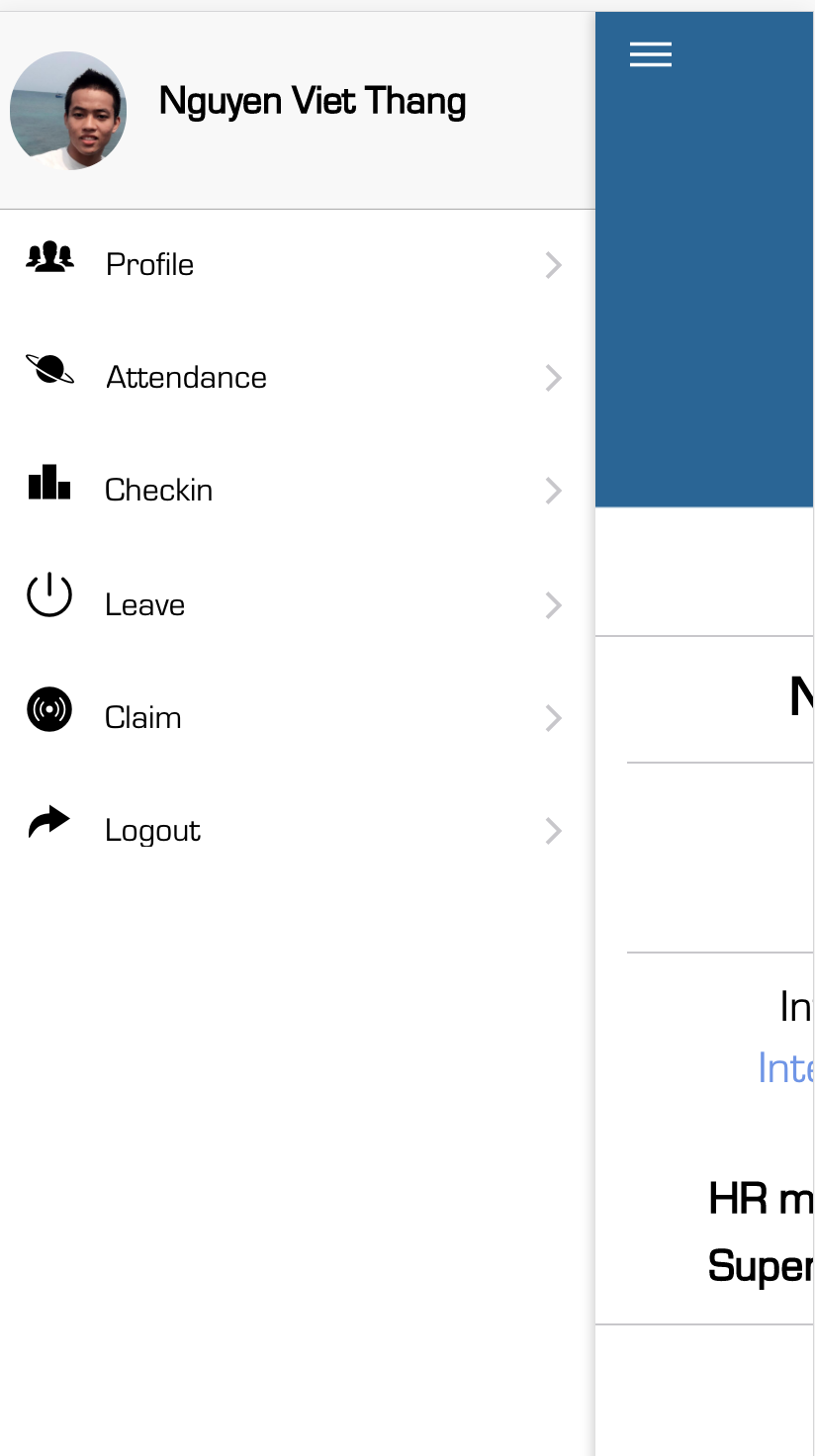
****

Figure 83. <User Manual> Mobile application - Menu

1. **View Profile**

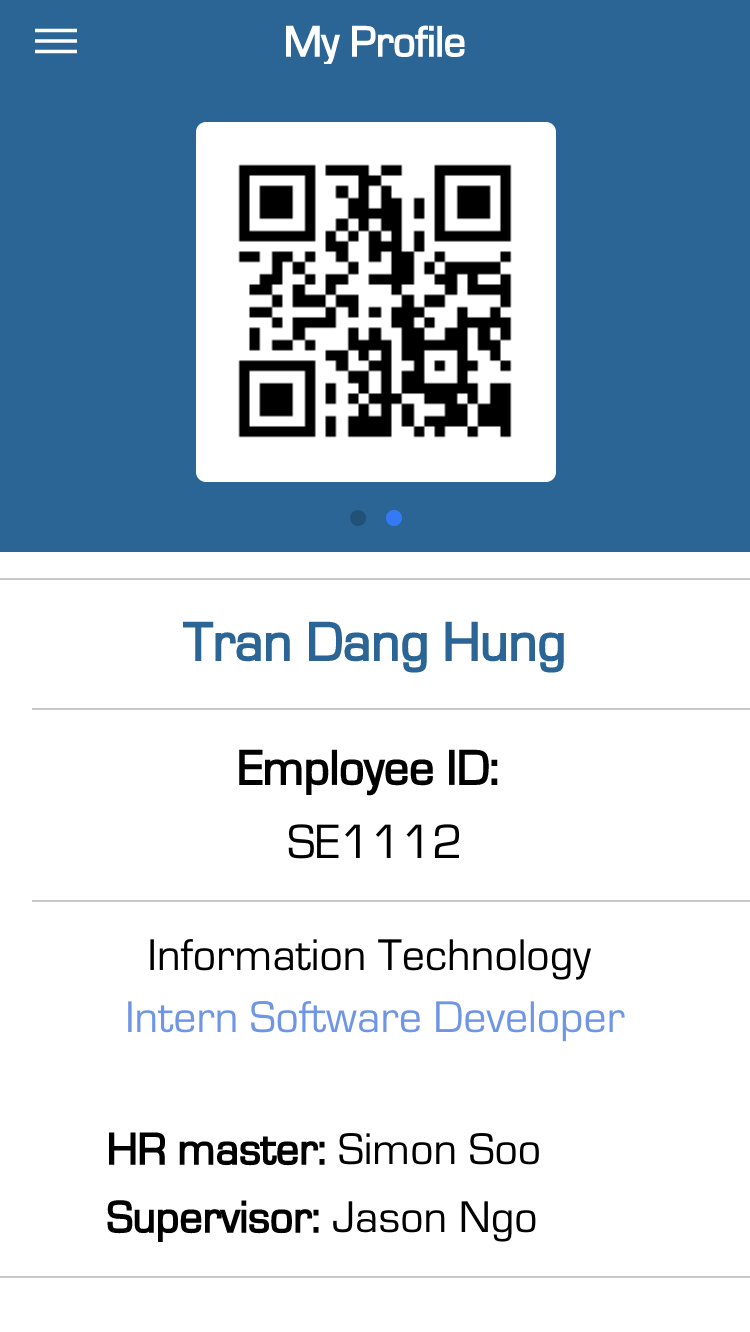
****

Figure 84. <User Manual> Mobile application - View Profile

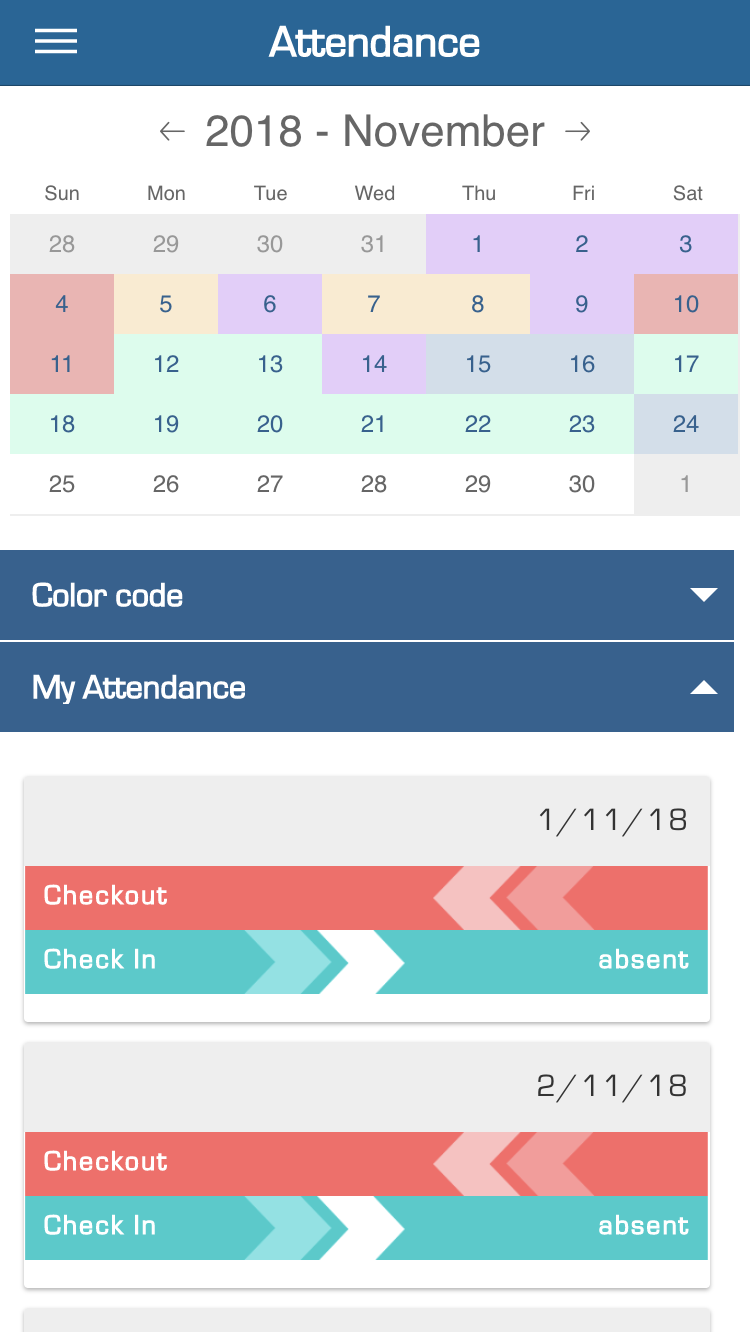
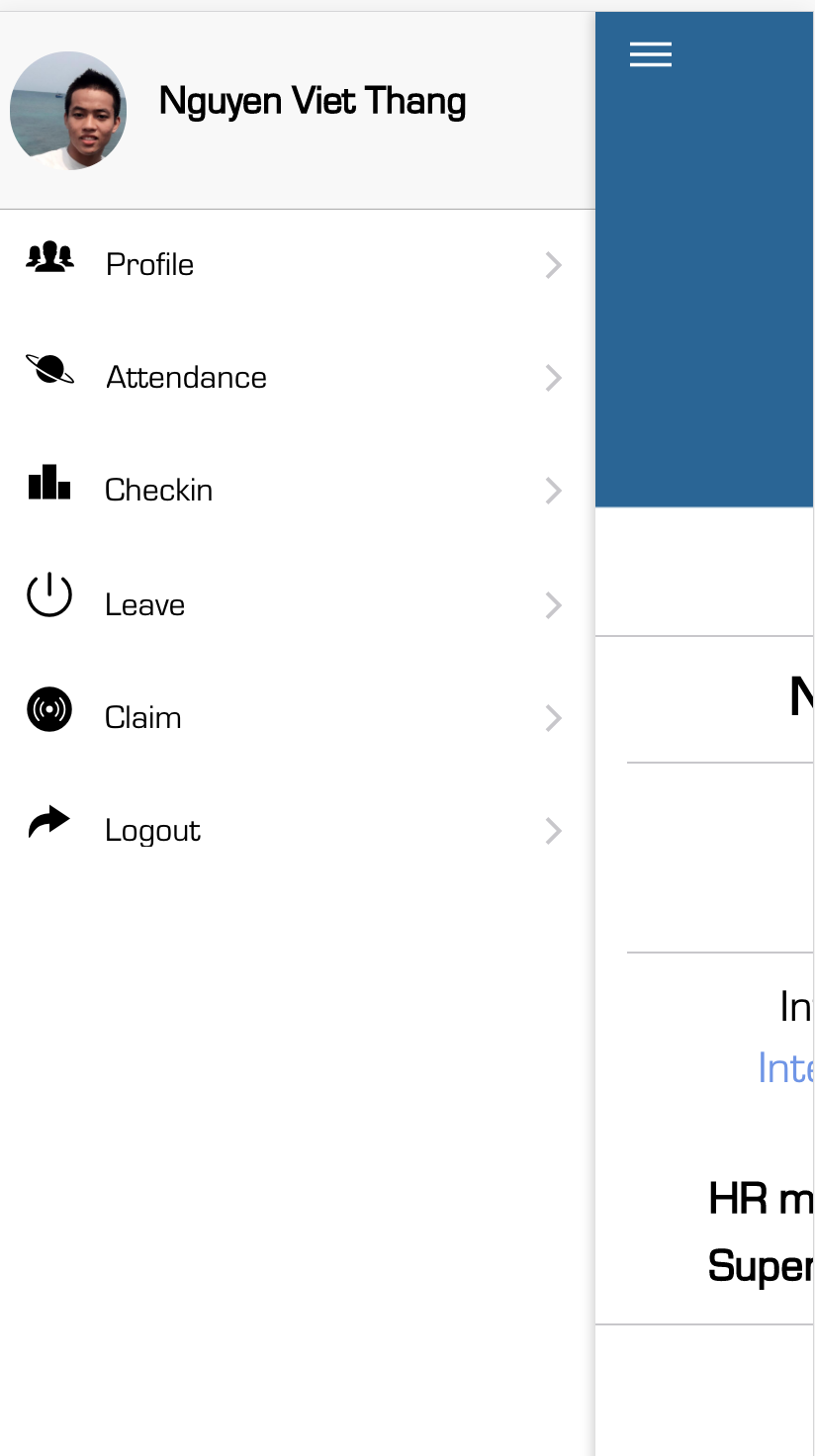
1. **View Attendance**

Figure 85. <User Manual> Mobile application - Attendance

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Click navbar |
| 2 | Click Attendance |
| 3 | View Calendar attendance follow color code |

Table 87. <User Manual> Mobile application - View Attendance

1. **Check in**

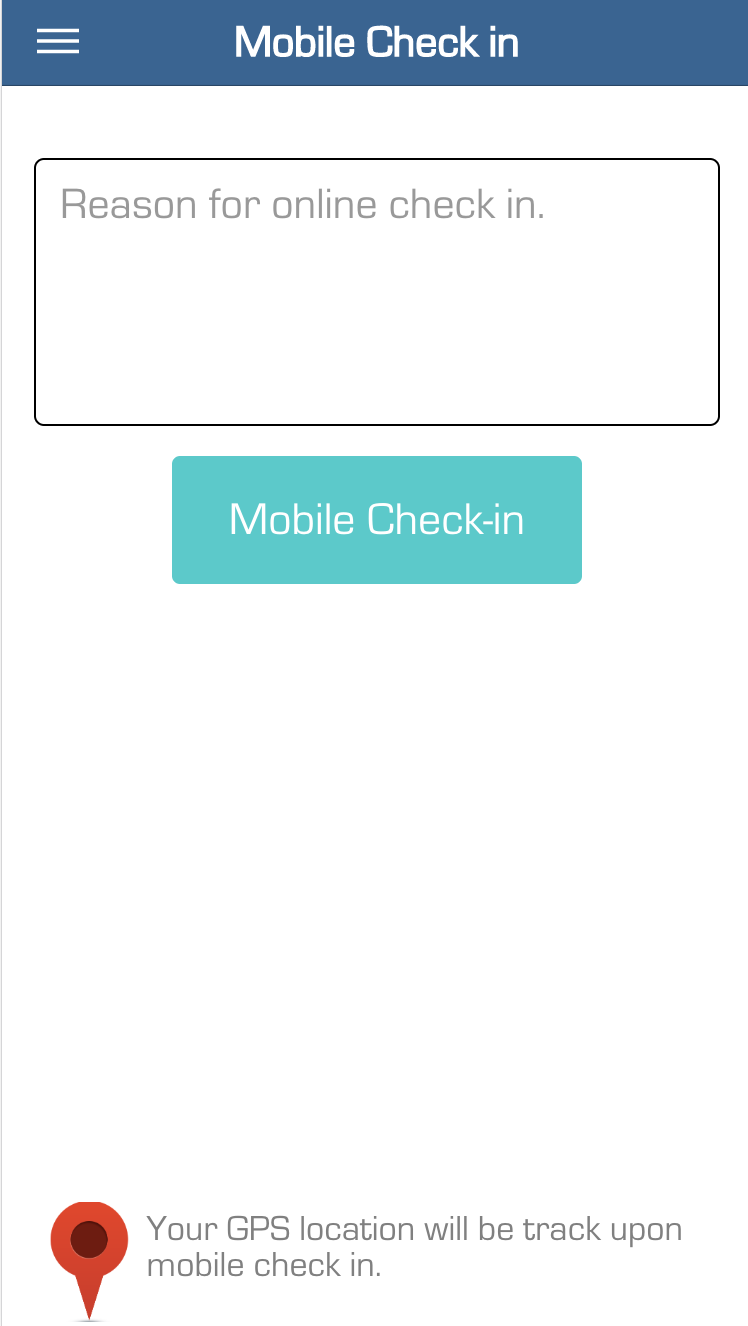
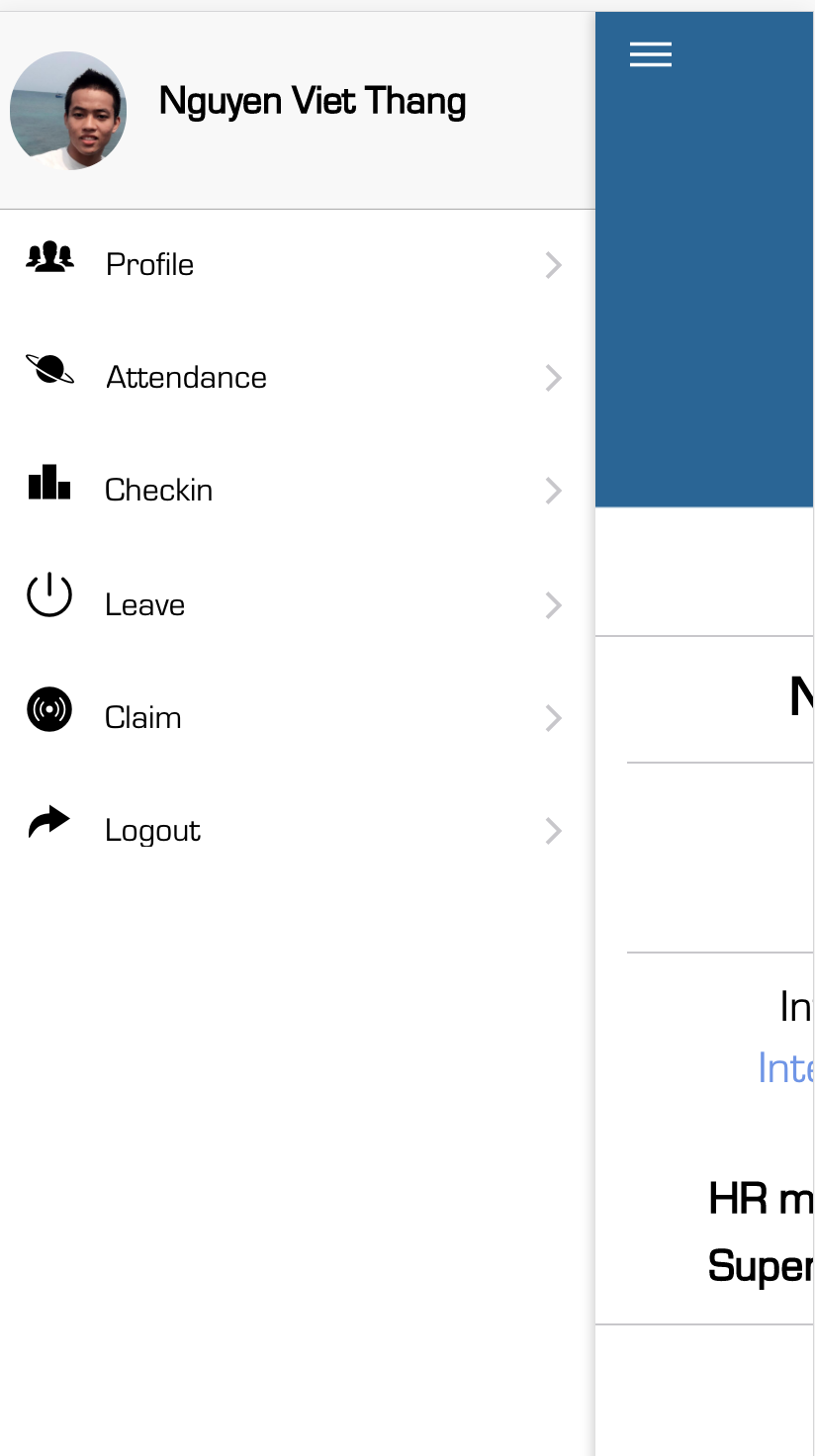
****

Figure 86. <User Manual> Mobile application - View Attendance

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Click navbar |
| 2 | Click Check In |
| 3 | Enter Reason for check in |
| 4 | Click Mobile Check In |

Table 88. <User Manual> Mobile application - Check in Online

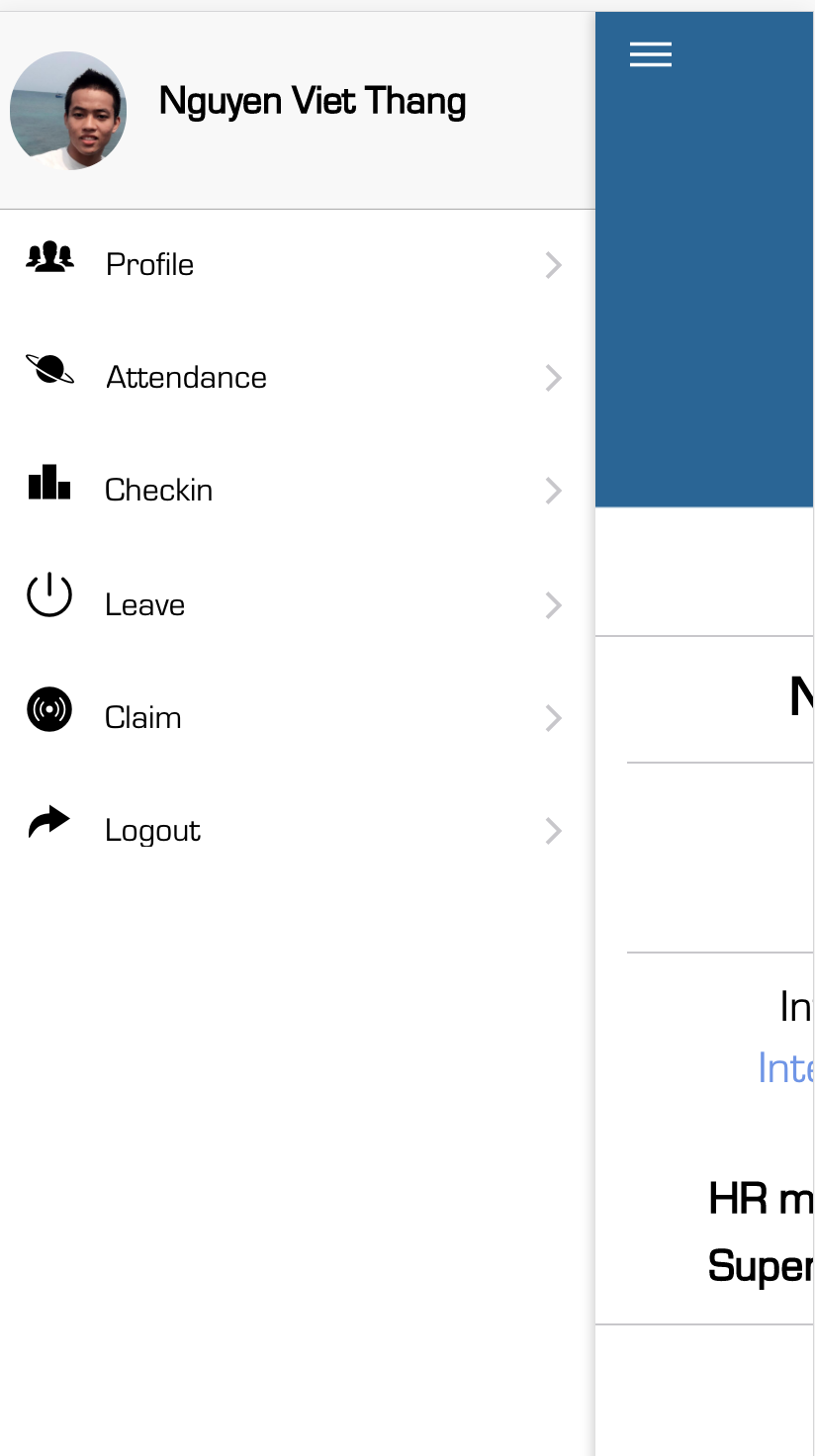
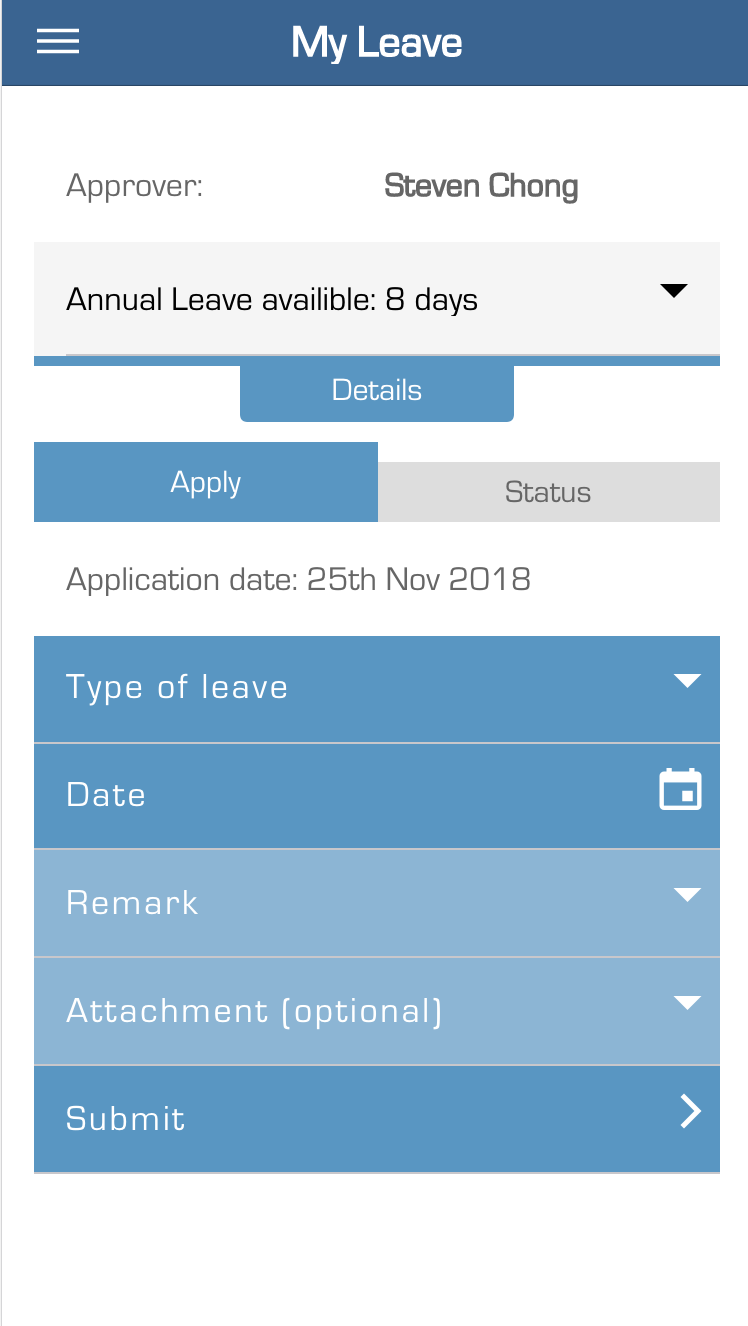
1. **Apply Leave**

Figure 87. <User Manual> Mobile application - Apply Leave

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Click navbar |
| 2 | Click Leave |
| 3 | Enter Input Type of Leave, Date, Remark, Attachment |
| 4 | Click Submit |

Table 89. <User Manual> Mobile application - Apply Leave

1. **View Leave**

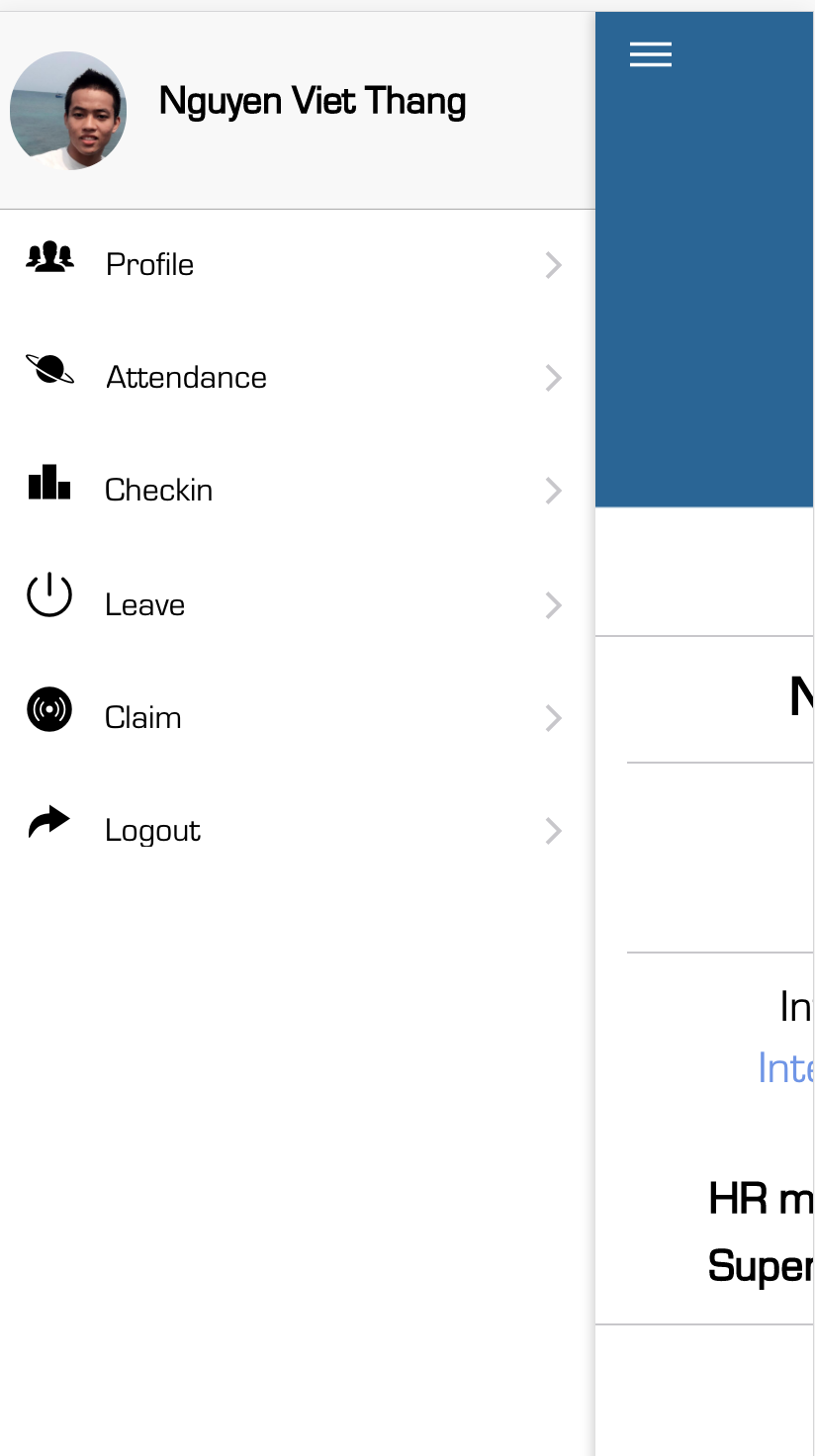
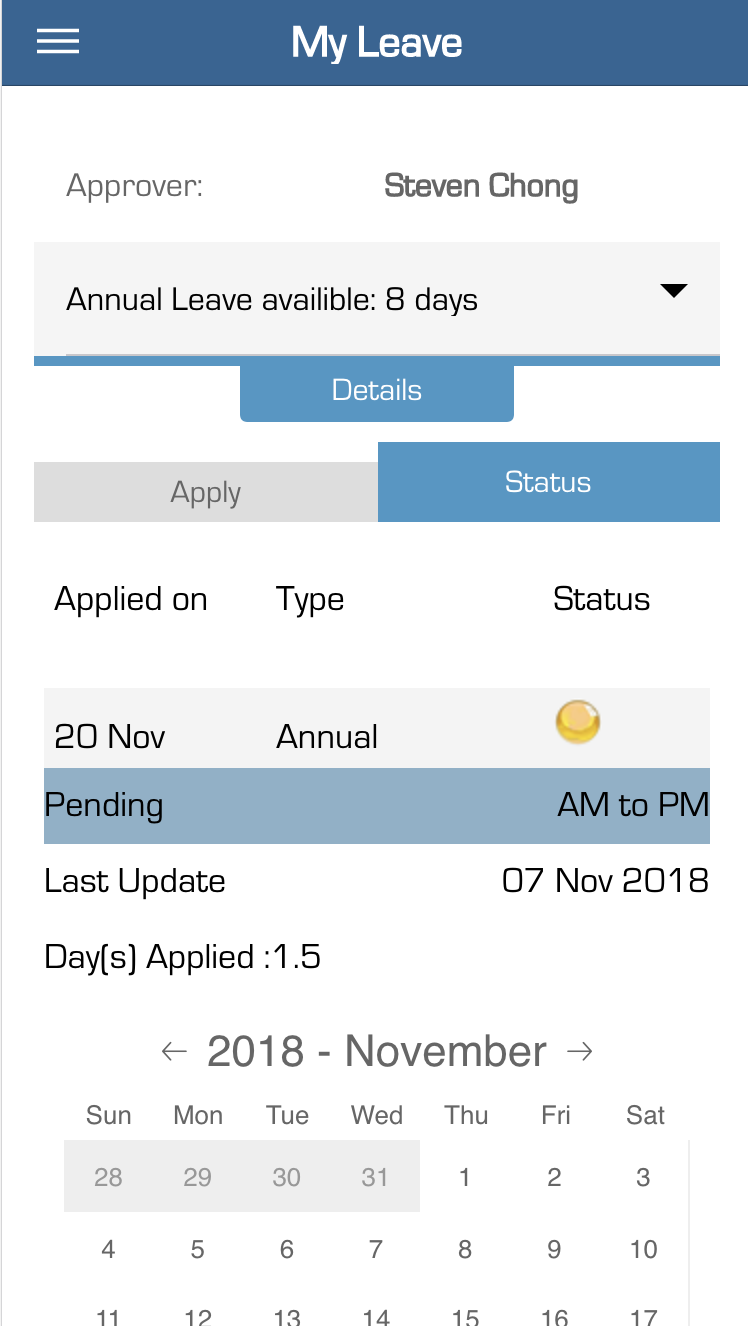
****

Figure 88. <User Manual> Mobile application - My Leave

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Click navbar |
| 2 | Click Leave |
| 3 | Click Status |
| 4 | Click on Leave to View Details |

Table 90. <User Manual> Mobile application - View My Leave

1. **Apply Claim**

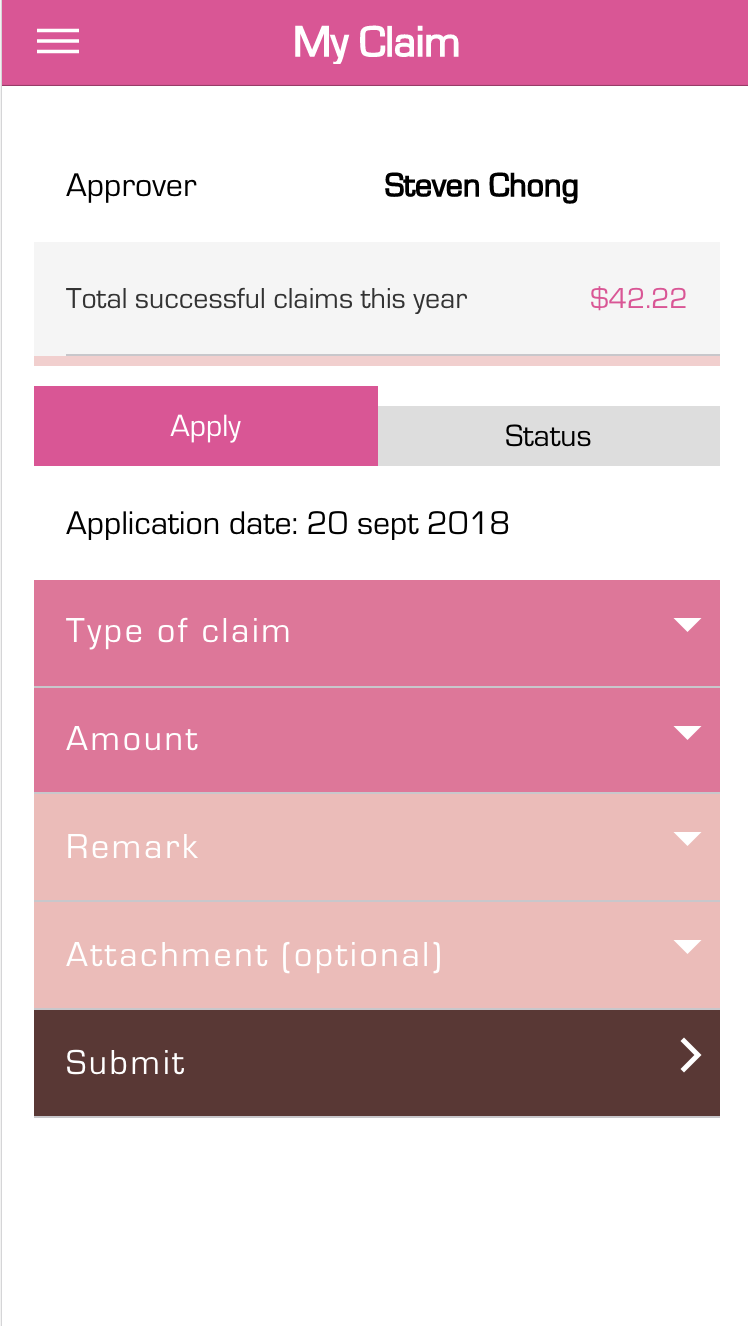
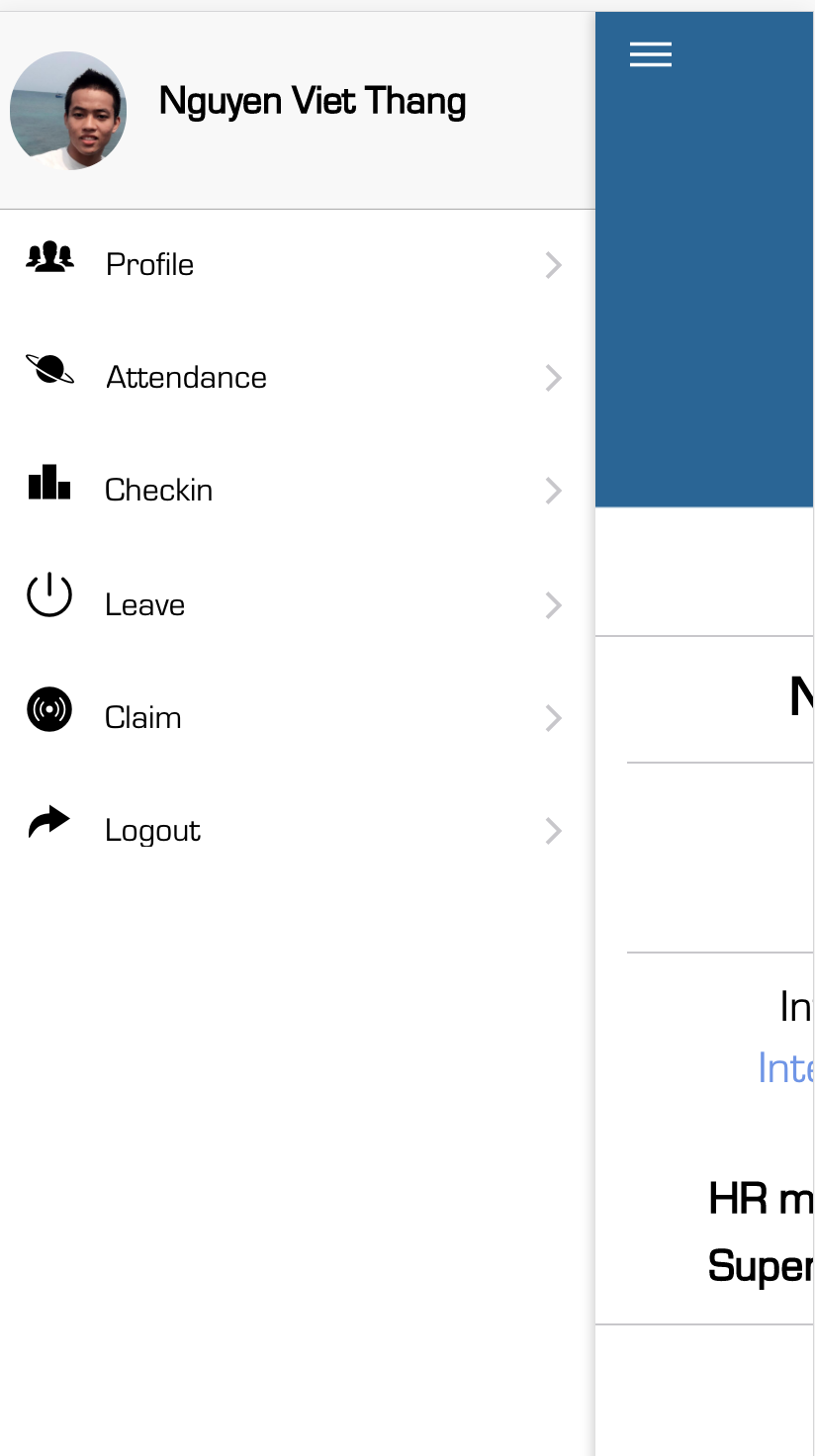
****

Figure 89. <User Manual> Mobile application - Apply Claim

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Click navbar |
| 2 | Click Claim |
| 3 | Enter Input Type of Claim, Amount, Remark, Attachment |
| 4 | Click Submit |

Table 91. <User Manual> Mobile application - Apply Claim

1. **View Claim**

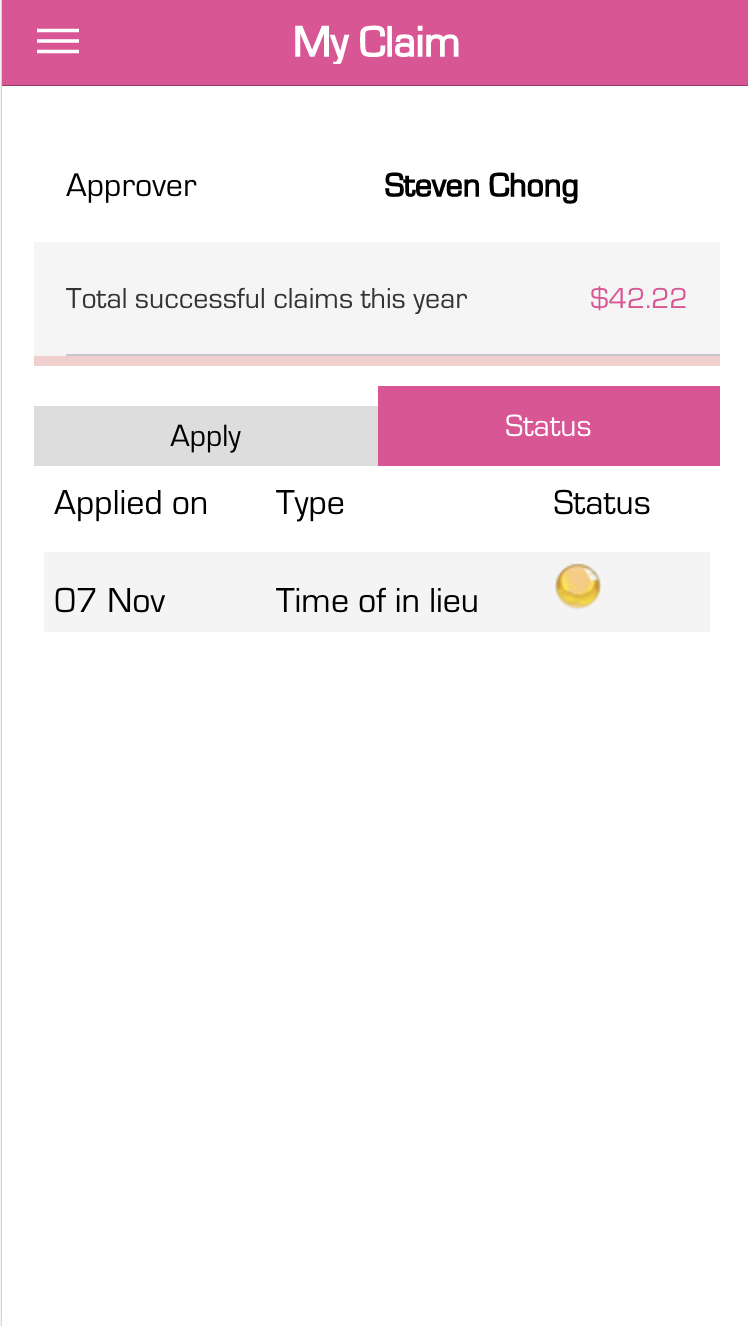
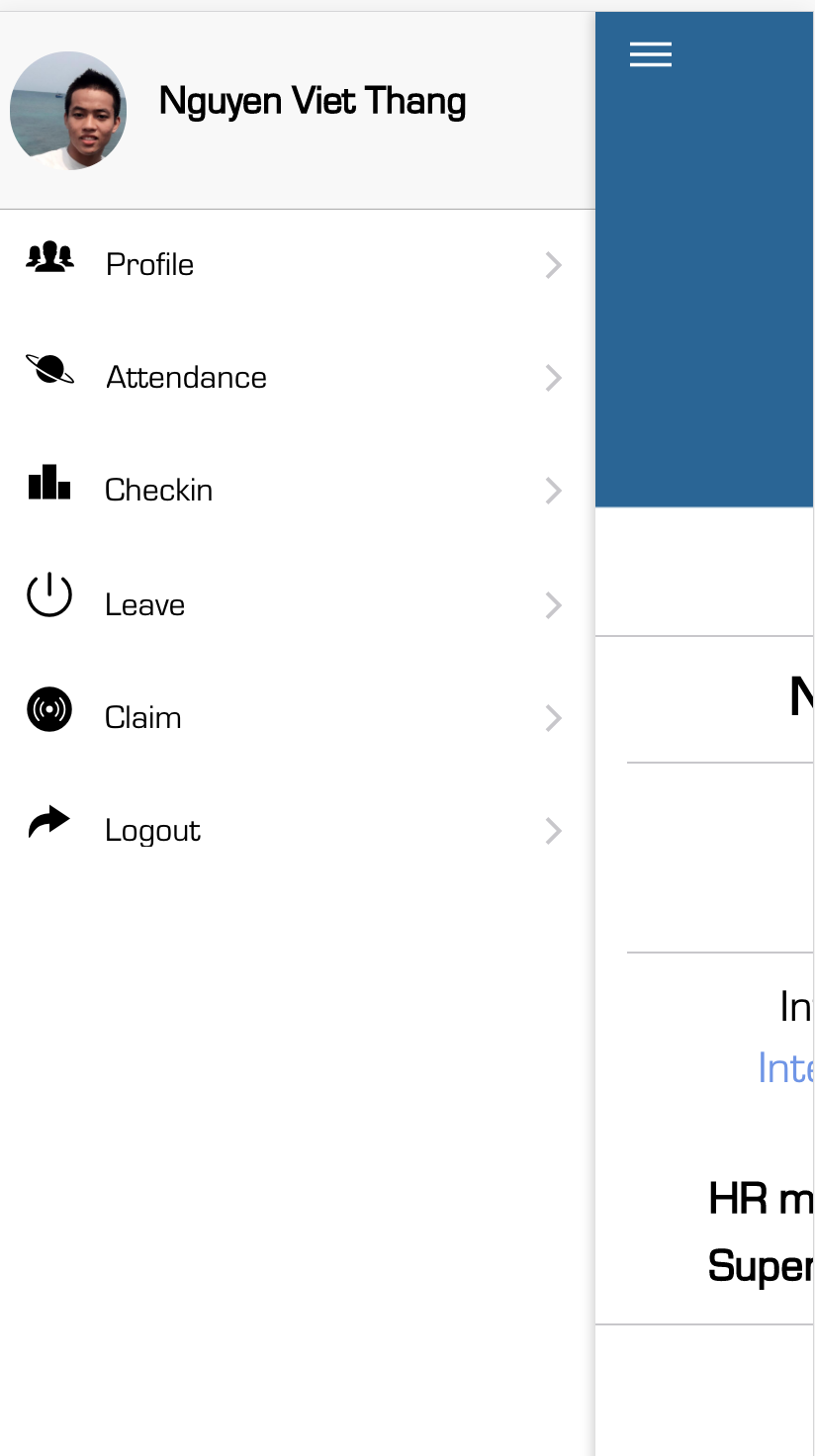
****

Figure 90. <User Manual> Mobile application - My Claim

|  |  |
| --- | --- |
| **Step** | **Description** |
| 1 | Click navbar |
| 2 | Click Claim |
| 3 | Click Status |
| 4 | Click on Claim to View Details |

Table 92. <User Manual> Mobile application - View My Claim

**G.** **Appendix**

1. **“Capstone Project Guideline” by FPT University**